

Manager: Aggregating Insights from Unimodal Experts in Two-Tower VLMs and MLLMs

Xiao Xu, Libo Qin, Wanxiang Che and Min-Yen Kan, *Senior Member, IEEE*

Abstract—Two-Tower Vision–Language Models (VLMs) have demonstrated strong performance across various downstream VL tasks. While BridgeTower further enhances performance by building bridges between encoders, it (i) suffers from ineffective layer-by-layer utilization of unimodal representations, (ii) restricts the flexible exploitation of different levels of unimodal semantic knowledge, and (iii) is limited to the evaluation on traditional low-resolution datasets only with the Two-Tower VLM architecture. In this work, we propose Manager, a lightweight, efficient and effective plugin that adaptively aggregates insights from different levels of pre-trained unimodal experts to facilitate more comprehensive VL alignment and fusion. First, under the Two-Tower VLM architecture, we introduce ManagerTower, a novel VLM that introduces the manager in each cross-modal layer. No matter with or without VL Pre-training, ManagerTower outperforms previous strong baselines and achieves superior performance on 4 downstream VL tasks. Moreover, we extend our exploration to the latest Multimodal Large Language Model (MLLM) architecture. We demonstrate that LLaVA-OV-Manager significantly boosts the zero-shot performance of LLaVA-OV across different categories of capabilities, images, and resolutions on 20 downstream datasets, whether the multi-grid algorithm is enabled or not. In-depth analysis reveals that both our manager and the multi-grid algorithm can be viewed as a plugin that improves the visual representation by capturing more diverse visual details from two orthogonal perspectives (depth and width). Their synergy can mitigate the semantic ambiguity caused by the multi-grid algorithm and further improve performance. Code and models are available at <https://github.com/LooperXX/ManagerTower>.

Index Terms—Vision–Language Model, Multimodal Large Language Model, Representation Learning.

I. INTRODUCTION

RECENTLY, the field of Vision–Language (VL) representation learning has gained significant attention, driven by advancements in Vision–Language Pre-training (VLP) techniques. VLP aims to learn transferable multimodal knowledge from extensive image–text pairs into Vision–Language Models (VLMs), which can improve VL representation and thus further improve performance on various downstream tasks, such as visual question answering [2], visual entailment [3], visual reasoning [4], and image–text retrieval [5].

This work was supported by the National Natural Science Foundation of China (NSFC) via grant 62236004, 62441603 and 62476073. This work was done while Xiao Xu was visiting the National University of Singapore. (Corresponding authors: Wanxiang Che, Libo Qin.)

Xiao Xu and Wanxiang Che are with the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, 150001, Harbin, Heilongjiang, China (e-mail: xxu@ir.hit.edu.cn, car@ir.hit.edu.cn).

Libo Qin is with the School of Computer Science and Engineering, Central South University, 410083, Changsha, Hunan, China (e-mail: lbqin@csu.edu.cn).

Min-Yen Kan is with the School of Computing, National University of Singapore, 117417, Singapore (e-mail: knmyn@nus.edu.sg).

This work is an extension of our conference paper, ManagerTower [1], which was accepted at ACL 2023 (Oral), 10.18653/v1/2023.acl-long.811.

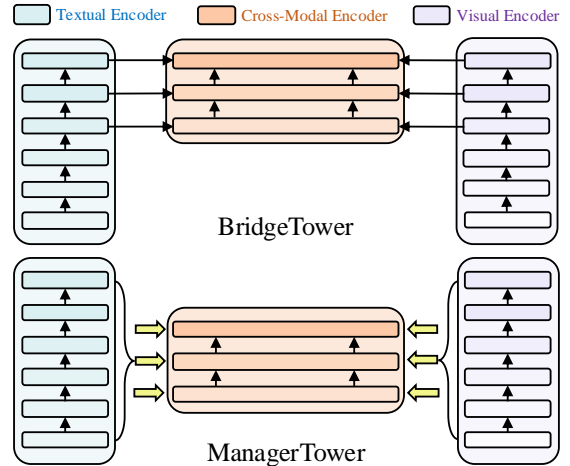


Fig. 1. A brief overview of BridgeTower and ManagerTower. Hollow arrows represent the transmission of multi-layer unimodal representations in ManagerTower, in contrast to the layer-by-layer transmission in BridgeTower.

Two-Tower VLM is a general architecture that processes visual and textual modalities with corresponding unimodal encoders and then fuses them in a cross-modal encoder. METER [6] and BridgeTower [7] are two representative Two-Tower VLMs. METER uses CLIP-ViT [8] and RoBERTa [9] as pre-trained unimodal encoders, but **overlooks** different levels of unimodal semantic knowledge contained in multi-layer unimodal representations. It only feeds the last-layer representation from each unimodal encoder into the cross-modal encoder, which may limit the model capability. To tackle this issue, as shown in Fig. 1, BridgeTower builds connections between multiple top unimodal layers and each cross-modal layer in a layer-by-layer manner, to exploit unimodal semantic knowledge at different levels.

In this work, we build upon the research of BridgeTower and advance it in four aspects: (a) **Ineffective** layer-by-layer utilization of multi-layer unimodal representations. Each cross-modal layer is limited to using a pre-defined unimodal layer representation, which restricts the utilization of different levels of unimodal semantic knowledge and the model capability. (b) **Strictly bound** the number of cross-modal layers to the number of unimodal layer representations the model can use. An increase in either side leads to a corresponding increase in the other side, leading to more parameters and computation cost, and poor scalability. (c) **Only** exploring the utilization of multi-layer unimodal representations in the Two-Tower VLM architecture. Lack of exploration in other VLM architectures, e.g., Multimodal Large Language Model (MLLM), limits the generality of

the conclusions. (d) **Limited** post-fine-tuning evaluation on datasets with low-resolution natural images. Constrained by the capability of traditional VLMs, the model cannot perform more challenging zero-shot evaluations on broader datasets, such as high-resolution document understanding.

For the first two aspects, under the Two-Tower VLM architecture, we propose a novel VLM, ManagerTower, that introduces managers in each cross-modal layer to aggregate multi-layer unimodal representations, as shown in Fig. 1. Each manager takes multi-layer unimodal representations as **insights** from pre-trained unimodal **experts** at different levels (layers), and then aggregates them to facilitate more comprehensive vision–language alignment and fusion. Inspired by the linear combination of layers method [10], we explore the feasibility of various designs of managers by evaluating and analyzing the performance on VQA_{v2} and Flickr30K datasets. The best manager, Adaptive Aggregation Unimodal Manager (AAUM), can **adaptively** aggregate multi-layer unimodal representations for different tokens in different samples in each cross-modal layer. Then, we pre-train ManagerTower with commonly used 4M VLP data and evaluate it on 4 downstream datasets. With the same pre-training, fine-tuning and evaluation settings as previous strong Two-Tower VLMs such as METER and BridgeTower, ManagerTower achieves superior performances on all datasets, and 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. Moreover, in principle, managers are scalable and flexible enough to be used as a **plugin**, easily integrated into any cross-modal encoder, and works well with any unimodal encoder.

For the last two aspects, we further extend the exploration of managers to the latest MLLM architecture, and introduce the manager to LLaVA-OV [11] to get LLaVA-OV-Manager, as shown in Fig. 2. Benefiting from the strong LLM and the multi-grid algorithm [12] capable of improving the supported image resolution in MLLMs, we can zero-shot evaluate the effectiveness of managers on **broader** downstream datasets, especially on high-resolution images. We demonstrate that, no matter with or without the multi-grid algorithm, managers can **significantly** improve the performance of MLLMs on 20 downstream datasets across different categories of capabilities, images, and resolutions. Further analysis reveals that both the manager and the multi-grid algorithm can be viewed as a **plugin** that improves the input visual representation. The manager introduces different levels of semantic knowledge into MLLMs, which can increase the **diversity** of attention weights and attention heads, thus helping **guide** the attention of MLLMs that use the multi-grid algorithm. Their synergy can capture more diverse visual details from two orthogonal perspectives (**depth** and **width**), mitigate the semantic ambiguity caused by the multi-grid algorithm and further improve performance.

II. PRELIMINARY

We briefly introduce the basic components of Two-Tower VLMs used by METER, BridgeTower, and ManagerTower.

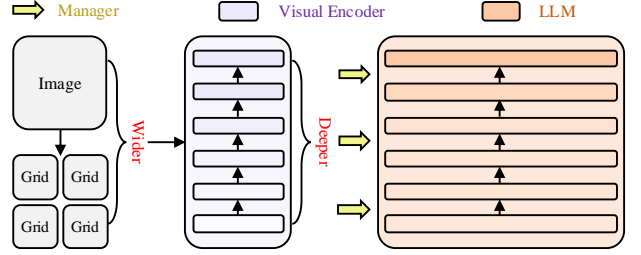


Fig. 2. Brief illustrations of LLaVA-OV-Manager. The base image and grids are encoded independently. Hollow arrows indicate the transmission of multi-layer visual representations aggregated by managers to the LLM at intervals.

A. Visual Encoder

CLIP-ViT, the visual encoder of CLIP [8], has been widely used in VLMs [6], [13]. Each input image is first transformed into a flattened sequence of patches, with a `[class]` token added at the beginning. Following a linear projection, position embeddings are added to the sequence to obtain the visual input \mathbf{V}_0 . The ℓ^{th} visual layer representation is computed as: $\mathbf{V}_\ell = \text{Encoder}_\ell^V(\mathbf{V}_{\ell-1})$, $\ell = 1 \dots L_V$, where ℓ is the layer index and L_V is the number of layers in the visual encoder.

B. Textual Encoder

RoBERTa [9] is widely used in VLMs [6], [14] due to its robust performance. The input text is tokenized with the byte-level Byte-Pair Encoding (BPE) [15], [16]. `[<s>]` and `[</s>]` tokens are added to the start and end of the sequence, respectively. Word embeddings and positional embeddings are then applied to the tokenized sequence to generate the visual input \mathbf{T}_0 . Similarly, the ℓ^{th} textual layer representation is computed as: $\mathbf{T}_\ell = \text{Encoder}_\ell^T(\mathbf{T}_{\ell-1})$, $\ell = 1 \dots L_T$, where L_T denotes the number of layers in the textual encoder.

C. Cross-Modal Encoder

We use the transformer encoder [17] with a co-attention mechanism [18] as the cross-modal encoder. In each cross-modal layer, both modalities are equipped with 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. Encoder_ℓ^C , $\ell = 1 \dots L_C$ denotes the ℓ^{th} cross-modal layer, where L_C is the number of cross-modal layers. For brevity, it is computed as:

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

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

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

where $\mathbf{C}_\ell^V, \mathbf{C}_\ell^T$ are the visual and textual part of output representation of the ℓ^{th} layer, $\tilde{\mathbf{C}}_\ell^V, \tilde{\mathbf{C}}_\ell^T$ are inputs of each part. $\mathbf{C}_0^V, \mathbf{C}_0^T$ are initialized with the last-layer representations from unimodal encoders: $\mathbf{C}_0^V = \mathbf{V}_{L_V} \mathbf{W}_V$, $\mathbf{C}_0^T = \mathbf{T}_{L_T} \mathbf{W}_T$, where $\mathbf{W}_V, \mathbf{W}_T$ are linear cross-modal projections. In this work, we use the same default setting as METER and BridgeTower for a fair comparison: pre-trained unimodal encoders with $L_V = L_T = 12$, randomly-initialized cross-modal encoder with $L_C = 6$, and only top $N = 6$ unimodal layer representations are used.

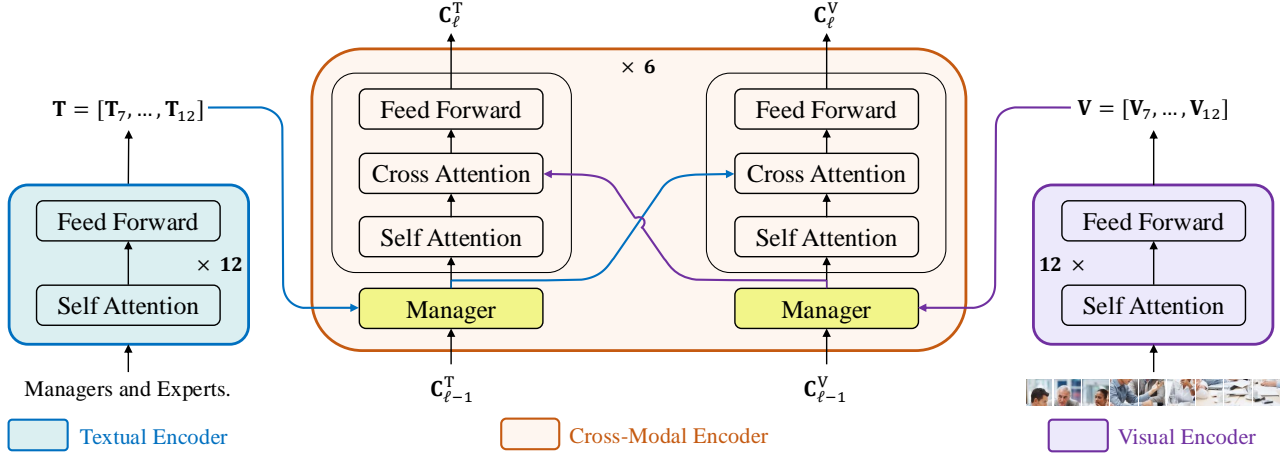


Fig. 3. An illustration of ManagerTower shows that each cross-modal layer includes a textual manager and a visual manager. Top $N=6$ unimodal layer representations $\mathbf{T}, \mathbf{V} \in \mathbb{R}^{N \times L \times D}$ along with the representations from the previous cross-modal layer $\mathbf{C}_{\ell-1}^T, \mathbf{C}_{\ell-1}^V, \ell=1 \dots 6$ are input into the textual manager \mathcal{M}_{ℓ}^T and the visual manager \mathcal{M}_{ℓ}^V , respectively. N refers to the number of pre-trained unimodal experts the model uses, and L denotes the length of the input sequence.

III. MANAGER DESIGN

Fig. 3 illustrates the overall framework of ManagerTower. It introduces managers in each cross-modal layer to aggregate insights from different levels of pre-trained unimodal experts. Under the Two-Tower VLM architecture, we will elaborate on the detailed design schema for the three types of managers, and conclude with the cross-modal encoder with our managers.¹

A. Static Aggregation Manager (SAM)

The effectiveness of layer fusion in learning comprehensive representations has been well demonstrated [10], [19], [20]. Inspired by this, we aim to apply this technique to VLMs. As a preliminary exploration, we adopt the linear combination of layers method [10], which is a simple yet effective way that aggregates the representations of previous layers using learned weights in each encoder layer. We directly adapt it to aggregate both unimodal and cross-modal representations of all previous layers and call it the Static Aggregation Manager (SAM). The calculation for the ℓ^{th} visual manager is given by:

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

where \mathcal{M}_{ℓ}^V represents the manager for the visual part of the ℓ^{th} cross-modal layer, and $\mathbf{W}^{V,\ell} \in \mathbb{R}^{(6+\ell-1) \times D}$ is a learnable parameter matrix. \odot denotes the element-wise product operation, and $\text{LN}(\cdot)$ refers to Layer Normalization [21]. We then omit the superscript V,ℓ of \mathbf{W} for brevity. \mathbf{W} can be seen as the learned aggregation weight and normalized by the softmax function with a learnable temperature. We initialize \mathbf{W} with $\frac{1}{6+\ell-1}$ on average to assign equal weights to the representations from all previous layers.

However, directly applying SAM to VLMs **does not** result in an expected performance improvement over BridgeTower, and instead leads to a notable decrease in performance. We

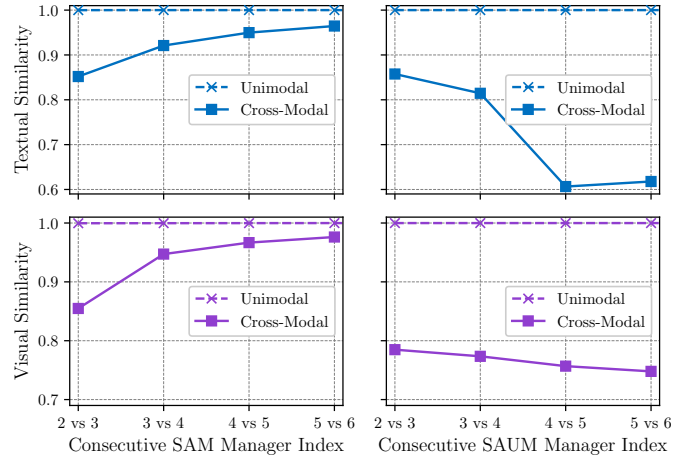


Fig. 4. Cosine similarity between the aggregated unimodal/cross-modal representations of each pair of consecutive textual/visual managers.

hypothesize that this performance drop is due to the average initialization of \mathbf{W} . It may not be suitable for both cross-modal and pre-trained unimodal layer representations as they have **different** numerical scales. To test this hypothesis, we propose dividing the parameter matrix \mathbf{W} into unimodal and cross-modal parts, and initializing them with $\frac{1}{6}$ and $\frac{1}{\ell-1}$, respectively,² and also learn the softmax temperature separately. The experimental result shows a **significant** improvement over the direct application of SAM, though the improvement is still somewhat **limited** compared to BridgeTower. These observations provide a compelling argument for re-examining how to aggregate multi-layer pre-trained unimodal representations.

B. Static Aggregation Unimodal Manager (SAUM)

Since the aggregated representations derived from Equation (4) consist of an unimodal part and a cross-modal

²We also experimented with other initialization methods: one, progressive, exponential moving average, BridgeTower-like one-hot, etc., but the results were similar to or worse than this initialization.

¹Details about pre-training and fine-tuning are described in Appendix B-F.

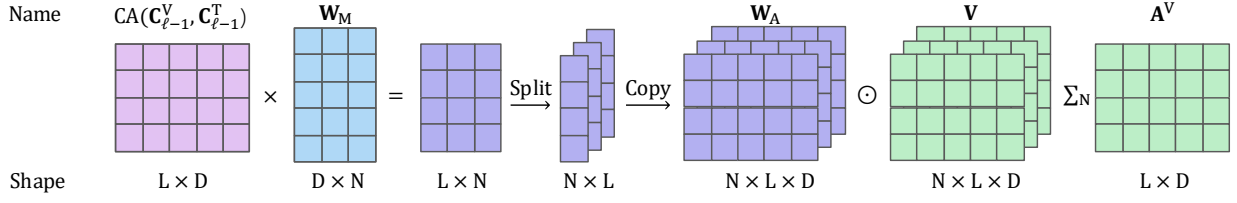


Fig. 5. An illustration of how the aggregated unimodal representations $\mathbf{A}^V \in \mathbb{R}^{L \times D}$ are calculated in the visual AAUM. CA refers to the cross-attention mechanism. $N=6$. For simplicity, we omit LN and softmax function.

part, we calculate the cosine similarity between aggregated unimodal/cross-modal representations of each pair of consecutive textual/visual managers. This can help further analyse insights aggregated in different SAMs, *i.e.*, inputs to different cross-modal layers. As shown in Fig. 4, for SAMs, the unimodal similarity **remains** close to 1, while the cross-modal similarity **increases** with depth and tends toward 1. This suggests that, the unimodal representations aggregated by different SAMs are **nearly identical**, and the aggregated cross-modal representations get more similar with depth.

We hypothesize that, since different SAMs provide **similar** aggregated unimodal representations for each cross-modal layer, the representations from 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.

To address this, we propose focusing on aggregating insights from pre-trained unimodal experts and retaining **only** the representation from the previous cross-modal layer. We refer to it as the Static Aggregation Unimodal Manager (SAUM). The calculation of the ℓ^{th} visual manager computes as:

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

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

The substantial improvement observed compared to BridgeTower provides empirical support for our hypothesis. Furthermore, as shown in Fig. 4, the cross-modal similarity of SAUM **decreases** with the depth, suggesting that more comprehensive and distinguishable cross-modal representations are aggregated as the depth increases.

C. Adaptive Aggregation Unimodal Manager (AAUM)

Despite the significant performance gains achieved by SAUM, it still faces two key limitations: (i) \mathbf{W} , the learned aggregation weight for unimodal representations is nearly **identical** across managers in different cross-modal layers, as demonstrated in Fig. 4, which contradicts the intuition that the requirement for unimodal semantic knowledge should **vary** among cross-modal layers; (ii) during inference, for each manager, the **same** aggregation weight \mathbf{W} learned during training is applied to all tokens in different samples, which

does not align with the intuition that the need for unimodal semantic knowledge should **vary** among tokens and samples.

To address the above limitations, we propose the Adaptive Aggregation Unimodal Manager (AAUM). During training and inference, AAUM can **adaptively** utilize different levels of unimodal semantic knowledge from pre-trained unimodal experts for different tokens across different samples. Take the visual AAUM for example, the ℓ^{th} visual manager computes as:

$$\begin{aligned} \mathcal{M}_\ell^V(\mathbf{V}_7, \dots, \mathbf{V}_{12}, \mathbf{C}_{\ell-1}^V) = & \sum_{i=1}^6 \mathbf{W}_{A,i} \odot \text{LN}(\mathbf{V}_{i+6}) + \mathbf{W}_C \odot \text{LN}(\mathbf{C}_{\ell-1}^V), \quad (6) \\ \mathbf{W}_A = & \text{softmax}(\text{LN}(\mathbf{C}_{\ell-1}^V) \times \mathbf{W}_M + \epsilon), \quad (7) \end{aligned}$$

where $\mathbf{W}_M \in \mathbb{R}^{D \times 6}$ denotes a linear projection layer. The generated aggregation weights $\mathbf{W}_A \in \mathbb{R}^{6 \times L \times D}$ can adaptively aggregate unimodal representations from different levels of pre-trained unimodal experts for each token. The softmax function features a learnable temperature and $\epsilon \sim \mathcal{N}(0, \frac{1}{6^2})$ denotes a Gaussian noise for exploration of aggregation [22].

Furthermore, to help managers better exploit unimodal semantic knowledge, 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.

D. Cross-Modal Encoder with Managers

Since the 1st cross-modal layer lacks the representation of the previous cross-modal layer as the query, we introduce SAUM in the 1st cross-modal layer and AAUMs in the subsequent layers. Therefore, Equation (1) & (2) for the 1st cross-modal layer with SAUMs is computed as:

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

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

For the 2nd and subsequent cross-modal layers with AAUMs:

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

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

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

Fig. 5 shows the adaptive aggregation of insights from pre-trained visual experts in AAUMs, which corresponds to the unimodal (right) part of Equation (6). As for SAUMs, the learned weights $\mathbf{W} \in \mathbb{R}^{6 \times D}$ are directly broadcast to \mathbf{W}_A , and then they aggregate insights similarly to AAUMs.

TABLE I

PERFORMANCE OF DIFFERENT TYPES OF MANAGERS AND QUERIES ON VQAV2 AND FLICKR30K. R_{MEAN} INDICATES THE MEAN RECALL METRICS FOR IMAGE-TEXT RETRIEVAL. BT DENOTES BRIDGETOWER.

Type	Visual Query	Weight	Test-Dev	R _{MEAN}
BT	-	$N \times 1$	75.91	93.33
SAM	-	$N \times 1$	76.19	93.57
	-	$N \times D$	76.18	93.73
SAUM	-	$N \times 1$	76.38	93.75
	-	$N \times D$	76.55	93.82
AAUM	$C_{\ell-1}^V$	$N \times L$	76.52	93.84
	$C_{\ell-1}^V, C_{\ell-1}^T$	$N \times L$	76.65	93.97
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

IV. EXPLORATION ON TWO-TOWER VLM

A. Implementation Details

ManagerTower comprises a pre-trained textual encoder, RoBERTa_{BASE} 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, totaling 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 16×16 . For a fair comparison with BridgeTower, we use an image resolution of 384×384 for Flickr30K and 576×576 for VQAv2. AdamW [23] optimizer with a base learning rate of $2e^{-5}$ and warmup ratio of 0.1 is used.

B. 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 unimodal encoders are initialized with their pre-trained weights.

1) *Type of Manager*: We first explore the performance of different types of managers and queries. Take the 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 insights from pre-trained visual experts.

From the perspective of aggregation weights W_A , SAM and SAUM are **static** sentence-level managers that share the same aggregation weights for all tokens across different samples. In contrast, AAUM is an **adaptive** token-level manager that adaptively **generates** different aggregation weights for different tokens across different samples. Besides, we also implement Equation (7) with common cross- and concat-attention mechanisms for comparison, detailed in Algorithm B-D.

The results are summarized in Table I. By focusing on aggregating insights from pre-trained unimodal experts, SAUM demonstrates **superior** performance over SAM on both datasets. Furthermore, with the help of the cross-modal fused query, AAUM **significantly** outperforms the other managers. This highlights the effectiveness of **adaptive** token-

TABLE II

PERFORMANCE OF BRIDGETOWER (BT) AND MANAGER TOWER (OURS) WITH DIFFERENT NUMBERS OF CROSS-MODAL LAYERS.

L_C	VQAv2 Test-Dev		Flickr30K R _{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	76.65 ($\uparrow 0.74$)	93.33	93.97 ($\uparrow 0.64$)
8	75.89	76.47 ($\uparrow 0.58$)	93.03	93.65 ($\uparrow 0.62$)

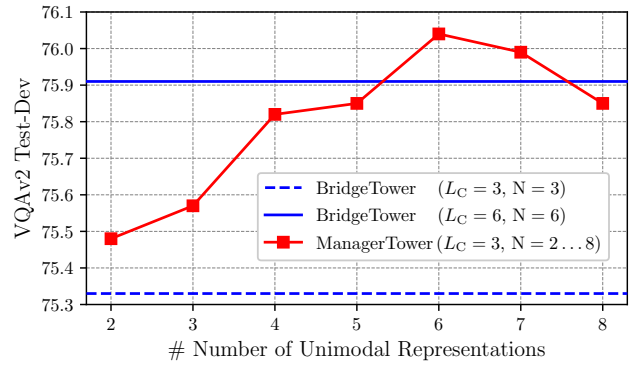


Fig. 6. VQAv2 Test-Dev Performance using different numbers of unimodal representations in ManagerTower ($L_C=3, N=2 \dots 8$).

level aggregation with the cross-modal fused query over **static** sentence-level aggregation. Notably, the cross-modal fused query incorporates both visual and textual parts of the previous cross-modal layer representation, which can better help managers correctly aggregate unimodal semantic knowledge required by the current cross-modal layer.

2) *Number of Cross-Modal Layers*: We conduct a comparison between ManagerTower and BridgeTower with different numbers of cross-modal layers in Table II, to further assess the effectiveness of ManagerTower. Regardless of the number of cross-modal layers, ManagerTower **consistently** and **significantly** outperforms BridgeTower on both datasets. More interestingly, the performance of ManagerTower with $L_C=3$ is even better than that of BridgeTower with $L_C=6$ ($76.04 > 75.91$, $93.41 > 93.33$).

In contrast to BridgeTower, N , the number of top unimodal layer representations used by ManagerTower, is not bound to the number of cross-modal layers L_C and can be flexibly adjusted. The default setting is $N=6$. Therefore, ManagerTower actually utilizes the same number of unimodal layer representations as BridgeTower, but achieves **superior** performance with **only half** the number of cross-modal layers. This further highlights the **flexibility** and **effectiveness** of ManagerTower in adaptive aggregation of unimodal semantic knowledge, in contrast to layer-by-layer exploitation in BridgeTower.

3) *Number of Unimodal Experts*: We further explore the impact of varying N in ManagerTower with $L_C=3$. As shown in Fig. 6, there exist two interesting observations: (i) ManagerTower ($L_C=3, N=3$) outperforms BridgeTower ($L_C=3, N=3$), suggesting that when the same number of unimodal layer representations are introduced, Manager-

TABLE III
COMPARISONS WITH PREVIOUS MODELS ON 4 DOWNSTREAM DATASETS AFTER VLP. THE BEST SCORE IS BOLDED. * INDICATES THAT THE MODEL ALSO USES VG-QA DATA TO FINE-TUNE ON VQAV2.

Model	# Pre-train Images	VQAv2		SNLI-VE		NLVR ²		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 _{BASE} [24]	4M	71.26	-	-	-	75.70	76.13	64.4	83.5
UNITER _{BASE} [25] *	4M	72.70	72.91	78.59	78.28	77.18	77.85	72.52	85.90
UNIMO _{BASE} [26]	4M	73.79	74.02	80.00	79.10	-	-	74.66	89.70
ALBEF _{BASE} [27] *	4M	74.54	74.70	80.14	80.30	80.24	80.50	82.8	94.3
METER-Swin _{BASE} [6]	4M	76.43	76.42	80.61	80.45	82.23	82.47	79.02	92.40
VLMO _{BASE} [28]	4M	76.64	76.89	-	-	82.77	83.34	79.3	92.3
METER-CLIP _{BASE} [6]	4M	77.68	77.64	80.86	81.19	82.33	83.05	82.22	94.30
BridgeTower _{BASE} [7]	4M	78.66	78.73	81.11	81.19	81.85	83.09	85.83	94.73
ManagerTower _{BASE} (Ours)	4M	79.39	79.15	81.26	81.44	82.81	83.34	86.56	95.64
<i>Models pre-trained on more data and/or with larger size</i>									
UNITER _{LARGE} [25] *	4M	73.82	74.02	79.39	79.38	79.12	79.98	75.56	87.30
UNIMO _{LARGE} [26]	4M	75.06	75.27	81.11	80.63	-	-	78.04	89.40
ALBEF _{BASE} [27] *	14M	75.84	76.04	80.80	80.91	82.55	83.14	85.6	95.9
SimVLM _{BASE} [29]	1.8B	77.87	78.14	84.20	84.15	81.72	81.77	-	-
BLIP _{BASE} [30] *	129M	78.24	78.17	-	-	82.48	83.08	87.3	97.3
SimVLM _{LARGE} [29]	1.8B	79.32	79.56	85.68	85.62	84.13	84.84	-	-

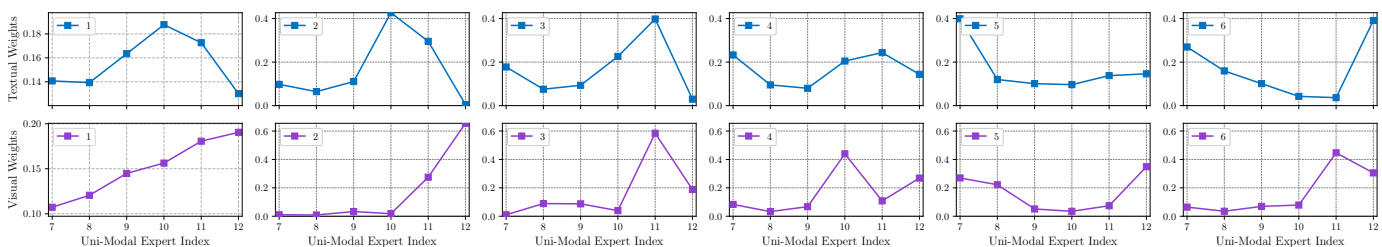


Fig. 7. A visualization of aggregation weights of textual and visual AAUMs in each cross-modal layer after VLP. The X-axis shows the index of the unimodal expert, and the legend shows the index of the cross-modal layer.

Tower allows more **effective** aggregation of unimodal semantic knowledge, thus facilitating vision–language alignment and fusion in each cross-modal layer; (ii) the performance of ManagerTower initially improves gradually, but decreases after $N > 6$. We assume that lower-layer unimodal representations **may not** help ManagerTower learn vision–language alignment and fusion, and may also increase the computational cost. This is also consistent with BridgeTower’s observations.

C. Comparison with Previous Arts

1) *Pre-train Settings*: We pre-train ManagerTower with two standard VLP objectives, masked language modeling (MLM) and image–text matching (ITM), on the widely-used 4M public data: Conceptual Captions [31], SBU Captions [32], MSCOCO Captions [33], and Visual Genome (VG) [34]. 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 288×288 and only center-crop [8] is used without any data augmentation.

2) *Main Results*: Table III shows the performance of ManagerTower compared with other previous works on 4 downstream datasets. With only 4M VLP data, ManagerTower

achieves **superior** performances on these datasets. Based on the same pre-training and fine-tuning settings and unimodal backbones as previous strong Two-Tower VLMs, *i.e.*, METER and BridgeTower, ManagerTower achieves **significant** improvements on all datasets, 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, managers in ManagerTower allow **effective** aggregation of multi-layer unimodal representations via well-designed managers. Managers can **adaptively** aggregate more required unimodal semantic knowledge to facilitate comprehensive vision–language 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.

D. Visualization of Aggregation Weights

We delve into managers by visualizing the average aggregation weights \mathbf{W}_A they generate across all samples in VQAv2 validation set in each cross-modal layer in Fig. 7. For each row, the first column displays the learned aggregation weights of SAUMs, while the remaining five columns show

the aggregation weights generated by AAUMs and share the Y-axis to provide easy horizontal comparison.

Interestingly, the aggregation weight distributions from managers are **completely different** from the one-hot distributions manually specified in BridgeTower, and there are two distinct trends: (i) For SAUMs in the 1st cross-modal layer, vertically, textual manager exhibits increasing and then decreasing weights, most favoring \mathbf{T}_{10} , unlike \mathbf{T}_{12} and \mathbf{T}_7 used in METER and BridgeTower, respectively; visual manager exhibits increasing weights, most favoring \mathbf{V}_{12} , similar to METER and BridgeTower. (ii) For AAUMs in the 2nd to 6th cross-modal layers, horizontally, whether textual or visual managers, they exhibit **diverse** aggregation weight distributions in different layers.

Overall, by comparing the aggregation weight distributions horizontally and vertically, we observe that ManagerTower learns **diverse** distributions in different cross-modal layers. This provides strong evidence that the introduced managers can **adaptively** aggregate unimodal semantic knowledge for more comprehensive vision–language representation learning.

V. EXPLORATION ON MLLM

A. Motivation

As stated in Sec. I, in principle, the manager is a lightweight and flexible plugin that can be easily integrated into various VLMs. Naturally, we can take the manager as a plugin and further explore its effectiveness in the latest MLLM architecture, which typically consists of a visual encoder and an LLM.

Moreover, traditional Two-Tower VLMs and MLLMs both use ViTs as their visual encoder, which have to resize the input image to a fixed resolution. This greatly **limits** their effectiveness in handling high-resolution images due to the **loss of visual details**. Recent multi-grid MLLMs [11], [35], [36] overcome this limitation by training with the multi-grid algorithm.³ During training and inference, they divide the padded input image into multiple image grids, and encode both the resized base image and multiple image grids with the visual encoder independently. Then, they combine the encoded features to obtain a longer input visual representation with more visual details.

Compared the manager with the multi-grid algorithm, they both can be seen as a **plugin** that improves the input visual representation and thus improves the VL representation. They are two **orthogonal** directions to supplement visual details, either by (i) **deeper**: introducing aggregation of insights from pre-trained visual experts at different levels/depths; or (ii) **wider**: directly improving image resolution by encoding multiple image grids, *i.e.*, a wider receptive field. Hence, we are motivated to explore the effectiveness of managers not only in MLLMs, but also in multi-grid MLLMs, to investigate the **synergy** between the manager and the multi-grid algorithm.

Besides, with the help of the MLLM architecture and the multi-grid algorithm, we can further **extend** downstream datasets, not only limited to traditional general datasets with low-resolution natural images, *e.g.*, VQAv2 and Flickr30K

used in Sec. IV, but also text-rich datasets with high-resolution abstract images (documents, charts, *etc.*), *e.g.*, DocVQA [37] and OCRBench [38], and real-world multimodal datasets. Without fine-tuning on specific datasets, we can provide more **comprehensive** and **challenging** zero-shot evaluations of the effectiveness of managers.

Overall, we aim to explore the effectiveness of managers in more diverse downstream datasets, to answer the questions: **(RQ1)** Can the manager be used as a plugin to help MLLMs and multi-grid MLLMs? **(RQ2)** When and why can managers improve performance, especially for multi-grid MLLMs?

B. Experimental Settings

1) *Baseline*: We take LLaVA-OneVision-0.5B-SI [11] as our baseline (LLaVA-OV for short), which is a widely used open-source multi-grid MLLM. It consists of a pre-trained 27-layer visual encoder SigLIP [39] with 0.4B parameters, a pre-trained 24-layer LLM Qwen2-0.5B-Instruct [40] with 0.5B parameters and a 2-layer MLP with 1.8M parameters. It releases most of the training data, which helps us reproduce not only the multi-grid version (Baseline+Grid), but also the plain version (Baseline). We follow the same training settings as the original LLaVA-OV and use about 8M data samples for multi-stage training of the autoregressive objective for answer tokens. The maximum length of the input token sequence is set to 16384, and the image patch size is 14×14. The last layer of the visual encoder is removed, and the visual representation of the penultimate layer is projected into the LLM word embedding space as the visual part of the input tokens of the LLM. More details can be found in Appendix C-E.

2) *Adapt Manager to MLLM*: Since the LLM in MLLM acts as both a textual module and a cross-modal module, as shown in Fig. 2, we **directly** introduce visual managers in LLaVA-OV, to aggregate multi-layer visual representations and inject them into the LLM at **equal** intervals, thus obtaining LLaVA-OV-Manager. Similar to LLaVA-OV, we train two versions of LLaVA-OV-Manager and name them as Baseline+Manager and Baseline+Grid+Manager, respectively. Managers aggregate insights from the top half of the visual encoder to improve the visual representations of both the base image and image grids independently. We inject 6 visual managers into the LLM with the interval of 4 as the default setting. Since AAUM achieves similar performance compared to SAUM in LLaVA-OV-Manager, we directly use SAUM for better efficiency in the following experiments.⁴ For brevity, the ℓ^{th} LLM layer with SAUM computes as:

$$\tilde{\mathbf{C}}_{\ell}^{\mathbf{V}} = \mathcal{M}_{\ell}^{\mathbf{V}}(\mathbf{V}_{14}, \dots, \mathbf{V}_{26}) \odot \epsilon + \mathbf{C}_{\ell-1}^{\mathbf{V}}, \quad (12)$$

$$\mathbf{C}_{\ell}^{\mathbf{V}}, \mathbf{C}_{\ell}^{\mathbf{T}} = \text{Encoder}_{\ell}^{\mathbf{C}}(\tilde{\mathbf{C}}_{\ell}^{\mathbf{V}}, \mathbf{C}_{\ell-1}^{\mathbf{T}}), \quad (13)$$

$$\mathcal{M}_{\ell}^{\mathbf{V}}(\mathbf{V}_{14}, \dots, \mathbf{V}_{26}, \mathbf{C}_{\ell-1}^{\mathbf{V}}) = \sum_{i=1}^{13} \mathbf{W}_i \odot \mathbf{V}_{i+13}. \quad (14)$$

Equation (14) is an optimized version of SAUM for MLLM. The original version does not work well in our preliminary experiments, as the LLM in MLLM has been well pre-trained, rather than the random-initialized cross-modal module

³An illustration of the multi-grid algorithm can be found in Appendix C-A.

⁴Discussions about managers in the MLLM can be found in Appendix A-A.

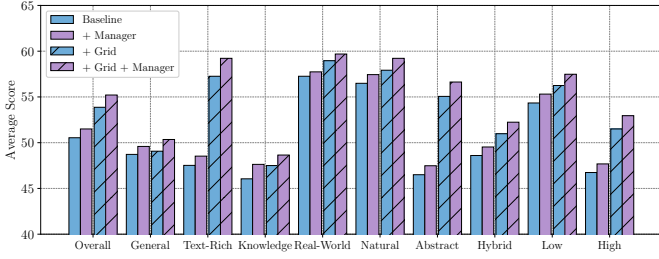


Fig. 8. Zero-shot performance of four baselines on 20 datasets. The overall average score and the average score of each capability category are shown.

in ManagerTower. Hence, we remove the \mathbf{W}_C , LN, and softmax in Equation (5), and initialize \mathbf{W} to zero, to reduce the interference with the pre-trained LLM in the early training stage [41], [42], which helps SAUM work well in MLLM. $\epsilon \sim \mathcal{U}(0.98, 1.02)$ is a multiplicative jitter noise uniformly sampled for exploration across experts during training [22].

3) *Evaluation*: We follow the same evaluation settings as the original LLaVA-OV, to evaluate the zero-shot performance of our four baselines on 20 datasets via their official evaluation tool, lmms-eval.⁵ From the perspective of **capability categories**, we can divide them into the following four categories:

- General: VQAv2 [2], OKVQA [43], GQA [44], MMVet [45], SEED-Bench [46], RealWorldQA [47].
- Text-rich: TextVQA [48], ChartQA [49], DocVQA [37], InfoVQA [50], OCRBench [38].
- Knowledge: AI2D [51], ScienceQA [52], MMMU [53], MathVista [54].
- Real-world: ImageDC [55], MM-LiveBench (07, 09) [56], LLaVA-Wild [57], LLaVA-Wilder [35].

For simplicity, we use the average score of the corresponding metric score (normalize to [0, 100]) as the overall performance of baselines. We also calculate the average score of each capability category for in-depth analysis. Furthermore, since these datasets contain not only low-resolution natural images, but also high-resolution abstract images, we can also analyse and divide these datasets from the perspective of **image categories** “Natural, Abstract, Hybrid” and **resolutions** “Low, High”.⁶

C. Results and Computational Budget

Fig. 8 shows the zero-shot performance of four baselines on 20 datasets after training with about 8M data samples following the original LLaVA-OV.⁷ The difference between baselines is with or without the multi-grid algorithm and managers. Similar to existing multi-grid MLLMs, we can observe that the multi-grid algorithm greatly helps Baseline and Baseline+Manager, especially on text-rich datasets, abstract images, and high-resolution images. When introducing managers, whether the multi-grid algorithm is enabled or not, the performance of Baseline+Manager and Baseline+Grid+Manager is **significantly** improved over the corresponding Baseline and Baseline+Grid on different categories of capabilities, images, and resolutions. Especially on datasets with capability

⁵<https://github.com/EvolvingLLMs-Lab/lmms-eval>

⁶More evaluation details can be found in Appendix C-F.

⁷Detailed results of each dataset can be found in Appendix ??.

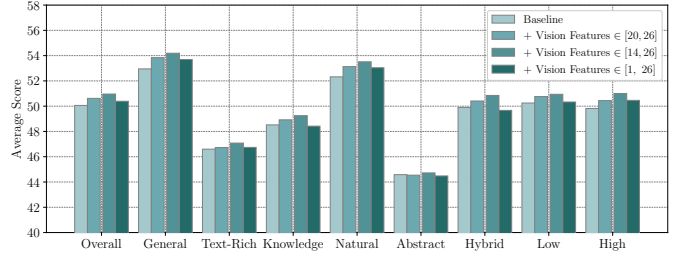


Fig. 9. Ablation study of visual representation selection on 9 datasets.

TABLE IV
COMPUTATIONAL BUDGET AND AVERAGE OVERALL PERFORMANCE OF FOUR BASELINES ON 20 DATASETS. THE NUMBERS IN PARENTHESES DENOTE THE RELATIVE CHANGE COMPARED TO BASELINE.

Model	# Params (M)	Training Time (ms/sample)	Inference Time (ms/sample)	Performance Overall
Baseline	893.62	11.84	13.97	50.61
+ Manager	893.70	12.22 ($\times 1.03$)	14.54 ($\times 1.04$)	51.67 ($\uparrow 1.06$)
+ Grid	893.62	51.95 ($\times 4.39$)	23.47 ($\times 1.68$)	53.87 ($\uparrow 3.26$)
+ Grid + Manager	893.70	54.17 ($\times 4.58$)	24.45 ($\times 1.75$)	55.21 ($\uparrow 4.60$)

category of “General, Knowledge”, Baseline+Manager even achieves better performance than Baseline+Grid with significantly lower computational cost.

Table IV shows the computational budget of baselines. We measure the average training time based on two $8 \times$ NVIDIA A100 GPU servers, and the average inference time on VQAv2 validation set with a single A100 GPU. Compare to Baseline, the multi-grid algorithm significantly increases the training time ($\times 4.39$), inference time ($\times 1.68$) and performance ($\uparrow 3.26$). No matter with or without the multi-grid algorithm, managers only brings **negligible** parameter overhead (0.08M) and computational cost ($\times 1.04$), but **significantly** improves performance ($\uparrow 1.06$ and $\uparrow 1.44$) on 20 datasets.⁸

In summary, for our **RQ1**, Fig. 8 and Tab. IV demonstrate that the manager is a **lightweight, efficient and effective** plugin that helps MLLMs and multi-grid MLLMs achieve **better** performance in different capability categories, image categories and resolutions, with **acceptable** computational cost. More interestingly, the collaboration between managers and the multi-grid algorithm not only supplements **visual details** from the **depth** and **width** directions, respectively, to improve performance, but also further boosts performance by their synergy ($1.44 > 1.06$).

D. Ablation Study on Adaptation of Managers in MLLMs

In this section, we further explore the adaptation of managers in MLLMs. We use $\frac{1}{4}$ of the training data (2M samples) and evaluate on 9 datasets for efficiency and robustness.

1) *Visual Representation Selection*: As shown in Fig. 9, overall, no matter what visual representations are selected, managers **consistently** improve the performance of Baseline. Similar to the observations in both BridgeTower and ManagerTower, visual representations from the top half of the visual encoder bring the best performance, and using visual

⁸ $54.17/51.95 \approx 1.04$, $24.45/23.47 \approx 1.04$ and $55.21 - 51.67 = 1.44$.

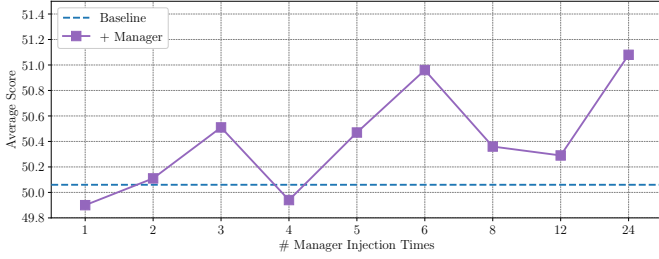


Fig. 10. Ablation study of manager injection times on 9 datasets.

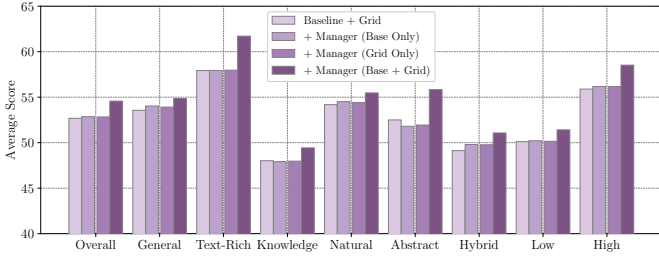


Fig. 11. Ablation study of how manager works with multi-grid on 9 datasets.

representations from all layers leads to the lowest performance improvement. We attribute this to the fact that the average attention distance of the visual encoder increases with the layer depth, especially in the top half of the visual encoder, where most attention heads attend **widely** across tokens [58] and capture global visual features.⁹

2) *Manager Injection Times*: We uniformly inject managers into the LLM from the first layer at a fixed layer interval. Specifically, for the LLM with $L_C = 24$, we can inject 6 managers with the interval of 4. As shown in Fig. 10, the injection times of managers will affect the performance, and the overall trend is that performance improves with increasing injection frequency, but with some fluctuations. Baseline+Manager can achieve **better** performance than Baseline most of the time. Compared to the injection times of 6, although injecting managers into each LLM layer slightly increases the average performance from 50.96 to 51.08, it also **increases** the computational cost by about 7% in both training and inference. Hence, we choose the injection times of 6 to achieve a good **balance** between performance and computational cost.

3) *Manager Meets Multi-Grid*: Both the manager and the multi-grid algorithm are plugins that can be easily combined and integrated into MLLMs. Their direct combination means that managers aggregate insights from pre-trained visual experts at different levels to improve the visual representations of the base image and multiple image grids, respectively. As shown in Fig. 11, managers **greatly** improve the performance of Baseline+Grid, especially on text-rich datasets, abstract images, and high-resolution images, which are exactly what the multi-grid algorithm excels at. This indicates that the manager and the multi-grid algorithm are orthogonal (depth and width) and **complementary** in complementing visual details, and their synergy can further improve performance. More interestingly, when managers only manage the base image or

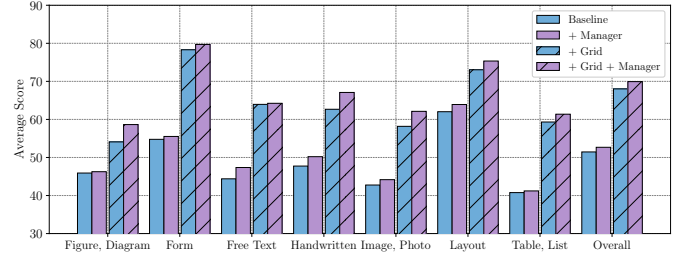


Fig. 12. Zero-shot performance of four baselines on DocVQA validation set.

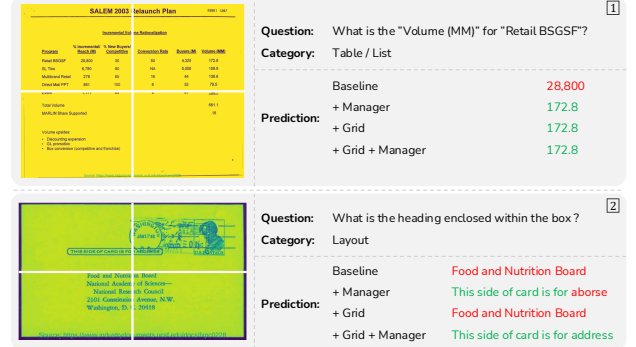


Fig. 13. Case studies of four baselines on DocVQA validation set. Red and green fonts represent incorrect and correct predictions, respectively. White lines indicate the boundaries of the image grids.

image grids, the performance is not obviously improved. We speculate that the change in part of the visual representation by managers may be considered as **noise** due to the numerical difference between the changed and unchanged parts.

E. Detailed Analysis and Case Study

To intuitively analyse the effectiveness of managers and answer our **RQ2**, we conduct a detailed analysis on different dimensions of specific datasets, including DocVQA, SEED-Bench, and OCRBench, and provide case studies.¹⁰

1) *DocVQA*: Based on the three dataset classification criterion we used in Section V-B3, DocVQA is a text-rich dataset with high-resolution abstract images. As shown in Fig. 12, the multi-grid algorithm helps Baseline on different types of abstract images in DocVQA. Furthermore, managers can further improve the performance of Baseline and Baseline+Grid on different dimensions. Take the case [1] in Fig. 13 as an example, both managers and the multi-grid algorithm can help Baseline **capture** visual details for accurate table understanding. Interestingly, in the case [2], both Baseline and Baseline+Grid fail to find the heading enclosed within the box, and take the first line of text below the box as the heading. The multi-grid algorithm also **cuts off** the boxed heading, may make it more difficult to find the heading. Baseline+Manager can **correctly** find it based on the visual details provided by different levels of semantic knowledge, but fails to recognize all characters. With the **collaboration** between the manager and the multi-grid algorithm, Baseline+Grid+Manager can correctly find it and recognize all characters.

¹⁰More detailed analysis and case studies on ScienceQA and OK-VQA can be found in Appendix C-C.

⁹Detailed explanations and visualizations are provided in Appendix C-B

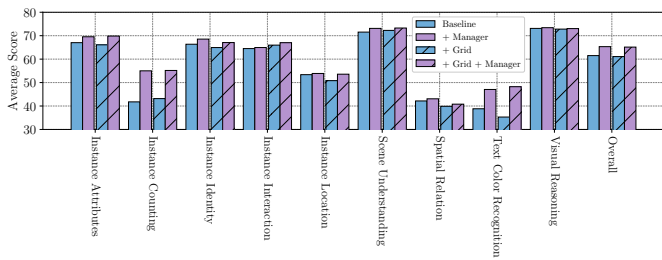


Fig. 14. Zero-shot performance of four baselines on SEED-Bench.

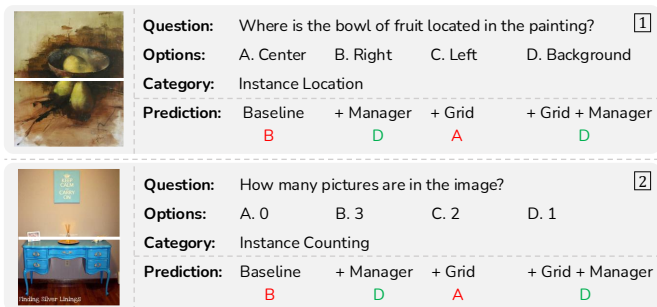


Fig. 15. Case studies of four baselines on SEED-Bench.

2) *SEED-Bench*: This is a general dataset with high-resolution natural images. Surprisingly, as shown in Fig. 14, the multi-grid algorithm does not improve the performance much and even leads to performance **degradation** on some dimensions, *i.e.*, “Instance Identity, Instance Location, Spatial Relation, Text Color Recognition”. They inspect the category, spatial and color information about instances in the image. Take Fig. 15 as an example, the multi-grid algorithm **cuts off** objects and connected regions, leading to higher understanding difficulty and bringing **semantic ambiguity** [59]. This **hinder** MLLMs from perceiving the spatial relationship between objects as well as the category and number of objects. Moreover, managers **consistently** brings performance improvements to Baseline and also help **overcome** the semantic ambiguity caused by the multi-grid algorithm by incorporating aggregation of **insights** from pre-trained visual experts at different levels, especially on “Instance Counting, Text Color Recognition”.

3) *OCRBench*: This is a text-rich dataset with low-resolution hybrid images. As shown in Fig. 16, for “Artistic Text Recognition, Handwriting Recognition” dimensions, both the manager and the multi-grid algorithm can only bring slight performance improvements or even performance degradation to Baseline. However, the collaboration between them can bring **significant** performance improvements on Baseline+Grid+Manager. This further demonstrates that their **synergy** can complement **visual details** from the depth and width directions and mitigate the semantic ambiguity caused by the multi-grid algorithm. Unexpectedly, for “Non-Semantic Text Recognition” dimension, which focuses on character combinations that lack semantics., the manager brings performance degradation to both baselines. Take the cases in Fig. 17 as an example, although managers can help capture visual details, *e.g.*, a single quote at the end of the word, Base-

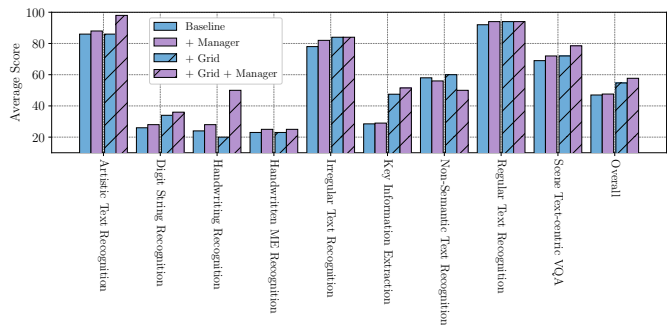


Fig. 16. Zero-shot performance of four baselines on OCRBench. “ME” in “Handwritten ME Recognition” is short for “Mathematical Expression”.



Fig. 17. Case studies of four baselines on OCRBench.

line+Grid+Manager incorrectly identifies the **non-semantic** text “wenar” and “ttrebe” as semantic text “wenar” and “trebe”,¹¹ respectively. Different levels of semantic knowledge brought by managers instead cause more interference, leading to performance degradation.

In summary, for our **RQ2**, the manager can not only **improve** the performance of MLLMs, but also help **alleviate** the semantic ambiguity caused by the multi-grid algorithm. Hence, their **synergy** can further improve performance, especially on the perception of category, spatial, color and number information of instances, and artistic, handwriting text recognition.

F. Visualization Analysis

To analyse the underlying reasons for the collaboration improvement between the manager and the multi-grid algorithm in MLLMs and further answer our **RQ2**, we conduct analyses from the perspective of consecutive layer representation similarity and attention weight distribution of each layer.

1) *Consecutive Layer Representation Analysis*: In Equation (13), the output representation of each LLM layer consists of a visual part and a textual part. For each part, we calculate the cosine similarity between output representations of consecutive layers in Baseline+Grid and Baseline+Grid+Manager. As shown in Fig. 18, managers **reduce** the similarity between representations of consecutive layers, especially for the **bottom** layers of MLLMs. Compare to Baseline+Grid, changes in the similarity become more frequent and drastic in the layers between manager **injections**. This indicates that the aggregation of different levels of semantic knowledge introduced by managers can **supplement** more insights and visual details, and facilitate more **diverse** vision–language representation learning in subsequent layers. It is worth noting that although we do not have textual managers, the textual

¹¹“wenar”: a surname of a person. “trebe”: a German noun for a runaway.

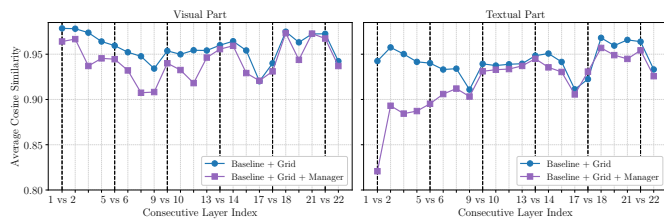


Fig. 18. Cosine similarity between output representations of consecutive layers. The dotted vertical lines indicate the layers where managers are injected, *i.e.*, # Layer Index = [1, 5, 9, 13, 17, 21].

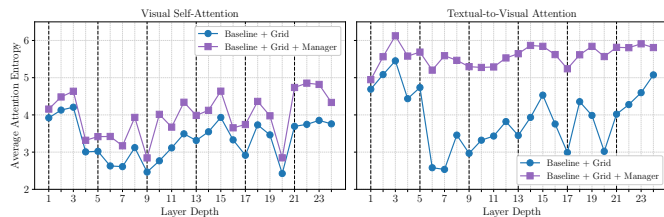


Fig. 19. Average entropy of attention weight distributions in each layer.

part of the output representation is **causally** influenced by the visual part in its front, resulting in a similar phenomenon.

2) *Attention Weight Distribution Analysis*: The attention mechanism [60] is a key component in deep neural networks, where attention weight distributions reflect how much attention each token pays to the other tokens. Following [61], we delve into attention weight distributions from the following two angles to provide an intuitive and interpretable analysis. Besides, for the attention weight distribution of each layer, we focus on the self-attention of the visual part, and the attention from the textual part at the back to the visual part at the front.¹²

a) *Attention Entropy*: The average entropy of attention weight distributions reflects the **diversity** of attention weights in each layer. Higher/lower attention entropy means that the attention weights are concentrated on **more/few** tokens. As shown in Fig. 19, compared to Baseline+Grid, managers **increase** the attention entropy in each layer. Such **broad** attention can help Baseline+Grid+Manager handle more complex and varied input, leading to greater diversity and flexibility, and thereby preventing focusing too narrowly on certain aspects of the input. Besides, interestingly, the entropy of textual-to-visual attention becomes more stable and significantly larger than the entropy of visual self-attention when managers manage the visual part of the input.

b) *KL Divergence*: The average Kullback–Leibler (KL) divergence [62] between attention weight distributions of different attention heads reflects the **diversity** of attention heads in each layer. Higher/lower KL divergence means that different attention heads pay attention to **different/similar** tokens. As shown in Fig. 20, compared to Baseline+Grid, managers **increase** the KL divergence between attention heads in most layers. Intuitively, low diversity across different attention heads may limit the model’s ability to capture diverse features. Managers can help Baseline+Grid+Manager focus on different

¹²Attention weight distribution analysis of Baseline and Baseline+Manager can be found in Appendix C-D.

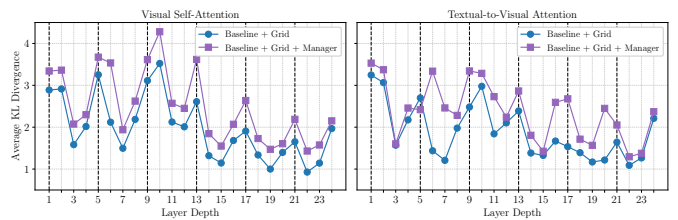


Fig. 20. Average KL divergence between attention weight distributions of attention heads in each layer.

aspects of the sequence to capture more **diverse** features, and prevent excessive focus on similar or redundant information.

In summary, for our **RQ2**, the manager introduces the aggregation of **insights** from visual experts at different levels into multi-grid MLLMs, which can **increase the diversity** of attention weights and attention heads. This can help **guide** the attention of multi-grid MLLMs, thus capturing more diverse visual details from both the manager (**depth**) and the multi-grid algorithm (**width**) directions, and also alleviating the semantic ambiguity caused by the multi-grid algorithm.

VI. RELATED WORK

A. Vision–Language Models

Although VLMs differ in model architecture, most of them use unimodal encoders to extract visual and textual representations, and then fuse them in a cross-modal module, which can be unified into the Two-Tower architecture [6], [8], [18], [24]–[30], [63]–[70]. As a representative model, METER [6] adopts pre-trained unimodal encoders and feeds their last-layer representations into the cross-modal encoder with the co-attention mechanism. BridgeTower [7] proposes building layer-by-layer connections between the top unimodal layers and each cross-modal layer to leverage multi-layer unimodal representations. However, they still cannot utilize adaptive and effective aggregation of multi-layer pre-trained unimodal representations in each cross-modal layer.

B. Utilization of Multi-Layer Unimodal Representations

Different layers of pre-trained unimodal encoders encoding different levels of semantic knowledge are well demonstrated in vision [58], [71], [72] and language [73]–[75]. As shown in prior work [58], [71], lower layers of ViTs tend to attend both locally and globally, while higher layers primarily focus on global features. Similarly, previous work [75] found that the intermediate layers of BERT [76] encode a hierarchy of linguistic knowledge, with surface features at the bottom, syntactic features in the middle, and semantic features at the top.

Furthermore, the effectiveness of multi-layer representation aggregation in learning comprehensive representations has been well demonstrated in vision [77]–[83] and language [10], [19], [20], [84]. Hence, some Two-Tower VLMs and MLLMs have explored the utilization of pre-trained multi-layer unimodal representations for better vision–language representation learning [6], [7], [85]–[87]. They simply feed the weighted sum or fusion of multi-layer unimodal representations into the first cross-modal layer, or exploit multiple top unimodal layer

representations layer by layer in each cross-modal layer, which is not only ineffective but also lack scalability. In this work, we take each layer of the pre-trained unimodal encoder as an unimodal **expert**, and the output representation of each layer as the **insight** of the unimodal expert into the current input. We propose managers to **adaptively** aggregate insights from unimodal experts at different levels for each cross-modal layer.

C. Multimodal Large Language Models

With the rapid development of Large Language Models (LLMs) [40], [88]–[90], MLLMs, a new class of VLMs that introduces a LLM as both a textual module and a cross-modal module, have emerged and shown superior zero-shot performance on various downstream tasks [11], [35], [91]. Although most existing MLLMs only feed the last-layer visual representation from the visual encoder into the LLM for simplicity and efficiency, some of them have explored different ways to improve the visual representation to further improve performance, especially high-resolution scenarios, such as: (i) adopt high-resolution visual encoders [92]–[95], which require additional high-resolution training data; (ii) adopt the multi-grid algorithm to directly split the image into multiple image grids [12], [36], [96], [97], which is a resource-efficient way but may bring semantic ambiguity [59], [98]. Since both the manager and the multi-grid algorithm can be viewed as a plugin that improves the visual representation from two orthogonal perspectives (**depth** and **width**), we further explore the effectiveness of managers in MLLMs and multi-grid MLLMs and the underlying reasons for their **collaboration** to improve performance based on extensive experiments and detailed analyses.

VII. CONCLUSION

In this work, we propose Manager, a **lightweight, efficient** and **effective** plugin that helps better utilize multi-layer pre-trained unimodal representations for vision–language representation learning, and demonstrate its effectiveness in both Two-Tower VLM and MLLM architectures. The manager can **adaptively** aggregate more required unimodal semantic knowledge to facilitate comprehensive vision–language alignment and fusion in each cross-modal layer. We first propose ManagerTower, a novel Two-Tower VLM that aggregates **insights** from pre-trained unimodal experts at different levels via introduced managers in each cross-modal layer. The feasibility of various designs of managers is well explored, and the effectiveness of ManagerTower on 4 downstream tasks is well demonstrated. Next, we further validate the effectiveness of managers in the latest MLLM architecture. Managers can **significantly** improve the zero-shot performance of MLLMs and multi-grid MLLMs on 20 downstream datasets across different categories of capabilities, images, and resolutions. Both the manager and the multi-grid algorithm can be seen as a **plugin** that improves the visual representation from two orthogonal perspectives (**depth** and **width**). Their synergy can capture and supplement more **diverse** visual details, to mitigate the semantic ambiguity caused by the multi-grid algorithm and further improve performance.

APPENDIX A

DISCUSSIONS, LIMITATIONS AND FUTURE WORK

In this paper, we propose Manager, a **lightweight, efficient** and **effective** plugin that helps better utilize multi-layer pre-trained unimodal representations for vision–language representation learning. We demonstrate its effectiveness in both Two-Tower VLM and MLLM architectures on 4 and 20 downstream datasets, respectively.

A. *The Spirit of Manager*

In Section III under the Two-Tower VLM architecture, we introduce three types of managers, *i.e.*, SAM, SAUM, and AAUM. We also provide an adaptation of SAUM to the latest MLLM architecture in Section V-B2. All managers are designed to aggregate insights from different levels of pre-trained unimodal experts, *i.e.*, incorporate different levels of unimodal semantic knowledge contained in the multi-layer unimodal representations from the pre-trained unimodal encoders. The detailed implementation of these managers obey the following design principles:

- **Lightweight and Flexible:** The manager should be a lightweight and flexible plugin that can be easily integrated into any VLM architecture and work with any pre-trained unimodal encoders.
- **Effective and Efficient:** The manager should be effective in aggregating multi-layer unimodal representations and efficient in terms of computational budget.
- **Aggregation and Multiple Injection:** The manager should aggregate multi-layer unimodal representations and inject them into the different layers of the cross-modal encoder in a flexible and adaptive way to facilitate more comprehensive VL alignment and fusion.
- **Spatial Consistency:** The manager should maintain spatial consistency (or locality) when aggregate multi-layer unimodal representations, which is crucial for vision–language alignment and fusion [99].

It would be interesting to explore more types of managers, more efficient and effective designs of managers, and more flexible and adaptive ways to select which layers of unimodal representations to aggregate and how to inject them into different layers of the cross-modal encoder or the LLM in the future. inject managers into VLMs in the future:

- **Manager Design:** Although AAUM achieves better performance under the Two-Tower VLM architecture with the help of the cross-modal fused query, it also slightly increases the computational budget, as we detailed discussed in Appendix B-C. More analysis and optimization are needed for AAUM and also for the other types of managers as shown in Appendix B-D. Especially for AAUM, in the MLLM architecture, it actually achieves similar or even slightly lower performance compared to SAUM in LLaVA-OV-Manager. Hence, we directly use SAUM for better efficiency and effectiveness in our experiments. In our preliminary experiments, we try different ways to get a good visual query or cross-modal fused query for AAUM in the MLLM architecture, but the performance is still not as good as SAUM. We attribute

this to the fact that, the visual query used by AAUM to generate the aggregation weights is quite different between the Two-Tower VLM and MLLM architectures. They are taken from bidirectional / unidirectional transformer encoder / decoder, respectively. The casual nature of the representation from the LLM in the MLLM architecture may not be suitable for the usage of visual query in AAUM. It contradicts the bidirectional nature of multi-layer visual representations from the visual encoder. Use bidirectional and casual attention for the visual and textual part of the LLM, respectively, may be a potential solution to this problem [100]–[102].

- **Visual Representation Selection:** As shown in Fig. 6, in Two-Tower VLM architecture, the performance of ManagerTower first increases gradually with the number of unimodal representations, but then stops increasing and even decreases when the number of unimodal representations exceeds 6. Similar trend is also observed in the MLLM architecture in Fig. 9. How to obtain better performance with acceptable computational budget by utilizing more/better insights of unimodal experts, especially when scaling the model or the MLLM architecture with deeper and wider modules, *e.g.*, 24-layer CLIP-ViT L-224/16 and 24-layer LLM Qwen2-0.5B-Instruct, is a question worth further exploration. For example, designing reasonable sparse activation functions for managers, instead of manually selection, simple top-N or top-p sampling (which did not work well in our preliminary experiments).
- **Manager Injection:** In the Two-Tower VLM architecture, we inject managers into each cross-modal layer. In the MLLM architecture, we uniformly inject managers into the LLM from the first layer at a fixed layer interval. How to inject managers into the LLM in a more flexible and adaptive way is also a question worth further exploration. For example, non-uniform injection and do not start from the first layer.

In addition, it would also be interesting to explore the effectiveness of managers in other VLM architectures [103], [104], multimodal in-context learning [105], [106] and multimodal chain-of-thought reasoning [107], [108], and the collaboration with multi-grained multimodal data [109], [110].

There are also some recent works that follow or share the spirit of our manager to aggregate multi-layer unimodal representations in VLMs and MLLMs [6], [7], [85]–[87]. Besides, some works [96], [111], [112] explore the utilization of different visual encoders (with different resolutions), *e.g.*, DINOv2 [113] and SigLIP [39], to improve the visual representation. They aggregate multiple visual representations from different visual encoders (experts), which is similar to our manager that aggregates multi-layer visual representations from different layers of the visual encoder (we treat each layer of the visual encoder as an expert).

B. *Semantic Ambiguity in the Multi-Grid Algorithm*

In Section V-E, we provide a detailed analysis and case study of the semantic ambiguity caused by the multi-grid algorithm in the MLLM architecture. In fact, the multi-grid

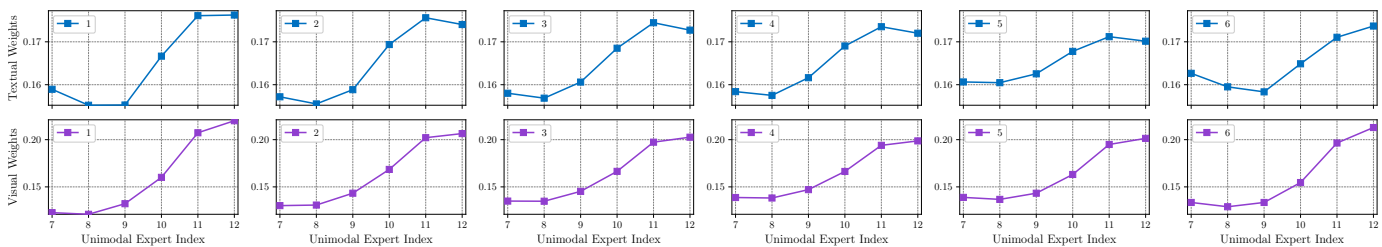


Fig. 21. A visualization of aggregation weights of textual and visual SAMs in each cross-modal layer. The X-axis is the index of the unimodal expert, and the legend shows the index of the cross-modal layer.

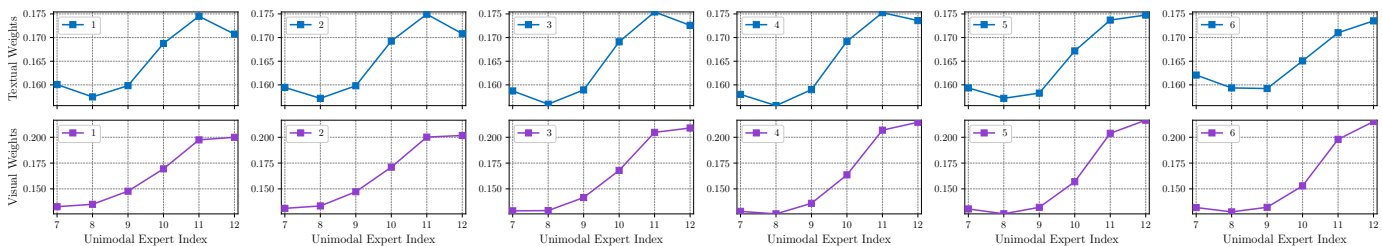


Fig. 22. A visualization of aggregation weights of textual and visual SAUMs in each cross-modal layer. The X-axis is the index of the unimodal expert, and the legend shows the index of the cross-modal layer.

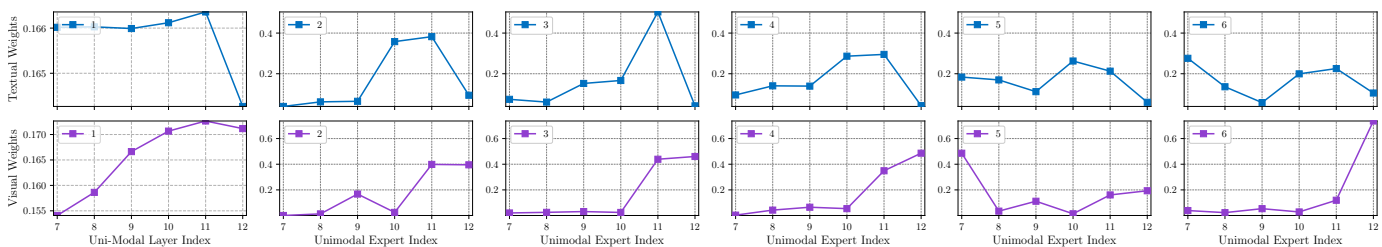


Fig. 23. A visualization of aggregation weights of textual and visual AAUMs in each cross-modal layer. The X-axis is the index of the unimodal expert, and the legend shows the index of the cross-modal layer.

algorithm is a resource-efficient way to directly split the image into multiple image grids, which can effectively improve the performance of MLLMs in high-resolution scenarios. It is widely used in both academic research [12], [36], [96], [97] and industrial applications [114]–[116]. It also called dynamic image tiling or image cropping in some works.

However, the multi-grid algorithm may bring semantic ambiguity since the grid line may cut off objects and connected regions in the image [59], [98], [117]. Current works try to mitigate the semantic ambiguity by introducing new modules to enhance the interaction between different image grids [12], [117], or by simultaneously adopting two different grid partitioning schemes [59]. They all bring additional training and computational costs. Furthermore, our manager provides an orthogonal perspective to alleviate the semantic ambiguity by incorporating different levels of unimodal semantic knowledge contained in the multi-layer unimodal representations from the visual encoder, which can capture and supplement more diverse visual details. The diversity of attention weights and attention heads in the multi-grid MLLM will be increased by our manager, which also demonstrates the guidance and collaboration between the manager and the multi-grid algorithm.

In fact, no matter the image patch in the ViTs or the feature map from a convolutional kernel in the CNNs, they all suffer

from the semantic ambiguity caused by partitioning the image into small regions. The self-attention mechanism in the ViTs and the convolutional operation in the CNNs can help to alleviate the semantic ambiguity by interacting with different regions of the image. Directly training a high-resolution visual encoder is a more straightforward and effective way to improve the supported image resolution of the visual encoder, but need precious and scarce high-resolution training data for continual training of the visual encoder [92]–[95]. From the perspective of improving the supported image resolution, the multi-grid algorithm can be seen as an ensemble of the visual encoder but with different image grids. The separate encodings of different image grids bring more diverse visual details, but also further intensify the semantic ambiguity. Appending the representation of the resized base image before the image grids alleviates the semantic ambiguity, but not enough based on our experiments. The collaboration between our manager (deeper representation) and the multi-grid algorithm (wider representation) with different interaction modules for different image grids is a promising direction to further improve the performance of multi-grid MLLMs.

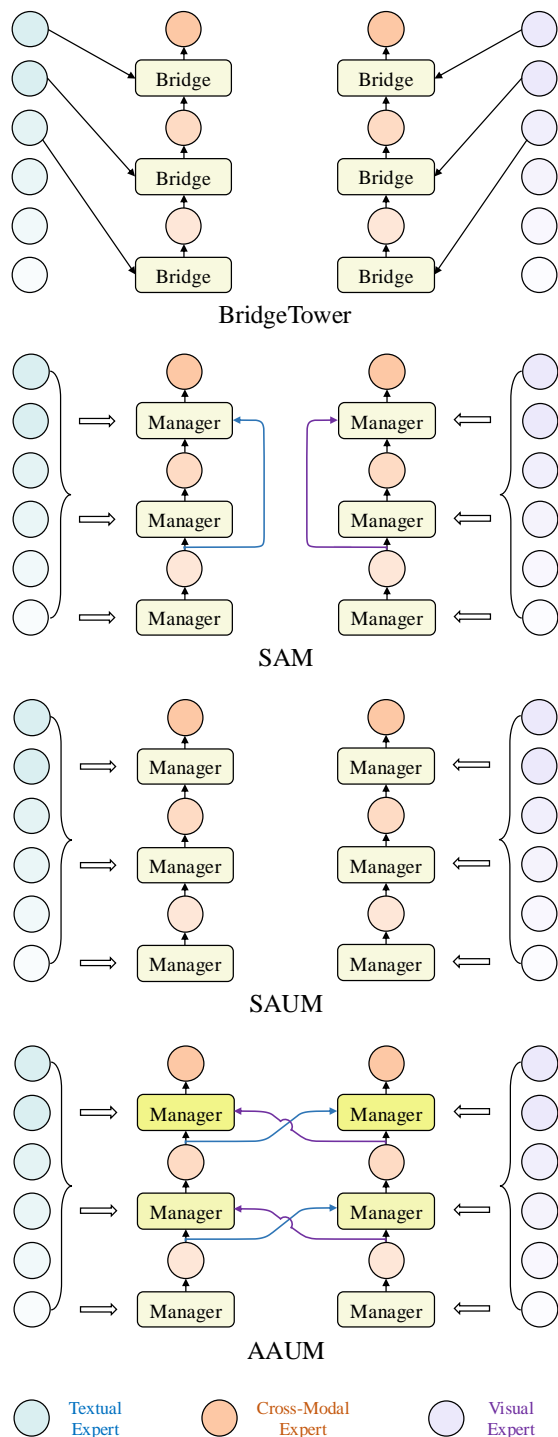


Fig. 24. Brief illustrations of BridgeTower and our ManagerTower with SAM, SAUM and AAUMs. Hollow arrows indicate the transmission of multi-layer unimodal representations in ManagerTower instead of layer-by-layer transmission in BridgeTower. Each unimodal or cross-modal layer is seen as an unimodal or 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.

APPENDIX B

EXPLORATION ON TWO-TOWER VLM

A. Intuitive Comparison Between BT&MT

We intuitively compare BridgeTower (BT) and ManagerTower (MT) with different type of managers in Fig. 24.

1) *BT vs. MT with SAUM*: In Table II & V, we provide the detailed performance comparison between BridgeTower and ManagerTower.¹³ In fact, BridgeTower can be seen as an approximate special case of ManagerTower with SAUMs if we replace the learned weights \mathbf{W} in each manager with layer-by-layer one-hot distributions¹⁴ used in BridgeTower. However, as shown in Fig. 22, the aggregation weight of textual and visual SAUMs share a similar progressive trend across cross-modal layers, which is completely different from the one-hot distributions in BridgeTower. This allows ManagerTower with SAUMs to achieve significant performance gains (75.91 vs. 76.55) compared to BridgeTower. Besides, the similar trend of aggregation weight distributions between different managers is consistent with the observations in Fig. 4, that is, the cosine similarity of aggregated unimodal representations between managers is always similar to 1.

2) *SAM vs. SAUM vs. AAUM*: We provide the visualizations of aggregation weights of SAM, SAUM and AAUM without VLP in Fig. 21 & 22 & 23. Comparing the visualization of three types of managers without VLP, we can find that: (i) the learned aggregation weights of SAM and SAUM share a similar progressive trend across cross-modal layers; (ii) for each AAUM, its generated aggregation weights vary significantly across 6 unimodal experts. Comparing different AAUMs, the aggregation weight distributions generated by AAUMs is also very different. (iii) when we compare Fig. 22 & 23, their respective aggregation weight distributions are completely different. This further demonstrates that compared with SAUMs, AAUMs can adaptively **generates** different aggregation weights for different tokens in different samples. Interestingly, the first column of two figures both comes from SAUMs, but the distributions are still **clearly** different. We presume that high-layer AAUMs may help low-layer SAUMs **rectify** their management of experts.

B. Switch Visual and Textual Backbones

We experiment with different pre-trained visual and textual backbones as unimodal encoders to further investigate the impact on performance of the managers of ManagerTower compared to the bridges of BridgeTower. As shown in Table V, 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 unimodal representations for VL representation learning.

C. Computational Budget

Table VI shows the computational budget and downstream task performance without VLP for BridgeTower and ManagerTower, including the number of parameters, the number of

¹³The 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 IV-B.

¹⁴It means that, for each cross-modal layer, only one unimodal expert is activated at a time in the bottom-up direction.

TABLE V
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.

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

TABLE VI
COMPUTATIONAL BUDGET AND DOWNSTREAM TASK PERFORMANCE WITHOUT VLP FOR BRIDGETOWER AND MANAGERTOWER. * DENOTES OUR RE-IMPLEMENTATION.

Model	Manager Type	Manager Visual Query	# Params (M)	# FLOPs (G)	Inference Time (ms/sample)	VQAv2 Test-Dev	Flickr30K R _{MEAN}
BridgeTower _{BASE} *	-	-	326.58	101.25	39.43	75.91	93.33
ManagerTower _{BASE}	SAUM	-	326.77	101.34	41.12 (×1.04)	76.55 (↑0.64)	93.73 (↑0.40)
ManagerTower _{BASE}	AAUM	$C_{\ell-1}^V$	326.77	101.35	41.80 (×1.06)	76.52 (↑0.61)	93.84 (↑0.51)
ManagerTower _{BASE}	AAUM	$C_{\ell-1}^V, C_{\ell-1}^T$	338.64	105.52	43.20 (×1.10)	76.65 (↑0.74)	93.97 (↑0.64)

Floating-Point operations (FLOPs),¹⁵ and the average inference time per instance.

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 384×384 . Compared with BridgeTower (1st row), ManagerTower (4th row) uses an acceptable additional computational budget (3.7% parameters, 4.2% FLOPs, and 3.8ms inference time) and achieves significant absolute performance improvements of 0.74% and 0.64% on VQAv2 and Flickr30K, respectively. We further analyse other well-performed variants of ManagerTower in the 2nd and 3rd 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 (4th row), as it is the only difference between the 3rd and 4th 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. Cross-Attention and Concat-Attention Managers

1) *Cross-Attention Managers*: We implement the standard cross-attention mechanism [17] and reduce the linear projection layer for value to save computational budget.¹⁶ 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.*, $V[:, 0] \in \mathbb{R}^{N \times D}$, as the key. Hence, the shape of generated aggregation weights is $N \times L$, which can

be broadcast to the aggregation weights $W_A \in \mathbb{R}^{N \times L \times D}$. The following calculation is the same as AAUMs in Fig. 5. The results in Table I show a significant decrease compared to other managers on Flickr30K. We attribute this to the fact that the cross-attention manager will break the locality (or spatial consistency) of unimodal representations [99], [112], which is crucial for the downstream tasks like image-text retrieval. We leave the detailed analysis of this phenomenon to the future work.

2) *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 $V \in \mathbb{R}^{N \times L \times D}$ along the last dimension as the concatenated query. It then directly projects the query to $W_A \in \mathbb{R}^{N \times L \times D}$. The following calculation is the same as AAUMs in Fig. 5. 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 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 III. Here we provide more details on the comparison with previous arts in Table VII.

F. Implementation Details

1) *Vision-Language Pre-training*: Following METER [6], we use two common VLP objectives for Two-Tower VLMs.

¹⁵We use Facebook Research’s *fvcore*, to calculate FLOPs.

¹⁶The calculation of cross-modal fused query also uses this simplified version of the cross-attention mechanism.

TABLE VII

COMPARISONS WITH PREVIOUS MODELS ON 4 DOWNSTREAM DATASETS. THE BEST SCORE IS BOLDDED. 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.

Model	# Pre-train Images	Visual Backbone	VQAv2		SNLI-VE		NLVR ²		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 _{BASE} [24]	4M	ViT B-384/32	71.26	-	-	-	75.70	76.13	64.4	83.5
UNITER _{BASE} [25] *	4M	Faster R-CNN	72.70	72.91	78.59	78.28	77.18	77.85	72.52	85.90
VILLA _{BASE} [118] *	4M	Faster R-CNN	73.59	73.67	79.47	79.03	78.39	79.30	74.74	86.60
UNIMO _{BASE} [26]	4M	Faster R-CNN	73.79	74.02	80.00	79.10	-	-	74.66	89.70
ALBEF _{BASE} [27] *	4M	DeiT B-224/16	74.54	74.70	80.14	80.30	80.24	80.50	82.80	94.30
VinVL _{BASE} [119]	5.7M	ResNeXt-152	75.95	76.12	-	-	82.05	83.08	-	-
METER-Swin _{BASE} [6]	4M	Swin B-384/32	76.43	76.42	80.61	80.45	82.23	82.47	79.02	92.40
VLMO _{BASE} [28]	4M	BEiT B-224/16	76.64	76.89	-	-	82.77	83.34	79.30	92.30
METER-CLIP _{BASE} [6]	4M	CLIP-ViT B-224/16	77.68	77.64	80.86	81.19	82.33	83.05	82.22	94.30
BridgeTower _{BASE} [7]	4M	CLIP-ViT B-224/16	78.66	78.73	81.11	81.19	81.85	83.09	85.83	94.73
ManagerTower _{BASE} (Ours)	4M	CLIP-ViT B-224/16	79.39	79.15	81.26	81.44	82.81	83.34	86.56	95.64
<i>Models pre-trained on more data and/or with larger size</i>										
UNITER _{LARGE} [25] *	4M	Faster R-CNN	73.82	74.02	79.39	79.38	79.12	79.98	75.56	87.30
VILLA _{LARGE} [118] *	4M	Faster R-CNN	74.69	74.87	80.18	80.02	79.76	81.47	76.26	87.90
UNIMO _{LARGE} [26]	4M	Faster R-CNN	75.06	75.27	81.11	80.63	-	-	78.04	89.40
ALBEF _{BASE} [27] *	14M	DeiT B-224/16	75.84	76.04	80.80	80.91	82.55	83.14	85.60	95.90
VinVL _{LARGE} [119]	5.7M	ResNeXt-152	76.52	76.63	-	-	82.67	83.98	-	-
BLIP _{BASE} [30] *	14M	DeiT B-224/16	77.54	77.62	-	-	82.67	82.30	87.20	96.60
SimVLM _{BASE} [29] *	1.8B	ResNet-101	77.87	78.14	84.20	84.15	81.72	81.77	-	-
BLIP _{BASE} [30] *	129M	DeiT B-224/16	78.24	78.17	-	-	82.48	83.08	87.30	97.30
SimVLM _{LARGE} [29] *	1.8B	ResNet-152	79.32	79.56	85.68	85.62	84.13	84.84	-	-
VLMO _{LARGE} [28]	4M	BEiT L-224/16	79.94	79.98	-	-	85.64	86.86	84.50	95.30
SimVLM _{HUGE} [29] *	1.8B	Larger ResNet-152	80.03	80.34	86.21	86.32	84.53	85.15	-	-

a) *Masked Language Modeling (MLM)*: For MLM, we follow the conditional masking approach used in UNITER [25] 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 top-layer representation of the textual part of the cross-modal encoder is used as input for the MLM task head.

b) *Image-Text Matching (ITM)*: 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 top-layer representations of [class] and [<s>] tokens are activated by the non-linear function Tanh. Then the concatenation of the above representations is fed into a linear classifier with cross-entropy loss for classification.

2) *Pre-training Settings*: Table VIII shows the statistics of the pre-train datasets. Following previous work [6], [24], [25], [27], we adopt four public image-caption datasets for pre-training, including Conceptual Captions (CC) [31], SBU Captions (SBU) [32], MSCOCO Captions (COCO) [33], and Visual Genome (VG) [34]. The total numbers of the unique images and image-caption pairs in the combined training data are 4M and 9M. Table IX describes the hyperparameters for pre-training the ManagerTower. The learning rate of the cross-modal encoder is five times higher than the base learning

TABLE VIII

STATISTICS OF THE PRE-TRAIN DATASETS. WE REMOVE DUPLICATE IMAGE-CAPTION PAIRS IN VG [6], [24] AND ONLY 2.9M IMAGE-CAPTION PAIRS CAN BE DOWNLOADED IN CC.

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

rate [6], e.g., used by unimodal encoders.

3) Fine-Tuning Settings:

a) *Dataset Setting*: Standard settings and splits are used for all datasets. Noted that, for Flickr30K [5], we follow the standard Karpathy Split [120]; for VQAv2 [2], we follow the common practice [2], [121]: 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.

b) *Image Augmentation*: We follow previous works [27], [30] to use the combination of RandomResizedCrop, RandomHorizontalFlip, and RandAugment [122] during training.

4) *Fine-Tuning Strategies*: For visual question answering (VQAv2 [2]), visual entailment (SNLI-VE [3]) and visual reasoning (NLVR² [4]), the fine-tuning strategy is similar to the strategy we used in the ITM 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 classifier.

For image-text retrieval (Flickr30K [5]), we follow the

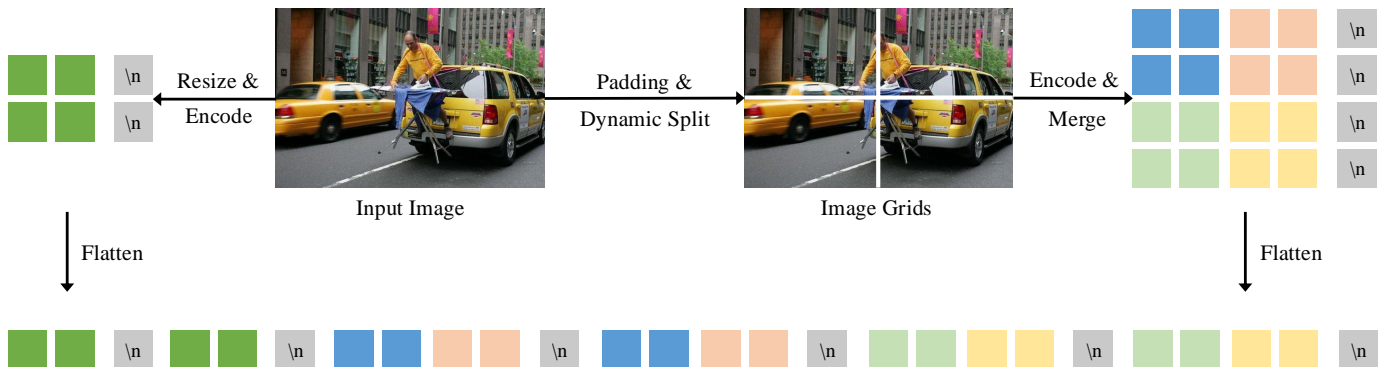


Fig. 25. A brief illustrations of the multi-grid algorithm used in LLaVA-OV. The bilinear interpolation operation is omitted for simplicity, which is used to reduce # input visual tokens. “\n” denotes the special token to indicate the end of a row and the shape of the input image. Illustration inspired by DeepSeek-VL2 [116].

approach used in ALBEF [27] 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 unimodal encoders and calculate the contrastive similarity of unimodal representations of image–text pairs by dot product to compute the 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.

Table XIV describes the hyperparameters for fine-tuning on 4 downstream datasets. Following previous work [6], [24], we apply cross-entropy loss for SNLI-VE, NLVR² and Flickr30K and binary cross-entropy loss for VQA_{v2}.

APPENDIX C EXPLORATION ON MLLM

A. A Brief Illustration of the Multi-Grid Algorithm

The multi-grid algorithm [12], [36], [96], [97] is a widely used technique in MLLMs to help the visual encoder, *e.g.*, SigLIP, efficiently process images with varying image resolutions and aspect ratios. We show a brief illustration of the multi-grid algorithm used in LLaVA-OV in Fig. 25, and recommend readers to refer to the original paper [11] for more details.

B. Average Attention Distance of SigLIP

As stated in Section V-D1, similar to the observations in both BridgeTower and ManagerTower, visual representations from the top half of the visual encoder bring the best performance, and using visual representations from all layers leads to the lowest performance improvement.

We visualize the average attention distance of SigLIP in Fig. 26. It is computed by averaging the 2D euclidean distance between the query patch (token) and all other patches

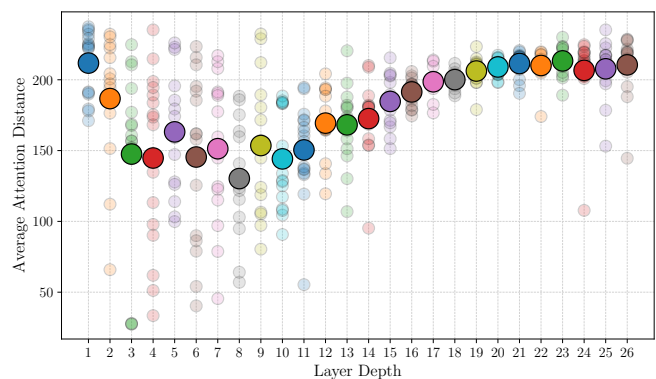


Fig. 26. Size of attended area by head and layer depth. Each small/large dot shows the mean attention distance of each head or all heads in each layer of the visual encoder. Image width is 384 pixels and patch size is 14×14 pixels.

(tokens), weighted by the attention weight of visual self-attention. Similar to the observations in ViT [58], in LLaVA-OV, the average attention distance vary significantly across heads in the bottom half of SigLIP, with some heads focusing on the most of the image while others focusing on the query location or a small region nearby. As depth increases, especially in the top half of the visual encoder, the average attention distance of all heads increases, where most heads attend **widely** across tokens and capture global visual features. Above observations provide a possible explanation for the performance difference between using visual representations from the top half and all layers of SigLIP in LLaVA-OV.

C. Detailed Analysis on ScienceQA and OK-VQA

As a supplement for Section V-E, we further provide a detailed analysis ScienceQA and OK-VQA, to intuitively analyse the effectiveness of managers.

1) *ScienceQA*: This is a knowledge-intensive dataset with low-resolution hybrid images. As shown in Fig. 27, both the manager and the multi-grid algorithm can bring performance improvements on different dimensions. For “Language Science”, the manager brings significant improvements to Baseline, and such improvement seems to overlap with that provided by the multi-grid algorithm. For “Natural Science,

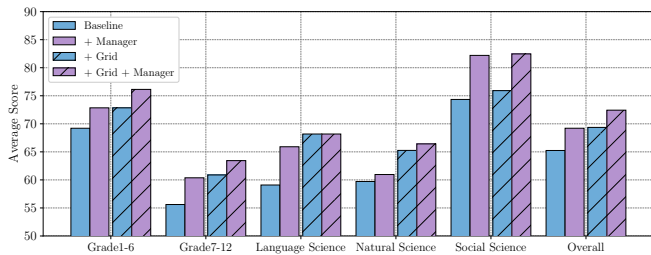


Fig. 27. Zero-shot performance of four baselines on ScienceQA-IMG test set. The overall average score and the average score of each dimension are shown.

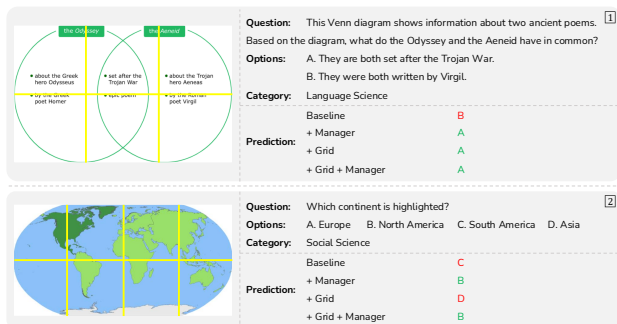


Fig. 28. Case studies of four baselines on ScienceQA-IMG test set. Red and green fonts represent incorrect and correct predictions, respectively. Yellow lines indicate the boundaries of the image grids.

Social Science”, the manager brings stable improvements. Especially in “Social Science”, we can notice that the multi-grid algorithm brings **slight** improvements to Baseline, while our manager can bring **significant** improvements to both Baseline and Baseline+Grid. Furthermore, their synergy can further improve performance. Take the case [1](#) in Fig. 28 as an example, in “Language Science”, most samples require analysis based on text information in the image or the question. Baseline misses visual details in the image, and both the manager and the multi-grid algorithm can help Baseline capture visual details “set after the Trojan War”. As for the case [2](#), the improvement of Manager mainly comes from map related problems. The multi-grid algorithm divides the map into multiple grids, which may make it more difficult to understand, e.g., the highlighted North America is divided into two grids, and then bring semantic ambiguity. By introducing different levels of semantic knowledge, our manager can not only help Baseline to capture the highlighted part, but also mitigate semantic ambiguity caused by the multi-grid algorithm.

2) *OK-VQA*: This is a general dataset with low-resolution natural images. As shown in Fig. 29, surprisingly, the multi-grid algorithm does not improve the performance much and even leads to **significant** degradation on many dimensions:

- Objects, Material, and Clothing; People and Everyday Life; Plants and Animals; Sports and Recreation: images about common objects, things, and scenes in daily life or in nature.
- Cooking and Food: close-ups of food or dining table.
- Geography, History, Language and Culture: complex buildings (groups), event scenes containing different types of objects, etc.

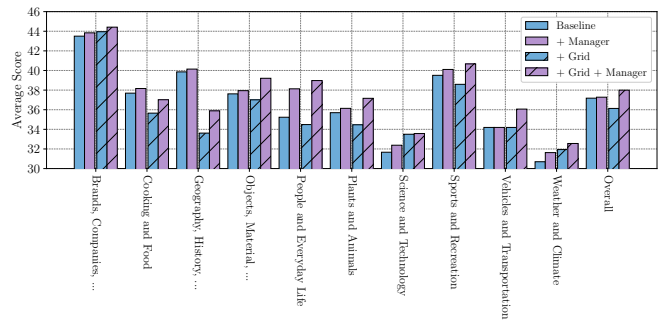


Fig. 29. Zero-shot performance of four baselines on OK-VQA validation set.

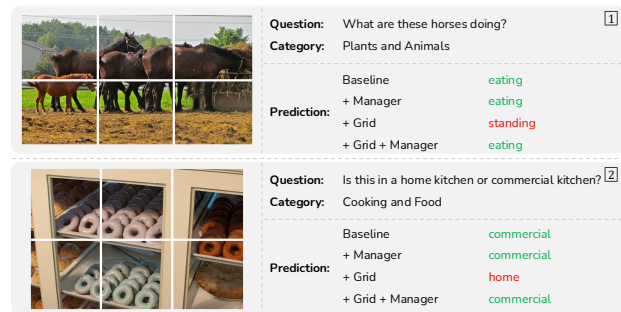


Fig. 30. Case studies of four baselines on OK-VQA validation set. Red and green fonts represent incorrect and correct predictions, respectively. White lines indicate the boundaries of the image grids.

Take cases in Fig. 30 as an example, the multi-grid algorithm may actually increase the understanding difficulty and bring semantic ambiguity. Especially for (complex) scenes with many (different types of) objects and things, it cuts off objects and connected regions, which may hinder the perception of objects, things and scenes. By aggregating insights from different levels of visual experts, our manager can not only improve Baseline, but also partially compensate for the performance loss caused by the multi-grid algorithm, and may even further improve the performance on some dimensions.

D. Attention Weight Distribution Analysis without Multi-Grid

As a supplement for Section V-F2 we further provide the average entropy and KL divergence of attention weight distributions in each layer of Baseline and Baseline+Manager. Similar trend can be observed in Fig. 31 and 32, benefiting from difference levels of semantic knowledge introduced by our manager, Baseline+Manager shows higher diversity of attention weight and attention heads in each layer. This helps the model capture richer and more diverse features, and further improve the performance on downstream tasks.

E. Implementation Details

In this paper, we take LLaVA-OneVision-0.5B-SI [11] as our baseline (LLaVA-OV for short), which is a widely used open-source multi-grid MLLM. We follow the same training settings as the original LLaVA-OV and use about 8M data samples for multi-stage training. Different Two-Tower VLMs

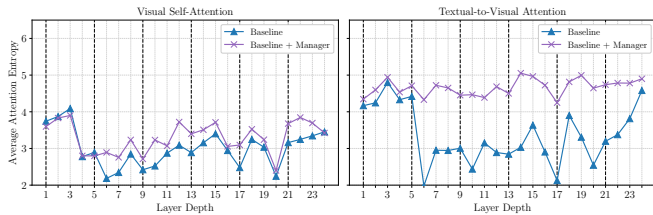


Fig. 31. Average entropy of attention weight distributions in each layer of Baseline and Baseline+Manager. The dotted vertical lines indicate the layers where managers are injected.

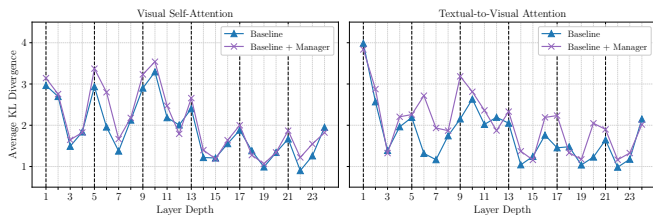


Fig. 32. Average KL divergence between attention weight distributions of attention heads in each layer of Baseline and Baseline+Manager. The dotted vertical lines indicate the layers where managers are injected.

that trained with various pre-training objectives and fine-tuning strategies for different downstream tasks, most of latest MLLMs are trained with the autoregressive objective, which teaches the model to generate the answer tokens one by one. LLaVA-OV adopts a curriculum learning strategy that trains the model with 3 different stages:

- **Stage-1:** Language-Image Alignment. A small amount of data is used to initially align the visual representation from the visual encoder to the word embedding space of the LLM. 558K image-caption pairs from LCS-558K [57] are used in this stage, and the captions come from the Internet.
- **Stage-1.5:** High-Quality Knowledge Learning. About 4M data are used in this stage to learn help the model further align the visual encoder and the LLM, and also learn high-quality knowledge. The data consists of re-captioned detailed description data and document/ocr data, and most of the data is synthetic.
- **Stage-2:** Visual Instruction Tuning. About 3.2M data are used in this stage to teach the model to solve diverse downstream tasks under the zero-shot setting. The data consists of a wide range of multimodal datasets.

We recommend readers to refer to the original paper [11] for more details about the training settings and data. The only two differences between our Baseline+Manager and the original LLaVA-OV are: (i) **Data:** partial training data are not open sourced; (ii) **Maximum length:** we use 16,384 instead of 32,768 as the maximum length of each input sample for better efficiency (more than 99% of the samples have a length less than 16,384). The above two differences bring slight performance differences between Baseline+Manager and the original LLaVA-OV, and we remove few some downstream datasets for better efficiency and robustness.

TABLE IX
HYPERPARAMETERS FOR PRE-TRAINING. THE FIRST BLOCK IS THE HYPERPARAMETERS FOR THE CROSS-MODAL ENCODER.

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 ϵ	$1e^{-8}$
Adam β_1	0.9
Adam β_2	0.98
Center-Crop	✓
Random Resized Crop	✗
Random Augmentation	✗
Random Horizontal Flipping	✗
Textual Encoder	RoBERTa _{BASE}
Visual Encoder	CLIP-ViT B-224/16
Patch Size	16
Image Resolution for VLP	288

F. Evaluation Details

As stated in Section V-B3, we follow the same evaluation settings as the original LLaVA-OV [11] for the zero-shot evaluation on 20 downstream datasets. We further divide these datasets into different groups from three perspectives, *i.e.*, category of capabilities [112], [123], images, and resolutions, to better analyze the effectiveness of managers in MLLMs and multi-grid MLLMs. Details about the evaluation datasets we used are shown in Table X.

G. Detailed Results

We provide results of each category for our four baselines in Table XI, across different categories of capabilities, images, and resolutions. Furthermore, we also provide detailed results of our four baselines on each dataset in Table XII, and XIII.

TABLE X

DETAILS OF 20 DOWNSTREAM DATASETS USED IN OUR EXPLORATION ON MLLMS. FOR IMAGE RESOLUTION AND ASPECT RATIO, WE PROVIDE THE MINIMUM, MAXIMUM, AND AVERAGE VALUES IN THE FORM OF $x/y/z$. THE THRESHOLD FOR LOW AND HIGH RESOLUTION IS $384 \times 2 = 768$ PIXELS. FOR MM-LIVEBENCH, WE USE TWO DIFFERENT SUBSETS OF IT (2407 AND 2409). * INDICATES THE DATASETS ARE ONE OF THE 9 DATASETS WE USED IN THE ABLATION STUDY FOR EFFICIENCY AND ROBUSTNESS.

Capability Category	Dataset	Description	Answer Category	Image Category	Resolution Category	Resolution	Aspect Ratio	# Samples
General	VQAv2 [2] *	Scene understanding VQA	Open form (Short)	Natural	Low	120 / 640 / 523	0.17 / 1.00 / 0.72	214.00K
	OKVQA [43]	External knowledge VQA	Open form (Short)	Natural	Low	208 / 640 / 524	0.26 / 1.00 / 0.72	5.05K
	GQA [44] *	Compositional VQA	Open form (Short)	Natural	Low	314 / 640 / 523	0.36 / 1.00 / 0.71	12.58K
	MMVet [45]	Multi-discipline	Open form (Long)	Hybrid	High	181 / 4180 / 909	0.27 / 1.00 / 0.74	0.22K
	SEED-Bench [46] *	Multi-discipline; Large-scale	Multi-choice	Natural	High	240 / 9906 / 791	0.50 / 1.00 / 0.66	18.00K
	RealWorldQA [47] *	Real-world VQA	Multi-choice; Open form (Short)	Natural	High	512 / 1469 / 1147	0.45 / 1.00 / 0.67	0.77K
Text-Rich	TextVQA [48] *	Scene Text Recognition	Open form (Short)	Natural	High	512 / 3991 / 872	0.25 / 1.00 / 0.73	5.00K
	ChartQA [49]	Chart Understanding	Open form (Short)	Abstract	Low	245 / 1199 / 665	0.39 / 0.99 / 0.71	2.50K
	DocVQA [37] *	Document Understanding	Extractive	Abstract	High	418 / 6209 / 1918	0.33 / 0.99 / 0.75	5.19K
	InfoVQA [50]	Infographic Understanding	Extractive; Numerical	Abstract	High	452 / 7655 / 1724	0.09 / 1.00 / 0.46	3.29K
	OCRBench [38]	Text Recognition	Open form (Short)	Hybrid	Low	20 / 5900 / 627	0.08 / 1.00 / 0.57	1.00K
Knowledge	AI2D [51] *	Science Diagrams	Multi-choice	Abstract	Low	152 / 1500 / 547	0.18 / 1.00 / 0.72	3.09K
	ScienceQA [52] *	High-school Science	Multi-choice	Hybrid	Low	114 / 685 / 378	0.10 / 1.00 / 0.66	2.02K
	MMMU [53] *	College-level Multi-discipline	Multi-choice; Open form (Short)	Hybrid	Low	185 / 2217 / 580	0.11 / 1.00 / 0.60	0.90K
	MathVista [54]	General Math Understanding	Multi-choice; Open form (Short)	Hybrid	Low	58 / 4275 / 499	0.10 / 1.00 / 0.72	1.00K
Real-World	ImageDC [55]	Image Detail Description	Open form (Long)	Hybrid	Low	470 / 512 / 499	0.60 / 1.00 / 0.76	0.10K
	MM-LiveBench [56]	Internet Content Understanding	Open form (Long)	Hybrid	High	980 / 2051 / 1788	0.51 / 0.92 / 0.61	0.25K
	LLaVA-Wild [57]	Real-world Chat	Open form (Long)	Hybrid	High	465 / 3921 / 1277	0.51 / 1.00 / 0.80	0.06K
	LLaVA-Wilder [35]	Real-world Chat	Open form (Long)	Hybrid	High	224 / 3492 / 870	0.37 / 1.00 / 0.72	0.21K

TABLE XI

ZERO-SHOT PERFORMANCE OF FOUR BASELINES ON 20 DATASETS. THE OVERALL AVERAGE SCORE AND THE AVERAGE SCORE OF EACH CAPABILITY CATEGORY ARE SHOWN.

Model	Overall	Capability Category				Image Category			Resolution Category	
		General	Text-Rich	Knowledge	Real-World	Natural	Abstract	Hybrid	Low	High
Baseline	50.54	48.72	47.52	46.05	57.26	56.49	46.50	48.60	54.34	46.74
+ Manager	51.50	49.59	48.53	47.63	57.74	57.44	47.48	49.53	55.31	47.68
+ Grid	53.87	49.07	57.26	47.50	58.96	57.92	55.06	50.98	56.25	51.50
+ Grid + Manager	55.21	50.35	59.22	48.65	59.69	59.22	56.63	52.24	57.48	52.95

TABLE XII

ZERO-SHOT PERFORMANCE OF FOUR BASELINES ON 11 DATASETS OF “GENERAL” AND “TEXT-RICH” CAPABILITY CATEGORIES. THE SCORE OF EACH DATASET AND THE AVERAGE SCORE OF EACH CAPABILITY CATEGORY ARE SHOWN. THE SCORE OF OCRBENCH WILL BE NORMALIZED BY DIVIDING BY 10 FOR THE AVERAGE CALCULATION.

Model	General							Text-Rich					
	VQAv2	OKVQA	GQA	MMVet	SEED-Bench	RealWorldQA	Average	TextVQA	ChartQA	DocVQA	InfoVQA	OCRBench	Average
	val	val	testdev	test	image	test		val	test	test	test	test	
Baseline	73.90	37.18	57.66	21.40	61.50	50.98	48.71	57.56	54.32	51.40	27.30	470.00	47.52
+ Manager	74.18	37.28	57.89	22.40	65.33	51.11	49.61	59.00	56.24	52.49	27.31	476.00	48.53
+ Grid	74.65	36.13	57.95	24.00	61.08	52.55	49.07	65.14	59.32	68.55	38.61	547.00	57.26
+ Grid + Manager	74.90	37.99	58.66	25.60	65.14	52.81	50.35	65.82	60.96	70.90	40.70	577.00	59.22

TABLE XIII

ZERO-SHOT PERFORMANCE OF FOUR BASELINES ON 9 DATASETS OF “KNOWLEDGE” AND “REAL-WORLD” CAPABILITY CATEGORIES. THE SCORE OF EACH DATASET AND THE AVERAGE SCORE OF EACH CAPABILITY CATEGORY ARE SHOWN.

Model	Knowledge					Real-World					
	AI2D	ScienceQA	MMMU	MathVista	Average	ImageDC	MM-LiveBench	LLaVA-Wild	LLaVA-Wilder	Average	
	test	test	val	testmini		test	July	Sep	test	test	
Baseline	52.95	65.25	31.78	34.20	46.05	89.18	38.76	41.50	68.40	50.00	57.57
+ Manager	53.89	69.21	33.11	34.30	47.63	89.44	39.74	43.27	69.00	50.70	58.43
+ Grid	53.76	69.36	30.78	36.10	47.50	89.73	41.28	43.80	68.80	51.20	58.96
+ Grid + Manager	53.95	72.43	31.22	37.00	48.65	89.95	41.89	45.13	69.30	52.20	59.69

TABLE XIV
HYPERPARAMETERS FOR FINE-TUNING MANAGER TOWER ON 4 DOWNSTREAM DATASETS. FT DENOTES FINE-TUNING. CE AND BCE ARE SHORT FOR CROSS-ENTROPY LOSS AND BINARY CROSS-ENTROPY LOSS, RESPECTIVELY.

Hyperparameters	VQAv2	SNLI-VE	NLVR ²	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 ϵ	$1e^{-8}$	$1e^{-8}$	$1e^{-8}$	$1e^{-8}$
Adam β_1	0.9	0.9	0.9	0.9
Adam β_2	0.98	0.98	0.98	0.98
Center-Crop	✗	✗	✗	✗
Random Resized Crop	✓	✓	✓	✓
Random Augmentation	✓	✓	✓	✓
Random Horizontal Flipping	✗	✓	✓	✓
Textual Encoder	RoBERTa _{BASE}	RoBERTa _{BASE}	RoBERTa _{BASE}	RoBERTa _{BASE}
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

REFERENCES

- [1] X. Xu, B. Li, C. Wu, S.-Y. Tseng, A. Bhiwandiwala, S. Rosenman, V. Lal, W. Che, and N. Duan, “ManagerTower: Aggregating the insights of uni-modal experts for vision-language representation learning,” in *Proc. of ACL*, 2023, pp. 14 507–14 525.
- [2] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the V in VQA matter: Elevating the role of image understanding in visual question answering,” in *Proc. of CVPR*, 2017, pp. 6325–6334.
- [3] N. Xie, F. Lai, D. Doran, and A. Kadav, “Visual entailment: A novel task for fine-grained image understanding,” *ArXiv preprint*, vol. abs/1901.06706, 2019.
- [4] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, “A corpus for reasoning about natural language grounded in photographs,” in *Proc. of ACL*, 2019, pp. 6418–6428.
- [5] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.
- [6] Z. Dou, Y. Xu, Z. Gan, J. Wang, S. Wang, L. Wang, C. Zhu, P. Zhang, L. Yuan, N. Peng, Z. Liu, and M. Zeng, “An empirical study of training end-to-end vision-and-language transformers,” in *Proc. of CVPR*, 2022, pp. 18 145–18 155.
- [7] X. Xu, C. Wu, S. Rosenman, V. Lal, W. Che, and N. Duan, “Bridgetower: Building bridges between encoders in vision-language representation learning,” in *Proc. of AAAI*, 2023, pp. 10 637–10 647.
- [8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *Proc. of ICML*, vol. 139, 2021, pp. 8748–8763.
- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *ArXiv preprint*, vol. abs/1907.11692, 2019.
- [10] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, “Learning deep transformer models for machine translation,” in *Proc. of ACL*, 2019, pp. 1810–1822.
- [11] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, Y. Li, Z. Liu, and C. Li, “Llava-onevision: Easy visual task transfer,” *ArXiv preprint*, vol. abs/2408.03326, 2024.
- [12] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, “Llava-next: Improved reasoning, ocr, and world knowledge,” 2024.
- [13] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K. Chang, Z. Yao, and K. Keutzer, “How much can CLIP benefit vision-and-language tasks?” in *Proc. of ICLR*, 2022.
- [14] W. Li, C. Gao, G. Niu, X. Xiao, H. Liu, J. Liu, H. Wu, and H. Wang, “UNIMO-2: End-to-end unified vision-language grounded learning,” in *Proc. of ACL Findings*, 2022, pp. 3187–3201.
- [15] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proc. of ACL*, 2016, pp. 1715–1725.
- [16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, 2019.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. of NeurIPS*, 2017, pp. 5998–6008.
- [18] J. Lu, D. Batra, D. Parikh, and S. Lee, “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks,” in *Proc. of NeurIPS*, 2019, pp. 13–23.
- [19] Q. Wang, F. Li, T. Xiao, Y. Li, Y. Li, and J. Zhu, “Multi-layer representation fusion for neural machine translation,” in *Proc. of COLING*, 2018, pp. 3015–3026.
- [20] X. Wei, H. Yu, Y. Hu, Y. Zhang, R. Weng, and W. Luo, “Multiscale collaborative deep models for neural machine translation,” in *Proc. of ACL*, 2020, pp. 414–426.
- [21] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *ArXiv preprint*, vol. abs/1607.06450, 2016.
- [22] W. Fedus, B. Zoph, and N. Shazeer, “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity,” *Journal of Machine Learning Research*, vol. 23, pp. 1–39, 2022.
- [23] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. of ICLR*, 2019.
- [24] W. Kim, B. Son, and I. Kim, “Vilt: Vision-and-language transformer without convolution or region supervision,” in *Proc. of ICML*, vol. 139, 2021, pp. 5583–5594.
- [25] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, “Uniter: Universal image-text representation learning,” in *Proc. of ECCV*, 2020.
- [26] W. Li, C. Gao, G. Niu, X. Xiao, H. Liu, J. Liu, H. Wu, and H. Wang, “UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning,” in *Proc. of ACL*, 2021, pp. 2592–2607.
- [27] J. Li, R. R. Selvaraju, A. Gotmare, S. R. Joty, C. Xiong, and S. C. Hoi, “Align before fuse: Vision and language representation learning with momentum distillation,” in *Proc. of NeurIPS*, 2021, pp. 9694–9705.
- [28] H. Bao, W. Wang, L. Dong, Q. Liu, O. K. Mohammed, K. Aggarwal, S. Som, S. Piao, and F. Wei, “Vlmo: Unified vision-language pre-training with mixture-of-modality-experts,” in *Proc. of NeurIPS*, 2022.
- [29] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, “SimVlm: Simple visual language model pretraining with weak supervision,” in *Proc. of ICLR*, 2022.
- [30] J. Li, D. Li, C. Xiong, and S. C. H. Hoi, “BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation,” in *Proc. of ICML*, vol. 162, 2022, pp. 12 888–12 900.
- [31] P. Sharma, N. Ding, S. Goodman, and R. Soicrut, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *Proc. of ACL*, 2018, pp. 2556–2565.
- [32] V. Ordonez, G. Kulkarni, and T. L. Berg, “Im2text: Describing images using 1 million captioned photographs,” in *Proc. of NeurIPS*, 2011, pp. 1143–1151.
- [33] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” *ArXiv preprint*, vol. abs/1504.00325, 2015.
- [34] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *International journal of computer vision*, 2017.
- [35] H. Liu, C. Li, Y. Li, and Y. J. Lee, “Improved baselines with visual instruction tuning,” in *Proc. of CVPR*, 2024, pp. 26 296–26 306.
- [36] Z. Li, B. Yang, Q. Liu, Z. Ma, S. Zhang, J. Yang, Y. Sun, Y. Liu, and X. Bai, “Monkey: Image resolution and text label are important things for large multi-modal models,” in *Proc. of CVPR*, 2024, pp. 26 763–26 773.
- [37] M. Mathew, D. Karatzas, and C. Jawahar, “Docvqa: A dataset for vqa on document images,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 2200–2209.
- [38] Y. Liu, Z. Li, M. Huang, B. Yang, W. Yu, C. Li, X. Yin, C. Lin Liu, L. Jin, and X. Bai, “Ocrbench: On the hidden mystery of ocr in large multimodal models,” *ArXiv preprint*, vol. abs/2305.07895, 2023.
- [39] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, “Sigmoid loss for language image pre-training,” in *Proc. of ICCV*, 2023, pp. 11 941–11 952.
- [40] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang *et al.*, “Qwen2 technical report,” *ArXiv preprint*, vol. abs/2407.10671, 2024.
- [41] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring plain vision transformer backbones for object detection,” *ArXiv preprint*, vol. abs/2203.16527, 2022.
- [42] R. Zhang, J. Han, C. Liu, P. Gao, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, and Y. Qiao, “Llama-adapter: Efficient fine-tuning of language models with zero-init attention,” *ArXiv preprint*, vol. abs/2303.16199, 2023.
- [43] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, “OK-VQA: A visual question answering benchmark requiring external knowledge,” in *Proc. of CVPR*, 2019, pp. 3195–3204.
- [44] D. A. Hudson and C. D. Manning, “GQA: A new dataset for real-world visual reasoning and compositional question answering,” in *Proc. of CVPR*, 2019, pp. 6700–6709.
- [45] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang, “MM-vet: Evaluating large multimodal models for integrated capabilities,” in *Proc. of ICML*, 2024.
- [46] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan, “Seed-bench: Benchmarking multimodal llms with generative comprehension,” *ArXiv preprint*, vol. abs/2307.16125, 2023.
- [47] x.ai, “Grok-1.5 vision preview,” 2019.
- [48] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, “Towards VQA models that can read,” in *Proc. of CVPR*, 2019, pp. 8317–8326.
- [49] A. Masry, X. L. Do, J. Q. Tan, S. Joty, and E. Hoque, “ChartQA: A benchmark for question answering about charts with visual and logical reasoning,” in *Proc. of ACL Findings*, 2022, pp. 2263–2279.
- [50] M. Mathew, V. Bagal, R. Tito, D. Karatzas, E. Valveny, and C. Jawahar, “Infographicvqa,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1697–1706.

- [51] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi, "A diagram is worth a dozen images," in *Proc. of ECCV*, 2016, pp. 235–251.
- [52] P. Lu, S. Mishra, T. Xia, L. Qiu, K. Chang, S. Zhu, O. Tafjord, P. Clark, and A. Kalyan, "Learn to explain: Multimodal reasoning via thought chains for science question answering," in *Proc. of NeurIPS*, 2022.
- [53] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, C. Wei, B. Yu, R. Yuan, R. Sun, M. Yin, B. Zheng, Z. Yang, Y. Liu, W. Huang, H. Sun, Y. Su, and W. Chen, "Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi," in *Proceedings of CVPR*, 2024.
- [54] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, "Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts," in *Proc. of ICLR*, 2024.
- [55] B. Li, H. Zhang, K. Zhang, D. Guo, Y. Zhang, R. Zhang, F. Li, Z. Liu, and C. Li, "Llava-next: What else influences visual instruction tuning beyond data?" 2024.
- [56] K. Zhang, B. Li, P. Zhang, F. Pu, J. A. Cahyono, K. Hu, S. Liu, Y. Zhang, J. Yang, C. Li *et al.*, "Lmms-eval: Reality check on the evaluation of large multimodal models," *ArXiv preprint*, vol. abs/2407.12772, 2024.
- [57] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proc. of NeurIPS*, 2023.
- [58] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. of ICLR*, 2021.
- [59] M. Huang, Y. Liu, D. Liang, L. Jin, and X. Bai, "Mini-monkey: Alleviating the semantic sawtooth effect for lightweight mllms via complementary image pyramid," *ArXiv preprint*, vol. abs/2408.02034, 2024.
- [60] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. of ICLR*, 2015.
- [61] Z. Xie, Z. Geng, J. Hu, Z. Zhang, H. Hu, and Y. Cao, "Revealing the dark secrets of masked image modeling," in *Proc. of CVPR*, 2023, pp. 14 475–14 485.
- [62] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, pp. 79–86, 1951.
- [63] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: pre-training of generic visual-linguistic representations," in *Proc. of ICLR*, 2020.
- [64] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang, "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training," in *Proc. of AAAI*, 2020, pp. 11 336–11 344.
- [65] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *Proc. of ECCV*, 2020.
- [66] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," *ArXiv preprint*, vol. abs/2202.03052, 2022.
- [67] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som *et al.*, "Image as a foreign language: Beit pretraining for all vision and vision-language tasks," *ArXiv preprint*, vol. abs/2208.10442, 2022.
- [68] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, "Coca: Contrastive captioners are image-text foundation models," *ArXiv preprint*, vol. abs/2205.01917, 2022.
- [69] J. Luo, Y. Li, Y. Pan, T. Yao, J. Feng, H. Chao, and T. Mei, "Exploring vision-language foundation model for novel object captioning," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2024.
- [70] W. Zhou and Z. Zhou, "Unsupervised domain adaption harnessing vision-language pre-training," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, pp. 8201–8214, 2024.
- [71] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" in *Proc. of NeurIPS*, 2021, pp. 12 116–12 128.
- [72] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan, and M. Yang, "Intriguing properties of vision transformers," in *Proc. of NeurIPS*, 2021, pp. 23 296–23 308.
- [73] M. E. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih, "Dissecting contextual word embeddings: Architecture and representation," in *Proc. of EMNLP*, 2018, pp. 1499–1509.
- [74] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith, "Linguistic knowledge and transferability of contextual representations," in *Proc. of NAACL*, 2019, pp. 1073–1094.
- [75] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *Proc. of ACL*, 2019, pp. 3651–3657.
- [76] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL*, 2019, pp. 4171–4186.
- [77] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. of CVPR*, 2017, pp. 936–944.
- [78] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of CVPR*, 2017, pp. 2261–2269.
- [79] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. of CVPR*, 2018, pp. 2403–2412.
- [80] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Proc. of NeurIPS*, 2021, pp. 12 077–12 090.
- [81] D. Huang, X. Zhu, X. Li, and H. Zeng, "Clsr: Cross-layer interaction pyramid super-resolution network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, pp. 6273–6287, 2023.
- [82] Y. Zhang and X. Zhu, "Attention-based layer fusion and token masking for weakly supervised semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, pp. 7912–7921, 2024.
- [83] Y. Chen, S. Zhang, Y. Sun, J. Yang, W. Liang, and H. Wang, "Artificial-spiking hierarchical networks for vision-language representation learning," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2024.
- [84] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018, pp. 2227–2237.
- [85] Z. Dou, A. Kamath, Z. Gan, P. Zhang, J. Wang, L. Li, Z. Liu, C. Liu, Y. LeCun, N. Peng, J. Gao, and L. Wang, "Coarse-to-fine vision-language pre-training with fusion in the backbone," in *Proc. of NeurIPS*, 2022.
- [86] H. Yao, W. Wu, T. Yang, Y. Song, M. Zhang, H. Feng, Y. Sun, Z. Li, W. Ouyang, and J. Wang, "Dense connector for mllms," *ArXiv preprint*, vol. abs/2405.13800, 2024.
- [87] W. Li, Y. Yuan, J. Liu, D. Tang, S. Wang, J. Qin, J. Zhu, and L. Zhang, "Tokenpacker: Efficient visual projector for multimodal llm," *ArXiv preprint*, vol. abs/2407.02392, 2024.
- [88] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proc. of NeurIPS*, 2020.
- [89] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *ArXiv preprint*, vol. abs/2307.09288, 2023.
- [90] L. Qin, Q. Chen, X. Feng, Y. Wu, Y. Zhang, Y. Li, M. Li, W. Che, and P. S. Yu, "Large language models meet nlp: A survey," *ArXiv preprint*, vol. abs/2405.12819, 2024.
- [91] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proc. of ICML*, vol. 202, 2023, pp. 19 730–19 742.
- [92] R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, and S. Taşlılar, "Introducing our multimodal models," 2023.
- [93] Y. Li, Y. Zhang, C. Wang, Z. Zhong, Y. Chen, R. Chu, S. Liu, and J. Jia, "Mini-gemini: Mining the potential of multi-modality vision language models," *ArXiv preprint*, vol. abs/2403.18814, 2024.
- [94] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge *et al.*, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *ArXiv preprint*, vol. abs/2409.12191, 2024.
- [95] Z. Liu, Y. Dong, Z. Liu, W. Hu, J. Lu, and Y. Rao, "Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution," *ArXiv preprint*, vol. abs/2409.12961, 2024.
- [96] Z. Lin, C. Liu, R. Zhang, P. Gao, L. Qiu, H. Xiao, H. Qiu, C. Lin, W. Shao, K. Chen *et al.*, "Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models," *ArXiv preprint*, vol. abs/2311.07575, 2023.
- [97] B. Shi, Z. Wu, M. Mao, X. Wang, and T. Darrell, "When do we not need larger vision models?" in *Proc. of ECCV*, 2025, pp. 444–462.
- [98] Y. Liu, B. Yang, Q. Liu, Z. Li, Z. Ma, S. Zhang, and X. Bai, "Textmonkey: An ocr-free large multimodal model for understanding document," *ArXiv preprint*, vol. abs/2403.04473, 2024.

- [99] J. Cha, W. Kang, J. Mun, and B. Roh, "Honeybee: Locality-enhanced projector for multimodal llm," in *Proc. of CVPR*, 2024, pp. 13 817–13 827.
- [100] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H. Hon, "Unified language model pre-training for natural language understanding and generation," in *Proc. of NeurIPS*, 2019, pp. 13 042–13 054.
- [101] Y. Hao, H. Song, L. Dong, S. Huang, Z. Chi, W. Wang, S. Ma, and F. Wei, "Language models are general-purpose interfaces," *ArXiv preprint*, vol. abs/2206.06336, 2022.
- [102] S. Liu, L. Fan, E. Johns, Z. Yu, C. Xiao, and A. Anandkumar, "Prismer: A vision-language model with multi-task experts," *Transactions on Machine Learning Research*, 2024.
- [103] J. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. L. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a visual language model for few-shot learning," in *Proc. of NeurIPS*, 2022.
- [104] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song *et al.*, "Cogvlm: Visual expert for pretrained language models," *ArXiv preprint*, vol. abs/2311.03079, 2023.
- [105] H. Zhao, Z. Cai, S. Si, X. Ma, K. An, L. Chen, Z. Liu, S. Wang, W. Han, and B. Chang, "MMICL: Empowering vision-language model with multi-modal in-context learning," in *Proc. of ICLR*, 2024.
- [106] L. Qin, Q. Chen, H. Fei, Z. Chen, M. Li, and W. Che, "What factors affect multi-modal in-context learning? an in-depth exploration," *ArXiv preprint*, vol. abs/2410.20482, 2024.
- [107] Z. Zhang, A. Zhang, M. Li, hai zhao, G. Karypis, and A. Smola, "Multimodal chain-of-thought reasoning in language models," *Transactions on Machine Learning Research*, 2024.
- [108] Q. Chen, L. Qin, J. Zhang, Z. Chen, X. Xu, and W. Che, "M³CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought," in *Proc. of ACL*, 2024, pp. 8199–8221.
- [109] Y. Zeng, X. Zhang, and H. Li, "Multi-grained vision language pre-training: Aligning texts with visual concepts," in *Proc. of ICML*, vol. 162, 2022, pp. 25 994–26 009.
- [110] X. Xu, T. Niu, Y. Xie, L. Qin, W. Che, and M.-Y. Kan, "Exploring multi-grained concept annotations for multimodal large language models," *ArXiv preprint*, vol. abs/2412.05939, 2024.
- [111] M. Shi, F. Liu, S. Wang, S. Liao, S. Radhakrishnan, D.-A. Huang, H. Yin, K. Sapra, Y. Yacoub, H. Shi *et al.*, "Eagle: Exploring the design space for multimodal llms with mixture of encoders," *ArXiv preprint*, vol. abs/2408.15998, 2024.
- [112] S. Tong, E. L. B. II, P. Wu, S. Woo, A. J. IYER, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang, X. Pan, R. Fergus, Y. LeCun, and S. Xie, "Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs," in *Proc. of NeurIPS*, 2024.
- [113] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.
- [114] W. Dai, N. Lee, B. Wang, Z. Yang, Z. Liu, J. Barker, T. Rintamaki, M. Shoeybi, B. Catanzaro, and W. Ping, "Nvlm: Open frontier-class multimodal llms," *ArXiv preprint*, vol. abs/2409.11402, 2024.
- [115] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu *et al.*, "Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling," *ArXiv preprint*, vol. abs/2412.05271, 2024.
- [116] Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, Z. Xie, Y. Wu, K. Hu, J. Wang, Y. Sun, Y. Li, Y. Piao, K. Guan, A. Liu, X. Xie, Y. You, K. Dong, X. Yu, H. Zhang, L. Zhao, Y. Wang, and C. Ruan, "Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding," 2024.
- [117] R. Huang, X. Ding, C. Wang, J. Han, Y. Liu, H. Zhao, H. Xu, L. Hou, W. Zhang, and X. Liang, "Hires-llava: Restoring fragmentation input in high-resolution large vision-language models," *ArXiv preprint*, vol. abs/2407.08706, 2024.
- [118] Z. Gan, Y. Chen, L. Li, C. Zhu, Y. Cheng, and J. Liu, "Large-scale adversarial training for vision-and-language representation learning," in *Proc. of NeurIPS*, 2020.
- [119] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao, "Vinvl: Revisiting visual representations in vision-language models," in *Proc. of CVPR*, 2021, pp. 5579–5588.
- [120] A. Karpathy and F. Li, "Deep visual-semantic alignments for generating image descriptions," in *Proc. of CVPR*, 2015, pp. 3128–3137.
- [121] D. Teney, P. Anderson, X. He, and A. van den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *Proc. of CVPR*, 2018, pp. 4223–4232.
- [122] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. of NeurIPS*, 2020.
- [123] H. Zhang, M. Gao, Z. Gan, P. Dufter, N. Wenzel, F. Huang, D. Shah, X. Du, B. Zhang, Y. Li *et al.*, "Mml: 5: Methods, analysis & insights from multimodal llm fine-tuning," *ArXiv preprint*, vol. abs/2409.20566, 2024.



Xiao Xu received the B.S. degree from Northeastern University, Shenyang, China, in 2020. He is currently working toward the Ph.D. degree with the Harbin Institute of Technology, Harbin, China. He has published research papers at international NLP/AI conferences and journals, such as ACL, EMNLP, AAAI, and TASLP. His research interests include natural language processing, vision-language learning and multimodal large language models.



language processing and large language models.

Libo Qin is a professor of School of Computer Science and Engineering, Central South University. He has published research papers at international NLP/AI conferences and journals, such as ACL, EMNLP, AAAI, and TASLP. His work has been selected as the Most Influential Paper by Paper Digest and won the Best Paper Award at the EMNLP2022 MMNLU Workshop. He has served as an Area Chair for EMNLP, NAACL, an Action Editor for ARR, and a Senior Program Committee Member for IJCAI. His research interests include natural



language processing and large language models.

Wanxiang Che is a professor of School of Computer Science and Technology, Harbin Institute of Technology. He is the vice director of Research Center for Social Computing and Information Retrieval. He is a young scholar of "Heilongjiang Scholar" and a visiting scholar of Stanford University. He is currently the vice director and secretary-general of the Computational Linguistics Professional Committee of the Chinese Information Society of China; Officer and Secretary of AACL Executive Board; a senior member of the China Computer Federation (CCF).



Min-Yen Kan is an Associate Professor and Vice Dean of Undergraduate Studies at the National University of Singapore. Min is an active member of the Association of Computational Linguistics (ACL), currently serving as a co-chair for the ACL Ethics Committee, and previously as the ACL Anthology Director (2008–2018). He is an associate editor for Information Retrieval and the survey editor for the Journal of AI Research (JAIR). His research interests include digital libraries, natural language processing and information retrieval. He was recognized as a

distinguished speaker by the ACM for natural language processing and digital libraries research. Specific projects include work in the areas of scientific discourse analysis, fact verification, full-text literature mining, lexical semantics and large language models. He is a senior member of the IEEE.