This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

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. Whether 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: knmnyn@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.

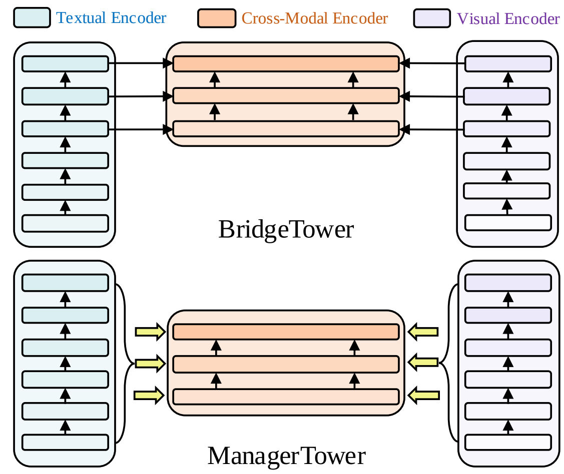

Fig. 1. A brief overview of BridgeTower and ManagerTower. Hollow arrows represent the transmission of multi-layer unimodal representations in ManagerTower, in contrast to the layer-by-layer transmission in BridgeTower.

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

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

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 VQAv2 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, whether 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.

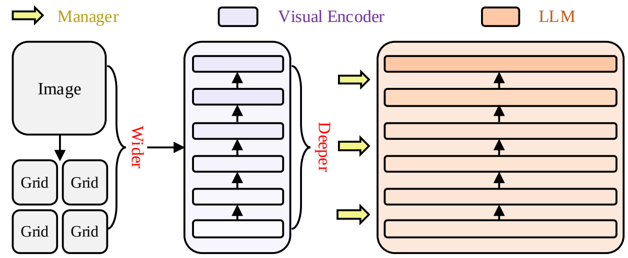

Fig. 2. Brief illustrations of LLaVA-OV-Manager. The base image and grids are encoded independently. Hollow arrows indicate the transmission of multi-layer visual representations aggregated by managers to the LLM at intervals.

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: $\mathbf{V}_\ell = \text{Encoder}^{\text{V}}_\ell(\mathbf{V}_{\ell-1}), \ell = 1 \dots L_{\text{V}}$, 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: $\mathbf{T}_\ell = \text{Encoder}^{\text{T}}_\ell(\mathbf{T}_{\ell-1}), \ell = 1 \dots L_{\text{T}}$, 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. $\text{Encoder}^{\text{C}}_\ell, \ell = 1 \dots L_{\text{C}}$ denotes the ℓ^{th} cross-modal layer, where L_{C} is the number of cross-modal layers. For brevity, it is computed as:

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

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

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

where $\mathbf{C}^{\text{V}}_\ell, \mathbf{C}^{\text{T}}_\ell$ are the visual and textual part of output representation of the ℓ^{th} layer, $\tilde{\mathbf{C}}^{\text{V}}_\ell, \tilde{\mathbf{C}}^{\text{T}}_\ell$ are inputs of each part. $\mathbf{C}^{\text{V}}_0, \mathbf{C}^{\text{T}}_0$ are initialized with the last-layer representations from unimodal encoders: $\mathbf{C}^{\text{V}}_0 = \mathbf{V}_{L_{\text{V}}}\mathbf{W}_{\text{V}}, \mathbf{C}^{\text{T}}_0 = \mathbf{T}_{L_{\text{T}}}\mathbf{W}_{\text{T}}$, where $\mathbf{W}_{\text{V}}, \mathbf{W}_{\text{T}}$ 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 $L_{\text{V}} = L_{\text{T}} = 12$, randomly-initialized cross-modal encoder with $L_{\text{C}} = 6$, and only top $\text{N} = 6$ unimodal layer representations are used.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

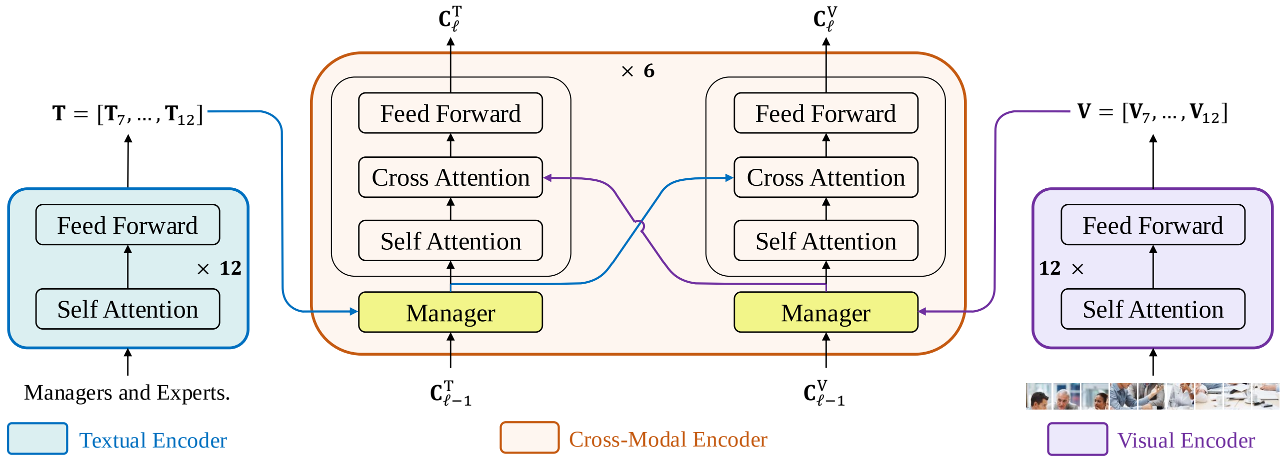

Fig. 3. An illustration of ManagerTower shows that each cross-modal layer includes a textual manager and a visual manager. Top $\mathrm{N}=6$ unimodal layer representations $\mathbf{T}, \mathbf{V} \in \mathbb{R}^{\mathrm{N}\times\mathrm{L}\times\mathrm{D}}$ along with the representations from the previous cross-modal layer $\mathbf{C}_{\ell-1}^{\mathrm{T}}, \mathbf{C}_{\ell-1}^{\mathrm{V}}, \ell=1\dots6$ are input into the textual manager $\mathcal{M}_\ell^{\mathrm{T}}$ and the visual manager $\mathcal{M}_\ell^{\mathrm{V}}$, 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:

$$\begin{aligned}&\mathcal{M}_\ell^{\mathrm{V}}(\mathbf{V}_7,\dots,\mathbf{V}_{12},\mathbf{C}_1^{\mathrm{V}},\dots,\mathbf{C}_{\ell-1}^{\mathrm{V}}) = \\ &\sum_{i=1}^{6}\mathbf{W}_i^{\mathrm{V},\ell}\odot\mathrm{LN}(\mathbf{V}_{i+6})+\sum_{i=1}^{\ell-1}\mathbf{W}_{i+6}^{\mathrm{V},\ell}\odot\mathrm{LN}(\mathbf{C}_i^{\mathrm{V}}),\end{aligned} \tag{4}$$

where $\mathcal{M}_\ell^{\mathrm{V}}$ represents the manager for the visual part of the ℓ^{th} cross-modal layer, and $\mathbf{W}^{\mathrm{V},\ell} \in \mathbb{R}^{(6+\ell-1)\times\mathrm{D}}$ is a learnable parameter matrix. $\odot$ denotes the element-wise product operation, and $\mathrm{LN}(\cdot)$ refers to Layer Normalization [21]. We then omit the superscript ${}^{\mathrm{V},\ell}$ of $\mathbf{W}$ for brevity. $\mathbf{W}$ can be seen as the learned aggregation weight and normalized by the $\mathrm{softmax}$ 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 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, and also learn the $\mathrm{softmax}$ 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.

[1]Details about pre-training and fine-tuning are described in Appendix B-G.

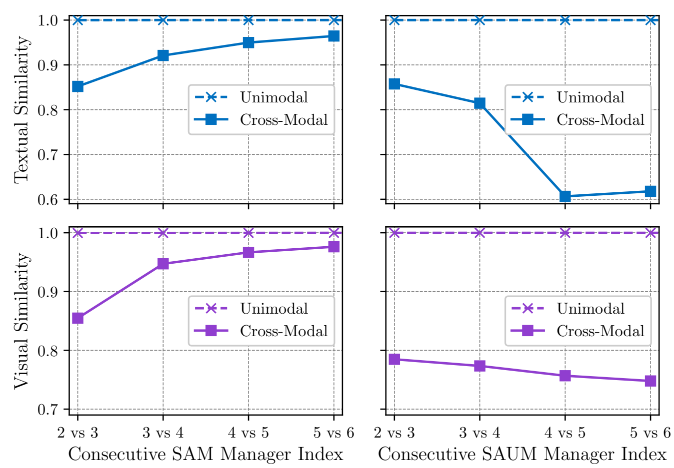

Fig. 4. Cosine similarity between the aggregated unimodal/cross-modal representations of each pair of consecutive textual/visual managers. The aggregated representations derived from Equation (4) can be divided into unimodal and cross-modal parts. For each part, we analyse the representations similarity between every two consecutive managers, modality-wise.

B. Static Aggregation Unimodal Manager (SAUM)

Since the aggregated representations derived from Equation (4) consist of an unimodal part and a cross-modal part, we calculate the cosine similarity between aggregated

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

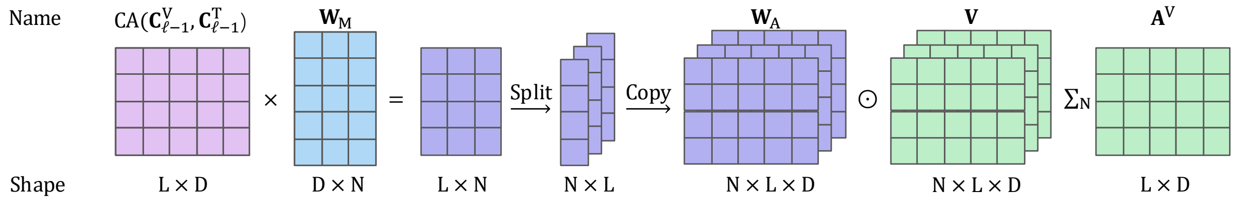

Fig. 5. An illustration of how the aggregated unimodal representations $\mathbf{A}^{\mathrm{V}} \in \mathbb{R}^{\mathrm{L}\times\mathrm{D}}$ are calculated in the visual AAUM. CA refers to the cross-attention mechanism. $\mathrm{N}=6$. For simplicity, we omit LN and $\mathrm{softmax}$ function.

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:

$$\begin{aligned}&\mathcal{M}^{\mathrm{V}}_{\ell}(\mathbf{V}_7,\ldots,\mathbf{V}_{12},\mathbf{C}^{\mathrm{V}}_{\ell-1}) = \\ &\sum_{i=1}^{6}\mathbf{W}_i \odot \mathrm{LN}\left(\mathbf{V}_{i+6}\right)+\mathbf{W}_{\mathrm{C}} \odot \mathrm{LN}(\mathbf{C}^{\mathrm{V}}_{\ell-1}),\end{aligned} \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{6\times\mathrm{D}}$ and $\mathbf{W}_{\mathrm{C}} \in \mathbb{R}^{1\times\mathrm{D}}$ are learnable parameter matrices, initialized with $\frac{1}{6}$ and 1 on average, respectively. The $\mathrm{softmax}$ 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:

$$\begin{aligned}&\mathcal{M}^{\mathrm{V}}_{\ell}(\mathbf{V}_7,\ldots,\mathbf{V}_{12},\mathbf{C}^{\mathrm{V}}_{\ell-1}) = \\ &\sum_{i=1}^{6}\mathbf{W}_{\mathrm{A},i} \odot \mathrm{LN}\left(\mathbf{V}_{i+6}\right)+\mathbf{W}_{\mathrm{C}} \odot \mathrm{LN}(\mathbf{C}^{\mathrm{V}}_{\ell-1}),\end{aligned} \tag{6}$$

$$\mathbf{W}_{\mathrm{A}} = \mathrm{softmax}(\mathrm{LN}(\mathbf{C}^{\mathrm{V}}_{\ell-1}) \times \mathbf{W}_{\mathrm{M}} + \epsilon), \tag{7}$$

where $\mathbf{W}_{\mathrm{M}} \in \mathbb{R}^{\mathrm{D}\times 6}$ denotes a linear projection layer. The generated aggregation weights $\mathbf{W}_{\mathrm{A}} \in \mathbb{R}^{6\times\mathrm{L}\times\mathrm{D}}$ can adaptively aggregate unimodal representations from different levels of pre-trained unimodal experts for each token. The $\mathrm{softmax}$ 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 $\mathbf{C}^{\mathrm{V}}_{\ell-1}$ in Equation (7) with the cross-modal fused query $\mathrm{CA}(\mathbf{C}^{\mathrm{V}}_{\ell-1}, \mathbf{C}^{\mathrm{T}}_{\ell-1})$ 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:

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

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

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

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

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

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 $\mathbf{W} \in \mathbb{R}^{6\times\mathrm{D}}$ are directly broadcast to $\mathbf{W}_{\mathrm{A}}$, and then they aggregate insights similarly to AAUMs.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

TABLE I
PERFORMANCE OF DIFFERENT TYPES OF MANAGERS AND QUERIES ON VQAV2 AND FLICKR30K. $\mathrm{R_{MEAN}}$ INDICATES THE MEAN RECALL METRICS FOR IMAGE–TEXT RETRIEVAL. BT DENOTES BRIDGETOWER.

Type	Visual Query	Weight	Test-Dev (%)	$\mathrm{R_{MEAN}}$ (%)
BT	-	$\mathrm{N} \times 1$	75.91	93.33
SAM	-	$\mathrm{N} \times 1$	76.19	93.57
	-	$\mathrm{N} \times \mathrm{D}$	76.18	93.73
SAUM	-	$\mathrm{N} \times 1$	76.38	93.75
	-	$\mathrm{N} \times \mathrm{D}$	76.55	93.82
AAUM	$\mathbf{C}^{\mathrm{V}}_{\ell-1}$	$\mathrm{N} \times \mathrm{L}$	76.52	93.84
	$\mathbf{C}^{\mathrm{V}}_{\ell-1}, \mathbf{C}^{\mathrm{T}}_{\ell-1}$	$\mathrm{N} \times \mathrm{L}$	**76.65**	**93.97**
Concat-Attention	$\mathbf{V}, \mathbf{C}^{\mathrm{V}}_{\ell-1}$	$\mathrm{N} \times \mathrm{L} \times \mathrm{D}$	76.38	93.78
	$\mathbf{V}, \mathbf{C}^{\mathrm{V}}_{\ell-1}, \mathbf{C}^{\mathrm{T}}_{\ell-1}$	$\mathrm{N} \times \mathrm{L} \times \mathrm{D}$	76.43	93.83
Cross-Attention	$\mathbf{C}^{\mathrm{V}}_{\ell-1}$	$\mathrm{N} \times \mathrm{L}$	76.41	92.15
	$\mathbf{C}^{\mathrm{V}}_{\ell-1}, \mathbf{C}^{\mathrm{T}}_{\ell-1}$	$\mathrm{N} \times \mathrm{L}$	76.45	92.61

TABLE II
PERFORMANCE OF BRIDGETOWER (BT) AND MANAGERTOWER (OURS) WITH DIFFERENT NUMBERS OF CROSS-MODAL LAYERS.

L_{C}	VQAv2 Test-Dev (%)		Flickr30K $\mathrm{R_{MEAN}}$(%)	
	BT	Ours	BT	Ours
2	74.86	75.47 (↑0.61)	92.45	93.31 (↑0.86)
3	75.33	76.04 (↑0.71)	92.50	93.41 (↑0.91)
4	75.74	76.26 (↑0.52)	92.76	93.59 (↑0.83)
6	75.91	**76.65** (↑0.74)	93.33	**93.97** (↑0.64)
8	75.89	76.47 (↑0.58)	93.03	93.65 (↑0.62)

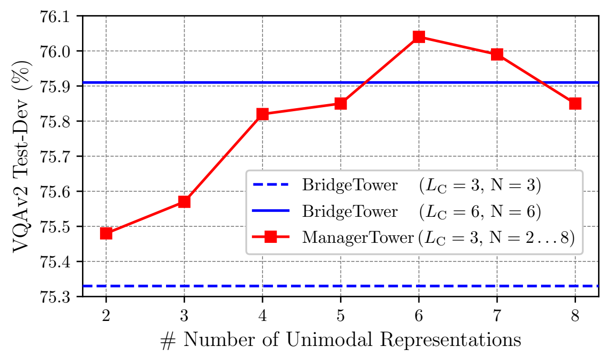

Fig. 6. VQAv2 Test-Dev Performance using different numbers of unimodal representations in ManagerTower ($L_{\mathrm{C}} = 3, \mathrm{N} = 2 \ldots 8$), where L_{C} is the number of cross-modal layers, and N is the number of top unimodal layer representations used in each bridge or manager.

IV. EXPLORATION ON TWO-TOWER VLM

A. Implementation Details

ManagerTower comprises a pre-trained textual encoder, $\mathrm{RoBERTa_{BASE}}$ 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 $\mathrm{N}=6$ visual layer representations $\mathbf{V} \in \mathbb{R}^{\mathrm{N} \times \mathrm{L} \times \mathrm{D}}$ from CLIP-ViT, different managers provide the aggregation weights that can be broadcast to $\mathbf{W}_{\mathrm{A}}$ for aggregating insights from pre-trained visual experts.

From the perspective of aggregation weights $\mathbf{W}_{\mathrm{A}}$, 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 Appendix B-E.

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 that **adaptive** token-level aggregation with a cross-modal fused query outperforms **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]

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 $L_{\mathrm{C}}=3$ is even better than that of BridgeTower with $L_{\mathrm{C}}=6$ ($76.04\% > 75.91\%,\ 93.41\% > 93.33\%$).

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 $\mathrm{N}=6$. 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 $L_{\mathrm{C}} = 3$. As shown in Fig. 6, there exist two interesting observations: (i) ManagerTower ($L_{\mathrm{C}} = 3, \mathrm{N} = 3$) outperforms BridgeTower ($L_{\mathrm{C}} = 3, \mathrm{N} = 3$), suggesting that when the same number

[2] Further elaboration of the relationship between different types of managers can be found in Appendix B-A&B-B.

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 (%) Test-Dev	VQAv2 (%) Test-Std	SNLI-VE (%) Dev	SNLI-VE (%) Test	NLVR^2 (%) Dev	NLVR^2 (%) Test-P	Flickr30K (%) IR@1	Flickr30K (%) TR@1
Base-size models pre-trained on 4M public data									
ViLT$_{\text{BASE}}$ [24]	4M	71.26	-	-	-	75.70	76.13	64.4	83.5
UNITER$_{\text{BASE}}$ [25] $*$	4M	72.70	72.91	78.59	78.28	77.18	77.85	72.52	85.90
UNIMO$_{\text{BASE}}$ [26]	4M	73.79	74.02	80.00	79.10	-	-	74.66	89.70
ALBEF$_{\text{BASE}}$ [27] $*$	4M	74.54	74.70	80.14	80.30	80.24	80.50	82.8	94.3
METER-Swin$_{\text{BASE}}$ [6]	4M	76.43	76.42	80.61	80.45	82.23	82.47	79.02	92.40
VLMo$_{\text{BASE}}$ [28]	4M	76.64	76.89	-	-	82.77	83.34	79.3	92.3
METER-CLIP$_{\text{BASE}}$ [6]	4M	77.68	77.64	80.86	81.19	82.33	83.05	82.22	94.30
BridgeTower$_{\text{BASE}}$ [7]	4M	78.66	78.73	81.11	81.19	81.85	83.09	85.83	94.73
ManagerTower$_{\text{BASE}}$ **(Ours)**	4M	**79.39**	**79.15**	**81.26**	**81.44**	**82.81**	**83.34**	**86.56**	**95.64**
Models pre-trained on more data and/or with larger size									
UNITER$_{\text{LARGE}}$ [25] $*$	4M	73.82	74.02	79.39	79.38	79.12	79.98	75.56	87.30
UNIMO$_{\text{LARGE}}$ [26]	4M	75.06	75.27	81.11	80.63	-	-	78.04	89.40
ALBEF$_{\text{BASE}}$ [27] $*$	14M	75.84	76.04	80.80	80.91	82.55	83.14	85.6	95.9
SimVLM$_{\text{BASE}}$ [29]	1.8B	77.87	78.14	84.20	84.15	81.72	81.77	-	-
BLIP$_{\text{BASE}}$ [30] $*$	129M	78.24	78.17	-	-	82.48	83.08	87.3	97.3
SimVLM$_{\text{LARGE}}$ [29]	1.8B	79.32	79.56	85.68	85.62	84.13	84.84	-	-

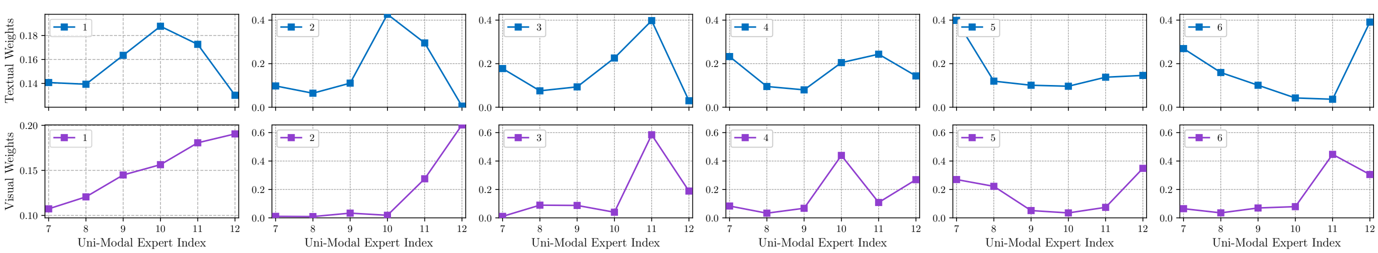

Fig. 7. A visualization of aggregation weights of textual and visual AAUMs in each cross-modal layer after VLP. The X-axis shows the index of the unimodal expert, and the legend shows the index of the cross-modal layer.

of unimodal layer representations are introduced, ManagerTower 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 $\mathrm{N} > 6$. 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.[3]

D. Visualization of Aggregation Weights

We delve into managers by visualizing the average aggregation weights $\mathbf{W}_{\mathrm{A}}$ 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

[3]Comparison of computational budget can be found in Appendix B-D.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

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.[4] 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 autoreregressive objective for answer tokens. The maximum length of the input token sequence is set to 16384, and the image patch size is $14{\times}14$. 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.[5] Since AAUM achieves similar performance compared to SAUM in LLaVA-OV-Manager, we directly use SAUM for better efficiency in the following experiments.[6] For brevity, the ℓ^{th} LLM layer with SAUM computes as:

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

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

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

Equation (14) is an optimized version of SAUM for MLLM. The original version does not work well in our preliminary

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

[5]Ablation study for the default setting can be found in Section V-D2.

[6]Discussions about managers in the MLLM can be found in Appendix A-A.

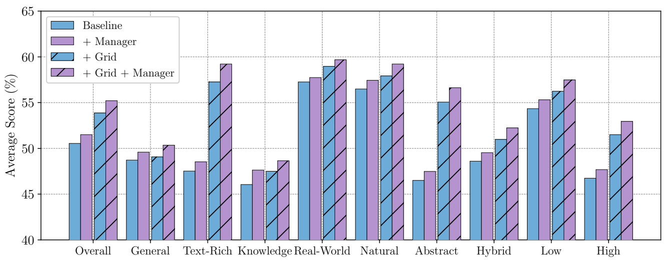

Fig. 8. Zero-shot performance of four baselines on 20 datasets. The overall average score and the average score of each capability category are shown.

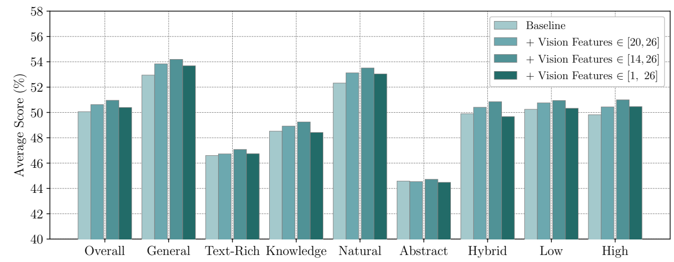

Fig. 9. Ablation study of visual representation selection on 9 datasets.

experiments, as the LLM in MLLM has been well pre-trained, rather than the random-initialized cross-modal module in ManagerTower. Hence, we remove the $\mathbf{W}_{\mathrm{C}}$, LN, and $\mathrm{softmax}$ 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.[7] 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".[8]

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.[9] 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 category of "General, Knowledge", Baseline+Manager even achieves better performance than Baseline+Grid with significantly lower computational cost.

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)	# FLOPs (G)	Training Time (ms/sample)	Inference Time (ms/sample)	Performance Overall (%)
Baseline	893.62	827.29	11.84	13.97	50.61
+ Manager	893.70	844.68 (×1.02)	12.22 (×1.03)	14.54 (×1.04)	51.67 (↑1.06)
+ Grid	893.62	1469.34 (×1.78)	51.95 (×4.39)	23.47 (×1.68)	53.87 (↑3.26)
+ Grid + Manager	893.70	1504.12 (×1.82)	54.17 (×4.58)	24.45 (×1.75)	**55.21** (↑4.60)

Table IV shows the computational budget and average overall performance of four MLLM 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 FLOPs ($\times 1.78$), training time ($\times 4.39$), inference time ($\times 1.68$) and performance ($\uparrow 3.26\%$). Whether with or without the multi-grid algorithm, managers only brings **negligible** parameter overhead (0.08M), FLOPs ($\times 1.02$), and computational cost ($\times 1.04$), but **significantly** improves performance ($\uparrow 1.06\%$ and $\uparrow 1.44\%$) on 20 datasets.[10]

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.

[7] https://github.com/EvolvingLMMs-Lab/lmms-eval
[8] More evaluation details can be found in Appendix C-F.
[9] Detailed results of each dataset can be found in Appendix C-G.
[10] $1504.12/1469.34 \approx 1.02$, $54.17/51.95 \approx 1.04$, $24.45/23.47 \approx 1.04$ and $55.21 - 51.67 = 1.44$.

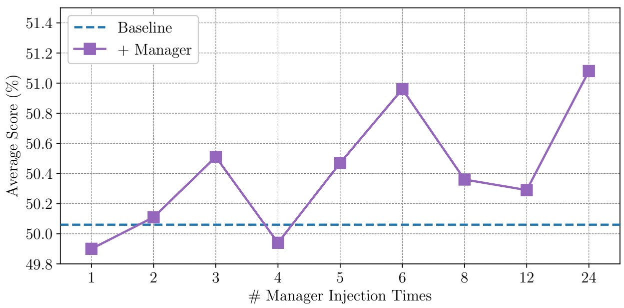

Fig. 10. Ablation study of manager injection times on 9 datasets.

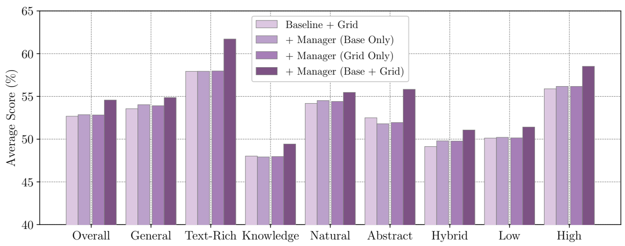

Fig. 11. Ablation study of how manager works with multi-grid on 9 datasets.

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 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.[11]

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 $L_{\mathrm{C}} = 24$, 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 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.

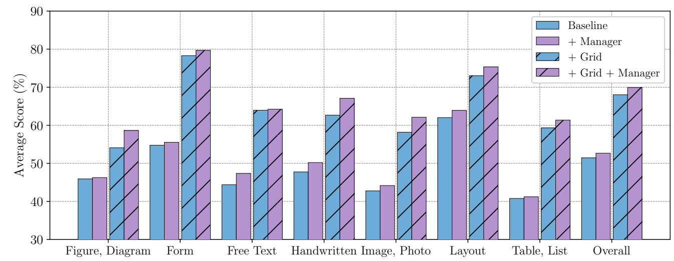

Fig. 12. Zero-shot performance of four baselines on DocVQA validation set.

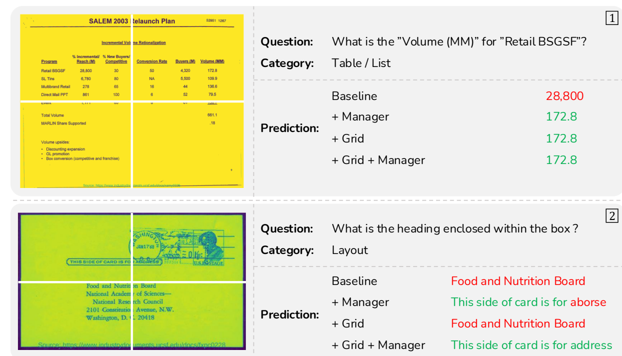

Fig. 13. Case studies of four baselines on DocVQA validation set. Red and green fonts represent incorrect and correct predictions, respectively. White lines indicate the boundaries of the image grids.

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.[12]

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 $\boxed{1}$ 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 $\boxed{2}$, 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

[11]Detailed explanations and visualizations are provided in Appendix C-B.

[12]Appendix C-C provides more cases on ScienceQA and OK-VQA.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

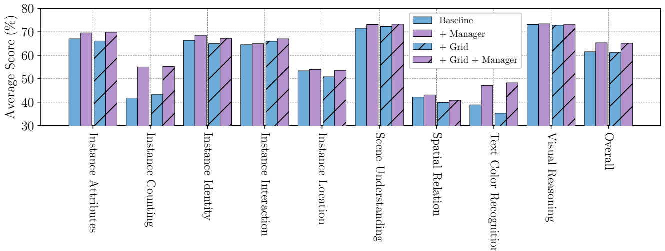

Fig. 14. Zero-shot performance of four baselines on SEED-Bench.

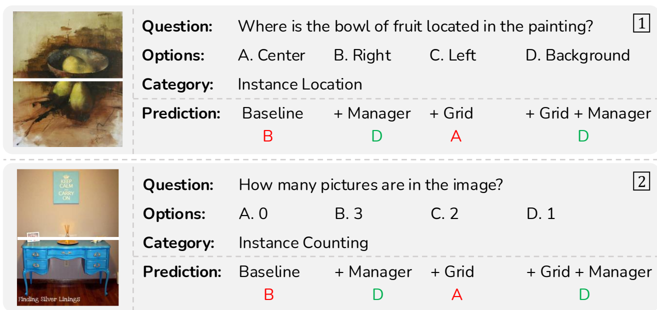

Fig. 15. Case studies of four baselines on SEED-Bench.

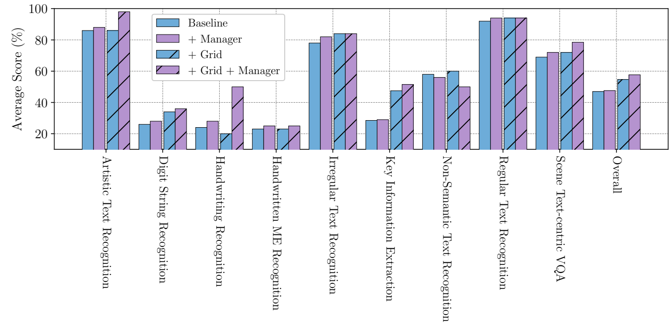

Fig. 16. Zero-shot performance of four baselines on OCRBench. "ME" in "Handwritten ME Recognition" is short for "Mathematical Expression".

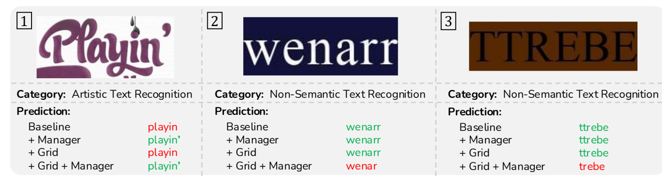

Fig. 17. Case studies of four baselines on OCRBench.

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.

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 are meaningless or 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, Baseline+Grid+Manager **incorrectly** identifies the **non-semantic** text "wenarr" and "ttrebe" as semantic text "wenar" and "trebe", respectively, where "wenar" is a surname of a person and "trebe" is a German noun for a runaway. Different levels of semantic knowledge brought by managers instead cause more interference, leading to performance degradation when work with the multi-grid algorithm in "Non-Semantic Text Recognition".

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 underling 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

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

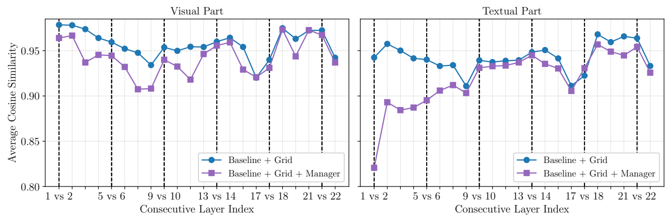

Fig. 18. Cosine similarity between output representations of consecutive layers. The dotted vertical lines indicate the layers where managers are injected, *i.e.*, # Layer Index= $[1, 5, 9, 13, 17, 21]$.

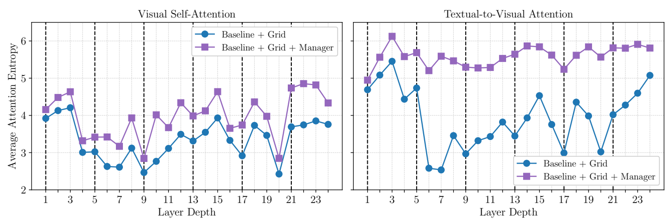

Fig. 19. Average entropy of attention weight distributions in each layer.

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 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.[13]

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

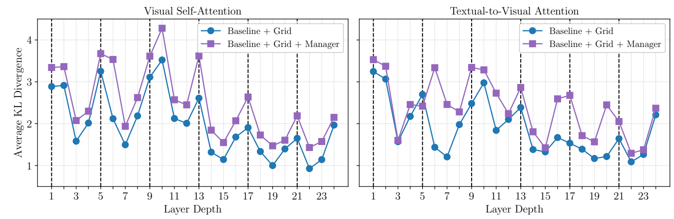

Fig. 20. Average KL divergence between attention weight distributions of attention heads in each layer.

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 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].[14] 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

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

[14] Detailed discussion of the related work for multimodal fusion from the perspective of architecture can be found in Appendix A-C.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3578266

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], which is a resource-efficient way but may bring semantic ambiguity [59], [97]. 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.

References

[1] X. Xu, B. Li, C. Wu, S.-Y. Tseng, A. Bhiwandiwalla, 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. Soricut, "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. Taşırlar, "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] 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.

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.

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). He received the AAAI 2013 Outstanding Paper Honorable Mention Award. His research interests include natural language processing and large language models.

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.