# ADEM-VL: Adaptive and Embedded Fusion for Efficient Vision-Language Tuning

Zhiwei Hao\*, Jianyuan Guo\*, Li Shen, Yong Luo, Han Hu, and Yonggang Wen

Abstract-Recent advancements in multimodal fusion have witnessed the remarkable success of vision-language (VL) models, which excel in various multimodal applications such as image captioning and visual question answering. However, building VL models requires substantial hardware resources, where efficiency is restricted by two key factors: the extended input sequence of the language model with vision features demands more computational operations, and a large number of additional learnable parameters increase memory complexity. These challenges significantly restrict the broader applicability of such models. To bridge this gap, we propose ADEM-VL, an efficient visionlanguage method that tunes VL models based on pretrained large language models (LLMs) by adopting a parameter-free crossattention mechanism for similarity measurements in multimodal fusion. This approach only requires embedding vision features into the language space, significantly reducing the number of trainable parameters and accelerating both training and inference speeds. To enhance representation learning in fusion module, we introduce an efficient multiscale feature generation scheme that requires only a single forward pass through the vision encoder. Moreover, we propose an adaptive fusion scheme that dynamically discards less relevant visual information for each text token based on its attention score. This ensures that the fusion process prioritizes the most pertinent visual features. With experiments on various tasks including visual question answering, image captioning, and instruction-following, we demonstrate that our framework outperforms existing approaches. Specifically, our method surpasses existing methods by an average accuracy of 0.77% on ScienceQA dataset, with reduced training and inference latency, demonstrating the superiority of our framework. The code is available at https://github.com/Hao840/ADEM-VL.

Index Terms—Multimodal fusion; Parameter-free Cross-attention; PEFT, LLMs

# I. INTRODUCTION

RECENTLY, vision-language (VL) modeling has made significant progress [1], [2]. The main goal of these models is to make predictions based on inputs from both visual and textual data. By leveraging the powerful prompt-following capabilities of pretrained autoregressive large language models (LLMs), fine-tuned VL models achieve remarkable results on

Zhiwei Hao and Han Hu are with the School of Information and Electronics, Beijing Institute of Technology, Beijing, China. E-mail: {haozhw, hhu}@bit.edu.cn.

Jianyuan Guo is with School of Computer Science, Faculty of Engineering, The University of Sydney, Sydney, Australia. E-mail: jianyuan.guo@sydney.edu.au.

Li Shen is with School of Cyber Science and Technology, Sun Yat-sen University, Shenzhen, China. E-mail: mathshenli@gmail.com.

Yong Luo is with School of Computer Science, Wuhan University, Wuhan, China. E-mail: luoyong@whu.edu.cn.

Yonggang Wen is with School of Computer Science and Engineering, Nanyang Technological University, Singapore. E-mail: ygwen@ntu.edu.sg. Corresponding to Han Hu.

various tasks and even surpass well-educated humans in some

Existing approaches for multimodal fusion can be roughly categorized into two categories based on whether the fusion is achieved in the feature space or input space. The first category fuses visual information into the LLM at intermediate layers [3]. Specifically, a pretrained CLIP [4] vision tower is used to extract visual features from the input image. These features are then aligned with the dimensions of LLMs via a projector. Cross-attention modules facilitate the fusion, with text tokens in the LLM serving as queries and visual features serving as keys and values. Typically, each LLM layer requires a dedicated cross-attention module. These additional modules introduce a significant number of new parameters and increase the overall computational complexity of VL models. The other category directly fuses vision and text information in the input space of LLMs [5]. After obtaining visual features extracted by the CLIP model, these methods adopt a learnable projection model to achieve dimension alignment. However, the visual features are directly regarded as input tokens and concatenated at the head of text tokens rather than being fused by complicated schemes. This kind of models require a twostage training process: in the first stage, only the projector is trained, and in the second stage, both the projector and the LLM are trained together. Although such models avoid the additional parameters introduced by cross-attention modules, they fine-tune the entire LLM in the second training stage. This results in billions of trainable parameters, demanding substantial storage space and computational resources. Furthermore, the extended input sequence length increases both training and inference costs quadratically. To enable broader and more costeffective applications of VL models while minimizing carbon emissions, it is essential for fine-tuned models to be more efficient in terms of both parameters and computation. While previous work [6] on parameter-efficient fine-tuning (PEFT) for pure LLMs can be adapted for VL models, these methods often fail to fully integrate visual information and typically do not achieve optimal performance.

To bridge this gap, we propose ADEM-VL, an efficient adaptive and embedded fusion framework for vision-language tuning at the intermediate layers of pretrained LLMs. To reduce the number of parameters in the cumbersome cross-attention modules of existing VL models, we first explore their simplified variants. By reformulating standard cross-attention into an abstract form, we replace the parameterized similarity measurement with a parameter-free approach. This eliminates most of the learnable parameters in these modules, except for a shared low-rank projector that aligns the dimensions for

embedding vision features. We further enhance the representation learning ability of our parameter-free fusion module by introducing multiscale visual features through pooling and concatenation operations. Unlike existing approaches that require multiple invocations of the vision encoder [7], [8], our method only requires a single forward pass, resulting in negligible additional computational cost. Furthermore, considering that not all image features contribute equally to prediction, we propose an adaptive fusion scheme. For each text token, the corresponding image features with lower attention scores are dropped. This allows the text tokens to focus on more relevant visual features and avoid interference from irrelevant information. These designs significantly reduce the learnable parameters in both tuning and inference stage while requiring no increase in input length. With enhanced multimodal fusion schemes, our ADEM-VL achieves superior performance while maintaining high efficiency. Our method differs from existing methods that introduce adapter modules into LLMs and fuse multimodal information in the input space [9], [10]. While these methods consider parameter efficiency, they often ignore computational complexity. In contrast, our framework can achieve both parameter and computationally efficient fusion for VL models.

To evaluate the proposed framework, we conduct experiments on three different vision-language tasks. On the visual question answering task using ScienceQA dataset, our finetuned LLaMA-13B model achieves 94.55% average accuracy, outperforming existing approaches by 0.77% while being 15% and 3% faster in training and inference stage, respectively. Additionally, the results on the image captioning task and the instruction-following task demonstrate comparable performance to existing methods, further validating the effectiveness of our proposed framework. The contributions of our paper can be summarized as follows:

- We propose ADEM-VL, an adaptive and embedded fusion framework for vision-language tuning. ADEM-VL is highly efficient in terms of both parameters and computational cost during training and inference.
- By reformulating the standard cross-attention and replacing the parameterized similarity measurement with a parameter-free one, we significantly reduce the number of trainable parameters.
- We introduce pooling and concatenation operations to generate multiscale visual features with a single invocation of vision encoder, resulting in negligible additional computational cost.
- We implement an adaptive fusion scheme that discards irrelevant image features with lower attention scores, enabling text tokens to focus on more pertinent visual information and reducing interference.

#### II. RELATED WORK

## A. Multimodal architectures.

The advancements in the field of multimodal learning have been significantly shaped by breakthroughs in natural language processing, particularly with the advent of attention-based models [11]. Inspired by BERT, numerous studies have

incorporated masked modeling to develop multimodal systems [12]–[21]. Additionally, some multimodal models have employed contrastive learning techniques during their training processes [4], [22]–[26]. Both approaches have demonstrated remarkable performance across a range of multimodal tasks.

2

Recently, driven by the success of LLMs in language, researchers have increasingly concentrated on developing autoregressive multimodal models to harness the exceptional capabilities of LLMs. A prior work, Flamingo [3], bridged the gap between pure language models and vision models through a cross-attention mechanism. Specifically, this model integrates gated cross-attention layers before each block of the language model, facilitating interaction between vision and language modalities. Flamingo surpassed previous models and showcased the potential of autoregressive architectures for multimodal learning. This achievement inspired a series of subsequent VL models, which can be broadly categorized into two classes based on how they incorporate vision information into the pretrained language model.

One class of models integrates vision information into the internal layers of the language model. Within this category, vision information is first extracted using a pretrained vision encoder, followed by a learnable projection layer or a resampling layer. The multimodal fusion is then achieved by inserting cross-attention layers into the LLMs, either before the the self-attention layers [3], [27] or after the self-attention layers [28]. In addition to standard cross-attention layers, some architectures employ customized layers specifically designed for fusion [9], [29]–[32].

Another class of models introduces vision information at the input stage of the language model. In this approach, the extracted vision information is directly concatenated with the language tokens before being input into the LLMs. Typically, a projection layer is employed to align the vision feature space with the language token space. This projection layer can be a simple linear layer [5], [7], [33], [34], a resampling layer [35], [36], or other customized layers designed to facilitate effective fusion [37]–[41]. Additionally, some works use tokenized source images as input instead of extracted features [42], [43]. This design enables their capability for image generation.

To facilitate a better understanding of VL model performance, various benchmarks and toolkits have been introduced, such as MME [44], MMB [45], MMMU [46], MMT [47], and AVIBench [48]. These benchmarks have driven significant advancements in the architectures of VL models.

Although existing multimodal architectures have achieved remarkable performance, inserting cross-attention modules at intermediate layers of LLMs introduces a substantial number of additional trainable parameters. On the other hand, concatenating vision features with language tokens extends the length of the input sequence, significantly increasing the computational resources needed for inference. Consequently, developing more efficient multimodal architectures is appealing.

## B. Parameter-efficient fine-tuning

Since the advent of LLMs, there has been a sharp increase in the number of model parameters. While these models deliver

3

outstanding performance, fine-tuning them for downstream tasks is challenging due to their massive size. To address this issue, various PEFT approaches have been developed.

PEFT techniques for language models can be broadly classified into two categories based on whether they introduce new trainable parameters into the pretrained models. The first category includes approaches that do not introduce any new parameters. The main idea behind these methods is to selectively fine-tune the existing parameters within the original model [49]–[51]. For example, BitFit [49] focuses on training only the bias values in the pretrained model, significantly reducing the number of trainable parameters required during fine-tuning. The second category of PEFT techniques involves adding additional trainable parameters while keeping the pretrained model fixed [6], [52]-[56]. Adapter [52] and LoRA [6] are two typical methods in this category. The Adapter approach introduces small linear layers after each sub-block of the pretrained model and makes only these new layers trainable. LoRA uses low-rank matrix decomposition to parameterize the pretrained weight matrices. During inference, these decomposed matrices can be merged back into the original model, thus incurring no additional inference cost.

Generally, most PEFT approaches designed for language models can be directly applied to the training of multimodal architectures. In addition to these, there are also PEFT methods specifically tailored for multimodal learning [9], [10], [29], [57]–[59]. VL-Adapter [57] and VL-PET [58] introduce vision information at the input stage and fine-tune pretrained language models with an encoder-decoder architecture using adapter modules. LLaMA-Adapter [9], [29] and LaVIN [10] focus on PEFT for multimodal learning using the more advanced LLaMA architecture by designing specific adapter modules. MenVP [59] accomplishes multimodal fusion by augmenting the linear layers of the pretrained language model with vision information.

However, while these approaches effectively reduce the number of learnable parameters in multimodal models, many of them extend the input sequence length by combining vision information with language tokens, leaving the challenge of high computational complexity during inference unresolved. Multimodal fusion at intermediate layers via cross-attention incurs lower computational complexity compared to extending input sequences. Yet, parameter-efficient techniques for optimizing cross-attention modules remain largely unexplored. In this paper, we aim to bridge this gap by developing methods that achieve both parameter and computational efficiency in multimodal fusion.

#### III. METHOD

To achieve effective and efficient multimodal fusion for VL models, we propose a framework named ADEM-VL, as shown in Figure 1. To ensure efficiency in both tuning and inference stage, we simplify the standard cross-attention module through removing trainable parameters and thus reduce its computational requirements. To deliver promising performance with the simplified fusion scheme, we further introduce an effective multiscale visual information generating scheme

to provide the language model with ample visual knowledge and an adaptive fusion scheme to help the model focus only on informative visual information. In this section, we first introduce the related background knowledge (III-A), followed by the simplified parameter-free cross-attention (III-B), the multiscale prompting (III-C), and the adaptive fusion (III-D). Finally, we present the overall framework of our ADEM-VL based on these proposed designs (III-E).

#### A. Background

Vision-Language models. VL models are designed to process both language and visual inputs to perform multimodal inference tasks such as visual question answering (VQA). Recently, due to the tremendous success of LLMs, there have been efforts to leverage these exceptional models to build VL systems. Generally, there are two main classes of approaches for integrating vision information into pretrained LLMs. One class of methods achieves multimodal fusion by adopting cross-attention to merge LLM features and vision features at intermediate layers. The other class of methods concatenates vision tokens with text tokens at the input space. However, these approaches either introduce considerable additional parameters or increase inference costs due to the introduction of new modules or the extension of input length. Compared to directly increasing the input length, multimodal fusion via cross-attention offers greater potential for optimizing efficiency, as computational complexity increases quadratically with input length. Therefore, we adopt cross-attention-based fusion approaches as the starting point to explore a more efficient multimodal fusion method for VL models.

**Cross-attention for fusion.** Existing methods that use cross-attention for visual-language fusion typically employ the language token  $X_l \in \mathbb{R}^{L \times d}$  as the query and the visual features  $X_v \in \mathbb{R}^{N \times d'}$  as the key and value to facilitate information interaction [3], [60]. This process can be formulated as:

$$\begin{aligned} \mathbf{X}\mathbf{A}\mathbf{t}\mathbf{t}\mathbf{n}(\boldsymbol{X}_{l}, \boldsymbol{X}_{v}) &= \operatorname{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{T}}{\sqrt{d_{k}}}\right)\boldsymbol{V} \\ &= \operatorname{softmax}\left(\frac{\boldsymbol{X}_{l}\boldsymbol{W}_{Q}\boldsymbol{W}_{K}^{T}\boldsymbol{X}_{v}^{T}}{\sqrt{d_{k}}}\right)\boldsymbol{X}_{v}\boldsymbol{W}_{V}\boldsymbol{W}_{o}^{T}, \end{aligned} \tag{1}$$

where  $W_Q$ ,  $W_K$ , and  $W_V$  are learned projection matrices for the query, key, and value, respectively.  $W_o^T$  is the output projection matrix, and  $d_k$  is the dimensionality of each key vector.

Following Equation 1, the adoption of cross-attention presents a remarkable drawback, where each cross-attention module contains four projection matrices, introducing a substantial number of additional trainable parameters. This is particularly significant when multimodal fusion is performed at each layer of the pretrained LLM using individual cross-attention modules [3]. To improve the efficiency of the fusion process, we need to take a closer look at the cross-attention mechanism and modify it to be more efficient.

# B. Parameter-free cross-attention

To improve the efficiency of cross-attention in VL models, we begin our analysis by obtaining an abstract form of

Fig. 1. Comparison of different vision-language tuning frameworks: (a) Methods that directly extend the input space of the language model with extracted vision features. (b) Methods that fuse vision information into the language model via cross-attention. (c) Our proposed ADEM-VL framework, which incorporates parameter-free cross-attention, multiscale visual prompting, and adaptive multimodal fusion designs. This approach ensures both parameter and computational efficiency while delivering promising performance.

Equation 1. In the cross-attention module, each query vector performs a dot product with all key vectors, followed by a softmax function to obtain the attention scores. The value vectors are then weighted and summed using these scores and subsequently multiplied by the output projection matrix to obtain the fused feature. Thus, Equation 1 can also be formulated as:

$$XAttn(\boldsymbol{X}_{l}, \boldsymbol{X}_{v})_{i} = \frac{\sum_{j} sim(\boldsymbol{Q}_{i}, \boldsymbol{K}_{j}) \boldsymbol{V}_{j}}{\sum_{j} sim(\boldsymbol{Q}_{i}, \boldsymbol{K}_{j})}, \tag{2}$$

where  $\sin(q, k) = \exp(\frac{q^T k}{\sqrt{d_k}})$  computs similarity between query vector q and key vector k. Generally, we can define different  $\sin(q, k)$  functions to obtain variants of standard cross-attention, such as polynomial attention, RBF kernel attention [61], or linear attention [62]. Since  $\sin(\cdot, \cdot)$  acts as a similarity metric, the only constraint is that it must produce non-negative values. Therefore, it can be any kernel function  $k(x,y): \mathbb{R}^{\|x\|+\|y\|} \to \mathbb{R}_+$ .

Based on kernel trick, the kernel function can be defined as  $k(x,y) = \phi(x)\phi(y)^T$ , where  $\phi$  is a projection function. On the other hand, in the general form of cross-attention, we have:  $\sin(\boldsymbol{Q}_i, \boldsymbol{K}_j) = \exp(\frac{\boldsymbol{X}_l \boldsymbol{W}_Q \boldsymbol{W}_K^T \boldsymbol{X}_v^T}{\sqrt{d_k}})$ . By selecting an appropriate non-parameterized projection function  $\phi$ , we can eliminate the need for parameters in the matrices  $\boldsymbol{W}_Q$  and  $\boldsymbol{W}_K$ . A possible class of such projections includes the modern activation function ReLU [63] and its modifications

like GeLU [64] and SiLU [65]. These are pointwise projections that can be computed efficiently while preserving strong representation extraction capabilities. To achieve better performance, variants of ReLU may output negative values when the input is close to zero. However, we find that this limitation has little effect on performance, and the SiLU activation function can serve as an adequate replacement for  $\phi$ .

Then, to further improve the parameter efficiency of the cross-attention module, we adopt identity matrices for the value vector projection and the output projection. This is based on the assumption that the powerful pretrained LLMs can work effectively with coarsely fused information to accomplish multimodal tasks. Finally, the embedded parameter-free cross-attention can be formulated as:

$$XAttn(\boldsymbol{X}_{l}, \boldsymbol{X}_{v}) = \phi(\boldsymbol{X}_{l})\phi(\boldsymbol{X}_{v})^{T}\boldsymbol{X}_{v},$$
(3)

where  $\phi(\cdot) = \text{SiLU}(\cdot)$ .

In VL models, the original feature sizes of the vision tower and the LLM are usually not the same. Existing work [5] adopts a learnable projection matrix to align their dimensions. We follow a similar idea but decompose the projection matrix into low-rank decompositions to reduce the parameter count. This process embeds visual features into the language feature space, enabling parameter-free attention-based fusion, which explains the "embedded" aspect in our method's name. Note

that the dimension alignment serves as a pre-processing step for visual features, meaning that the aligned features are directly utilized in parameter-free cross-attention modules across all layers of the pretrained LLM. As a result, we do not count its parameters as part of the cross-attention modules.

**computational complexity.** Suppose the dimension of feature vectors is d. In a standard cross-attention module with L query and N key/value vectors, the FLOPs of each original cross-attention module is  $2Ld^2+2Nd^2+2LNd$ , while in our simplified module, the FLOPs become 2LNd. Since LLMs usually have very large hidden dimensions d, the removal of the matrix projection requirement greatly reduces the computational complexity by approximately  $2Ld^2+2Nd^2$  FLOPs. Hence, our simplification of the cross-attention module helps improve both parameter and computational efficiency, facilitating PEFT of VL models.

#### C. Multiscale visual prompts

Multiscale features have been proven effective for improving model performance in various computer vision tasks [66], [67]. Given its success, we believe that introducing multiscale vision information in VL models could also help improve model performance.

To obtain multiscale visual information, we generate features of different scales by pooling operation based on the extracted features of the original image. Take CLIP encoder with an input resolution of 224×224 as an example, the input image is split into 256 patches, resulting in 256 extracted visual features, each corresponding to an original image patch. To provide the language model with high-level visual features, we merge adjacent tokens by pooling. Specifically, we first reshape the 256 one-dimensional tokens into a 16×16 twodimensional grid based on their positions in the original image. Then, we apply pooling operations with different kernel sizes to obtain multiscale visual features. By flattening the feature grid of each scale and concatenating them together, we obtain the final visual feature for LLMs. We compare different pooling configurations in terms of kernel type and size in our experiments.

In addition to the extracted features, the CLIP models also output a [cls] token, which acts as a more abstract global representation of the input image. To further utilize this token, we follow the input space prompting approaches [5] by concatenating it at the head of the text tokens. Unlike existing works that extend input length by dozens of tokens, we introduce only one additional token, which incurs negligible extra inference overhead, despite the computational complexity increasing quadratically with input length.

**Discussion.** The main restriction on the application of multiscale vision features on multimodal modeling is the limited availability of pretrained visual encoder models. Currently, CLIP is the most frequently used encoder. However, the publicly available well-pretrained CLIP models only support

input sizes of 224×224 or 336×336, preventing users from using inputs of other sizes. Existing works have explored designing approaches to inject multiscale vision information into language models [7], [8]. To bridge this gap, these works mainly divide and resize the original image and then process each patch individually to obtain multiscale visual features. For instance, Liu et al. [7] divided the original image into 2×2 sub-images and extract features from each resized subimage. Then, features of all sub-images and the original image are concatenated along with text tokens as the input of LLMs. Xu et al. [7] designed a more flexible method that can split image dynamically. However, these approaches require multiple forward passes through the vision encoder to process all subimages individually, without considering their relationships within the entire image. Furthermore, while these methods focus on achieving multimodal fusion in the input space, the use of multiscale information for cross-attention-based fusion remains underexplored. In comparison, our design generates features of different scales without invoking the vision encoder multiple times, thus maintaining efficiency.

# D. Adaptive multimodal fusion

The introduction of multiscale visual features provides the language model with more detailed information about the input image. However, not all of this information may be useful for each text query. For instance, for an inquiry about the left part of the input image, features corresponding to the right part of the image would be useless or could even mislead the model, preventing it from making correct predictions.

To address this challenge, we design an adaptive multimodal fusion scheme to help the text tokens dynamically extract visual information in the parameter-free cross-attention module. This allows them to focus more on informative visual features and mitigate interference from irrelevant ones. Following Equation 3, the attention score in our modified cross-attention is  $\phi(\mathbf{X}_l)\phi(\mathbf{X}_v\mathbf{W}')^T \in \mathbb{R}^{L\times N}$ . This matrix indicates the similarity score between each text token and each visual feature. Here we assume that for each text token, the visual features corresponding to larger similarity scores are more important for prediction, while those with lower scores provide less or even useless information. Based on this assumption, we can drop the less informative visual features to help improve performance. Specifically, we sort the attention score matrix over each row individually and mask the lowest values with a mask ratio of  $\gamma$ , indicating that the visual features with low scores will be dropped. As there is no restriction for the summation of attention scores for each text token to be 1, unlike the softmax function, dropping some visual features does not significantly change the attention mechanism in our modified cross-attention. By generating the mask individually for each row of the score matrix, the drop decision is adaptive to different text tokens and input images. This allows each text token to focus more on useful information, helping to achieve better fusion results.

 $<sup>^1</sup>$ For example, both LLaMA-7B and LLaMA2-7B have a hidden dimension of d=4096 and support input lengths of L=2048 and L=4096, respectively. However, in most VL settings, the input length l is usually much smaller than L (l << L).

#### E. Overall framework

```
Algorithm 1 ADEM-VL in PyTorch-style
Input: text x_t, image x_i, low-rank projection
      matrix W', scales S \in \mathbb{Z}_+^n, drop ratio \gamma
Output: fused feature X'_{l}
  1: X_l \leftarrow \text{Tokenizer}(x_t)
  2: \boldsymbol{X}_{v}, \boldsymbol{X}_{v,cls} \leftarrow \text{CLIP}(\boldsymbol{x}_{i})
  3: X'_l \leftarrow \operatorname{concat}([X_{v,cls}, X_l])
  4: X_v \leftarrow X_v W'
  5: X_v' \leftarrow X_v
  6: for s in S do
            X'_{v,s} \leftarrow \operatorname{pooling}(X_v, s)

X'_v \leftarrow \operatorname{concat}([X'_v, X'_{v,s}])

d for \triangleright Multiscale \ visual \ prompt \ (Sec.
  9: end for
       III-C
10: for layer in LLM do
            X'_i \leftarrow \text{layer}(X'_i)
11:
            attention A \leftarrow \text{silu}(X_l)\text{silu}(X'_n)^T
12:
       ▷ Parameter-free cross-attention (Sec. III-B)
13:
            A_{\text{sorted}} \leftarrow \text{torch.sort}(A, \text{dim}=1)
14:
            Index i \leftarrow \text{int}(\gamma \times \mathbf{A}.\text{size}(\text{dim}=1))
15
            threshold T \leftarrow A_{\text{sorted}}[:,i]
16:
            \operatorname{mask} M \leftarrow \operatorname{torch.ones\_like}(A)
17:
            M[\text{torch.where}(A < T)] \leftarrow 0
18:
            A \leftarrow A \cdot M \triangleright Adaptine fusion (Sec. III-D)
19:
            X_l' \leftarrow X_l' + AX_v'^T
20:
21: end for
```

With the parameter-free cross-attention, multiscale visual features, and adaptive fusion scheme, we can build our adaptive and embedded fusion framework (ADEM-VL). Specifically, following the common cross-attention-based framework [3], we insert our modified non-parameterized crossattention module at each layer of the target LLM. To align the dimension of visual feature and the [cls] token with that of the LLM while ensuring parameter efficiency, we adopt two low-rank projection matrices. Moreover, we adopt the idea of introducing lightweight adapters into the vision tower of CLIP and adding learnable positional embeddings E to the visual features [59], where the embeddings are shared across all cross-attention modules. The projection matrices, adapters for CLIP, and learnable positional embeddings consist of all trainable parameters in our framework. Finally, the fused feature can be denoted as:

$$X' = \alpha XAttn(X_l, \beta X_v + E)$$
 (4)

where  $\alpha$  and  $\beta$  are weighting hyperparameters for visual features and fused features. We compare the impact of these parameters in our experiments.

We provide details of our proposed ADEM-VL framework in PyTorch-style in Algorithm 1. Note that we add cross-attention modules after each LLM layer in this algorithm for clarification. In practice, we can insert these modules in the middle of LLM layers, and different configurations are compared in our experiments.

#### IV. EXPERIMENT

To evaluate the effectiveness of our ADEM-VL framework, we conduct a series of experiments on various datasets. In this section, we first introduce our experimental setup (IV-A). Then, we provide a quantitative comparison between different VL model tuning approaches (IV-B), followed by an ablation study to validate each component in our framework (IV-C). Finally, we present qualitative results of our trained model (IV-D).

#### A. Experimental setup

**Datasets.** We conduct experiments primarily on the ScienceQA dataset [68], a multimodal dataset for science question answering. The dataset consists of text-only and image-text samples covering 3 subjects, 26 topics, and 379 skills. These samples are split into train, validation, and test sets. We report the average accuracy on the test set. We also evaluate our method on the COCO Caption dataset [70] using the Karpathy split of the dataset [71]. We further train the model with instruction-following datasets. Specifically, we adopt Alpaca-52K [72], a text-only dataset generated by GPT-3.5 [73], and LLaVA-158K [5], a text-image dataset generated by GPT-4 [1]. To evaluate the zero-shot performance of the trained model, we utilize several image question answering datasets and image understanding benchmarks, including VQAv2 [74], GQA [75], MME [44], MMB [45], and MMMU [46].

**Models.** Our framework fine-tunes pretrained language models for VL tasks. To achieve this, we adopt LLaMA [2] as the language model, which is a series of foundation language models ranging from 7B to 65B parameters. Here, we mainly conduct experiments using the variants containing 7B and 13B parameters. To obtain visual information, we adopt the CLIP [4] model with the ViT-L/14 [76] backbone as the image encoder.

**Optimization.** The optimization configuration of our model on each dataset follows that of LaVIN [10]. Specifically, we train the model for 20, 5, and 15 epochs on the ScienceQA, COCO Caption, and instruction-following datasets, respectively, with a global batch size of 32. The learning rate is initialized at 9e-3 and decreases according to a cosine schedule. We adopt features of two scales as the multiscale visual prompt: the original features and  $2\times$  downsampled features using average pooling. Across all our experiments, hyperparameters  $\alpha$ ,  $\beta$ , and  $\gamma$  are set to 0.1, 0.01, and 0.2, respectively.

#### B. Quantitative results

**ScienceQA.** We first evaluate our proposed ADEM-VL framework on the ScienceQA dataset and report the results in Table I. The baseline results are obtained from [10], [59]. From the results, our method achieves the best average accuracy among all the compared methods. Overall, PEFT methods outperform zero- or few-shot methods and regular methods that fine-tune the whole LLM. Within PEFT approaches, our ADEM-VL achieves better performance than the best performed baselines while maintaining a similar number of

TABLE I

EVALUATION RESULTS ON SCIENCEQA TEST SET. NAT = NATURAL SCIENCE, SOC = SOCIAL SCIENCE, LAN = LANGUAGE SCIENCE, TXT = TEXT

CONTEXT, IMG = IMAGE CONTEXT, NO = NO CONTEXT, G1-6 = GRADES 1-6, G7-12 = GRADES 7-12.

M-41 J	#Para	m		Subject	t	Cont	ext Mo	dality	Gr	ade	
Method	Trainable	LLM	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Average
Zero-/few-shot metho	ds										
Human [68]	-	-	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
GPT-3.5 [68]	-	-	74.64	69.74	76.00	74.44	67.28	77.42	76.80	68.89	73.97
GPT-3.5 [68]	-	-	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68	75.17
GPT-4 [1]	-	-	84.06	73.45	87.36	81.87	70.75	90.73	84.69	79.10	82.69
Full training method:	5										
UnifiedQA [68]	223M	-	71.00	76.04	78.91	66.42	66.53	81.81	77.06	68.82	74.11
$MM-CoT_{Base}$ [69]	223M	-	87.52	77.17	85.82	87.88	82.90	86.83	84.65	85.37	84.91
MM-CoT <sub>Large</sub> [69]	733M	-	95.91	82.00	90.82	95.26	88.80	92.89	92.44	90.31	91.68
LLaVA [5]	7B	7B	-	-	-	-	-	-	-	-	89.84
LLaVA [5]	13B	13B	90.36	95.95	88.00	89.49	88.00	90.66	90.93	90.90	90.92
PEFT methods with	LLaMA										
LLaMA-Adapter [9]	1.8M	7B	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
LLaVA-LoRA [59]	4.4M	7B	91.70	94.60	86.09	91.25	90.28	88.64	91.52	89.65	90.85
LaVIN [10]	3.8M	7B	89.25	94.94	85.24	88.51	87.46	88.08	90.16	88.07	89.41
LaVIN [10]	5.4M	13B	90.32	94.38	87.73	89.44	87.65	90.31	91.19	89.26	90.50
MemVP [59]	3.9M	7B	94.45	95.05	88.64	93.99	92.36	90.94	93.10	93.01	93.07
MemVP [59]	5.5M	13B	95.07	95.15	90.00	94.43	92.86	92.47	93.61	94.07	93.78
ADEM-VL	4.5M	7B	95.52	95.39	89.18	95.36	93.95	90.94	93.87	93.80	93.85
ADEM-VL	5.5M	13B	96.00	94.94	91.27	95.45	93.95	93.03	94.46	94.73	94.55
PEFT methods with LLaMA2											
MemVP [59]	3.9M	7B	93.12	94.60	89.27	92.86	91.13	91.15	92.51	92.29	92.43
ADEM-VL	4.5M	7B	95.74	94.83	90.00	95.50	93.75	91.78	94.16	93.87	94.06

trainable parameters. Specifically, with LLaMA-7B as the LLM, our method achieves 93.85% average accuracy, outperforming the second-best baseline by 0.78%, with only 0.6M more parameters. When the LLM is LLaMA-13B, the average accuracy of our method increases to 94.55%, which is 0.77% higher than the second-best result. Note that unlike the implementation in MemVP [59], where LLaMA-13B uses longer positional embeddings than LLaMA-7B and introduces more additional parameters, the trainable parameters in our method increase more slowly with the increase of LLM size compared to MemVP.

Additionally, we conduct experiments using the more advanced pretrained LLM, LLaMA2-7B [2], applying the same VL tuning recipe as with LLaMA-7B. MemVP, when based on LLaMA2-7B, exhibits poorer performance compared to its LLaMA-7B counterpart. In contrast, our ADEM-VL method outperforms the LLaMA-7B variant, highlighting its potential to achieve even superior performance when equipped with more powerful pretrained LLMs. To ensure a fair comparison with existing approaches, which predominantly use LLaMA models as the pretrained LLM, we adhere to this setting in all subsequent experiments.

COCO caption. We further evaluate the proposed framework on the image captioning task using the COCO caption dataset, following the implementation of BLIP [60] and baseline results in LaVIN [10]. As shown in the results in Table II, the proposed method achieves comparable performance while requiring significantly fewer trainable parameters compared to large-scale pre-training approaches such as BLIP [60]

TABLE II
EVALUATION RESULTS ON COCO CAPTION USING THE KARPATHY TEST
SPLIT WITH LLAMA-13B AS THE LANGUAGE MODEL. #T. = TRAINABLE
PARAMETERS. \*PEFT METHODS.

Method	#T.	BLEU-4	CIDEr
ClipCap [77]	-	33.5	113.1
VisionLLM-H [78]	-	32.1	114.2
BLIP [60]	583M	40.4	136.7
BLIP-2 [35]	188M	43.7	145.3
*LLaMA-Adapter V2 [29]	14M	36.2	122.2
*LaVIN [10]	5.4M	37.8	131.7
*ADEM-VL	5.5M	38.5	133.2

and BLIP-2 [35]. Among PEFT approaches, our ADEM-VL outperforms existing baselines by a significant margin, with a 0.7 improvement in the BLEU-4 score and a 1.5 increase in the CIDEr score. These results further demonstrate the superiority of the proposed VL fusion framework.

Instruction following. To evaluate our framework on the instruction-following task, we fine-tune a LLaMA-13B model using a combination of the Alpaca-52K and LLaVA-158K datasets, following the implementation of LaVIN [10]. We start by assessing the trained model on the MME benchmark, comparing our results with those reported by LaVIN. Table III presents the corresponding results. Compared to the full fine-tuning method LLaVA, all PEFT methods achieve better performance, with their results showing a trade-off between the two metrics. The baseline, Prompt-Aware Adapter, achieves the best overall performance but introduces 256 extra input

TABLE III

EVALUATION RESULTS ON THE MME BENCHMARK WITH LLAMA-13B AS THE LANGUAGE MODEL. MME-C AND MME-P MEASURE THE PERCEPTION AND COGNITION ABILITIES OF THE MODEL, RESPECTIVELY. EXTRA TOKENS REFER TO THE NUMBER OF ADDITIONAL TOKENS PROCESSED BY THE LLM BEYOND THE STANDARD TEXT TOKENS. #T. = TRAINABLE PARAMETERS. \*PEFT METHODS.

Method	#Trainable param	#Extra tokens	MME-P	MME-C
LLaVA [5]	13B	256	502.8	214.6
*Prompt-Aware Adapter [79]	-	256	1375.0	289.3
*MiniGPT-4 [36]	-	256	866.5	292.1
*LayerNorm [80]	325M	256	929.3	254.3
*LayerNorm-simp. [80]	0.4M	256	824.3	221.1
*LLaMA-Adapter [9]	14M	-	972.6	248.9
*LaVIN [10]	5.4M	7	963.6	249.6
*ADEM-VL	5.5M	1	966.2	270.7

TABLE IV

COMPARISON AMONG DIFFERENT VL MODELS ON MORE IMAGE UNDERSTANDING TASKS. \*BASELINE RESULTS EVALUATED THROUGH OUR IMPLEMENTATION USING THE OFFICIAL CHECKPOINT.

Method	#Para	m	Image	QA	Benchmark		
Method	Trainable	LLM	VQAv2	GQA	MMB	MMMU	
Full training methods							
LLaVA [5]	13B	13B	-	-	34.1	32.3	
mPLUG-Owl2 [81]	8.2B	8.2B	79.4	56.1	64.5	-	
InternLM-XComposer2 [32]	7B	7B	-	-	79.6	42.0	
MoE-LLaVA- $1.6B \times 4$ -Top2 [82]	6.4B	6.4B	76.7	60.3	60.2	-	
PEFT methods							
MiniGPT-4 [36]	_	13B	-	-	23.0	-	
LaVIN [10]	5.4M	13B	68.6*	$48.8^{*}$	56.7*	35.0*	
ADEM-VL	4.5M	7B	71.7	52.4	52.4	34.2	
ADEM-VL	5.5M	13B	73.5	56.0	58.4	38.3	

tokens to the LLM. Since the LLM accounts for most of the computation in a VL model, this greatly increases both training and inference overhead. Our efficiency analysis in Table V illustrates this point in more detail. Compared to the efficient baseline LaVIN, which uses only 7 extra tokens, our proposed framework outperforms across both metrics with only 1 additional token, showcasing its exceptional generalization ability.

To further evaluate the instruction-following model, we assess its performance on additional image understanding tasks, comparing it against both full training methods and PEFT approaches. As shown in Table IV, across all four image question-answering tasks and benchmark toolkits, our LLaMA-13B-based ADEM-VL model achieves performance comparable to full training methods. Although there is a marginal performance gap, our approach involves significantly fewer trainable parameters than full training methods, greatly reducing the resource consumption for fine-tuning VL models. When compared with PEFT methods, ADEM-VL with LLaMA-13B delivers the best performance with a similar number of trainable parameters, further highlighting its effectiveness and efficiency.

**Efficiency.** To demonstrate the superiority of our method in delivering more efficient VL models across various PEFT methods, we report the FLOPs required to generate one token in Table V. The FLOPs is measured under a text sequence length of 256, which corresponds to a total context length of 512 when 256 additional image tokens are used for methods that fuse VL information at the input stage. Compared to

the most effective baseline, our method incurs only a slightly higher FLOPs, primarily due to the extended sequence length introduced by the [CLS] token. As demonstrated in our ablation study, this token significantly enhances performance while adding negligible computational overhead.

Additionally, we evaluate the training and inference speed of fine-tuned models and compare the results with existing approaches. Following the configuration in [59], we set the batch size to 4 during training and 64 during inference, using NVIDIA A800 GPUs to measure the speed. The evaluated results showing that our trained model achieves the lowest training and inference latency when the parameter size in the original LLMs is the same. The overall training time further demonstrates the efficiency of our proposed method. For instance, the GPU time required by the ADEM-VL framework to fine-tune a 13B parameter model is even lower than that of LaVIN when fine-tuning a 7B model.

In Table I and Table II, the results demonstrate that our method outperforms all other PEFT approaches while maintaining higher efficiency. For example, in the image captioning task, our proposed method surpasses LaVIN by 0.7 and 1.5 points in BLEU-4 and CIDEr scores, respectively, while being 36% faster in training and 10% faster in inference. This indicates that our design is highly effective for tuning efficient VL models without sacrificing performance. Although the performance is slightly lower than some input-stage fusion approaches, which require significantly more computational resources, our approach offers a promising alternative for

TABLE V

Training and inference speed of different approaches. Memory-saving or speed-up approaches such as checkpointing and flashattention are not adopted. FLOPs are estimated for generating a single new token with a text sequence length of 256. Experiments on COCO captioning and instruction-following were not implemented in the original papers of LLaVA-LORA and MemVP, so the overall training time for these tasks is unavailable.

Method	#Pa T.	ram LLM	FLOPs		(s/batch) Inference		training time (GP COCO caption	
LLaVA-LoRA [59]	4.4M	7B	110.44T	0.49	3.42	8.8	-	-
LaVIN [10]	3.8M	7B	56.19T	0.39	2.06	6.8	12.7	211.4
MemVP [59]	3.9M	7B	54.81T	0.28	1.88	5.1	-	-
MemVP [59]	5.5M	13B	132.76T	0.46	3.07	8.1	-	-
ADEM-VL	4.5M	7B	54.93T	0.25	1.86	4.3	8.0	134.8
ADEM-VL	5.5M	13B	133.26T	0.39	2.97	6.9	12.5	212.9

TABLE VI
ABLATION STUDY OF EACH MODULE IN OUR ADEM-VL FRAMEWORK WITH LLAMA-7B AS THE LANGUAGE MODEL.

Satting	#Trainable	Subject		Context Modality			Grade		Avaraga	
Setting	#11amable	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Average
Baseline	3.4M	93.49	95.05	88.21	92.85	91.28	90.92	92.50	92.35	92.45
+ [cls] token	4.0M	93.70	95.00	88.46	93.19	91.85	90.63	92.37	93.05	92.61
+ Parameter-free xattn	4.0M	94.60	95.65	89.00	94.56	93.19	90.89	93.42	93.27	93.37
+ Multiscale VP	4.5M	95.10	95.50	88.50	94.87	93.48	90.66	93.61	93.21	93.47
+ Adaptive fusion	4.5M	95.52	95.39	89.18	95.36	93.95	90.94	93.87	93.80	93.85

TABLE VII

COMPARISON OF DIFFERENT LOCATIONS FOR INSERTING CROSS-ATTENTION MODULES WITH LLAMA-7B AS THE LANGUAGE MODEL. "QUERY FROM" INDICATES WHICH FEATURES OF THE LANGUAGE MODEL SERVE AS INPUTS TO THE CROSS-ATTENTION MODULES, WHILE "ADD TO" INDICATES WHERE THE OUTPUT OF THESE MODULES IS FUSED INTO THE FEATURES OF THE LANGUAGE MODEL BY ADDITION.

Query from	Add to	Average
MHSA (in)	MHSA (in)	92.19
MHSA (in)	MHSA (out)	93.18
MHSA (out)	MHSA (out)	92.00
MLP (in)	MLP (in)	91.77
MLP (in)	MLP (out)	93.85
MLP (out)	MLP (out)	92.27

TABLE VIII COMPARISON OF DIFFERENT NON-PARAMETERIZED LINEAR PROJECTION IN EQUATION 3 WITH LLAMA-7B as the language model.

Projection	formula	Average
None	$x \to x$	92.16
Softmax	$x \to \operatorname{softmax}(x)$	79.42
ReLU	$x \to \text{relu}(x)$	91.99
ELU	$x \to \mathrm{elu}(x)$	92.45
SiLU	$x \to \mathrm{silu}(x)$	93.85
SiLU (positive)	$x \to \operatorname{silu}(x) - \min(x)$	38.58

achieving intermediate-layer fusion. It is more efficient than input-stage counterparts as it avoids the quadratically increased computational burden associated with longer input sequences. We believe our framework can inspire future research in designing more efficient VL models through intermediate-layer fusion strategies.

# C. Ablation study

To examine the effectiveness of each component in our proposed framework, we conduct a series of ablation studies. We begin with a coarse look at the whole framework and then investigate each module more carefully.

Ablation of the overall framework. The proposed framework consists of three main modules: parameter-free crossattention, multiscale visual prompts, and adaptive multimodal fusion. We study the effectiveness of each module by designing ablation experiments on the ScienceQA dataset with LLaMA-7B and report the results in Table VI. The baseline result, obtained without any of the three modules, achieves an average accuracy of 92.45%. In this setting, we adopt the cross attention in Equation 3 with identity projection  $\phi: x \to x$  as the projection function. On this basis, we first introduce the [cls] token into the input space, which slightly improves performance to 92.61% while introducing minimal additional resource consumption. Then, we adopt the design in our parameter-free cross-attention by using SiLU activation as the projection, which helps the trained model achieve an average accuracy of 93.37%. Further, we introduce multiscale visual information into the multimodal fusion process, which helps the accuracy reach 93.47%. Finally, by dropping the least important visual features based on the adaptive fusion module, the model achieves an average accuracy of 93.85%. These results demonstrate that each component in our proposed framework contributes to improved performance of fine-tuned VL models. The combination of all components leads to the best result, demonstrating the effectiveness of our design.

**Locations for fusion.** Cross-attention modules can be placed at different parts of the pretrained LLM for multimodal fusion. Previous works mainly position them either before or after the self-attention modules [83]. Here, we consider a more

general setting by decoupling the location of obtaining input and fusing output of cross-attention modules, allowing both the head and tail of the multilayer perceptron (MLP) module as possible positions. Results in Table VII compare different location configurations for fusion. From the results, simply placing cross-attention before or after self-attention or MLP modules is not as effective as positioning them to span these modules, i.e., input and output being located at the head and the tail of these modules, respectively. Compared to spanning across the self-attention module, making the cross-attention modules use the same input as the MLP modules and fusing them at the output layer leads to the best performance.

Non-parameterized projection. In the parameter-free cross-attention module, a fixed, non-parameterized projection is adopted to replace the learnable weight matrices and softmax operation, reducing both parameters and computational complexity. To investigate the impact of different projection options, we compare several choices in Table VIII. From the results, the identity projection achieves an average accuracy of 92.16%. When applying more complex projections, there is no guarantee of better performance, such as softmax and ReLU. In contrast, when adopting SiLU as the projection, the accuracy becomes 93.85%. Since the theoretical analysis result requires the output space of projection to be in  $\mathbb{R}_+$ , we also consider a variant of SiLU that satisfies this by subtracting the minimum value of the input feature. However, this projection leads to significantly worse performance, indicating such a requirement is not necessary in practice.

Multiscale visual prompts. Multiscale visual features are adopted as prompts in our framework. We compare the impact of different scales and approaches for obtaining such features through experiments and present the results in Table IX. We first examine the impact of scale by using two-dimensional average pooling as the downsampling approach. We apply downsampling with kernel sizes of 2 and 4, resulting in 64 and 16 downsampled visual features, respectively. When only visual features from a single scale are used, downsampling to either 64 or 16 leads to lower performance compared to using the full set of 256 visual features, with performance degradation becoming more pronounced as the number of visual features decreases. When multiscale features are obtained by concatenating multiple single-scale features, using a downsampling kernel size of 2 enhances performance of the model. However, as the kernel size increases to 4, the performance tends to decline in most cases. The only exception occurs when both  $2\times$  and  $4\times$  downsampled features are used together, yielding slightly better results than using either scale alone. We speculate that aggressive downsampling may produce overly coarse features, hindering the model to learn from fine-grained details. Therefore, a significant disparity between scales should be avoided when adopting multiscale visual prompts. Additionally, we compare max pooling with average pooling under the best configuration of scale. The results demonstrate that average pooling is a more preferable approach for downsampling.

**Hyperparameters.** In our proposed framework, there are three main hyperparameters: weighting parameters  $\alpha$  and  $\beta$  in Equation 4, and the drop ratio  $\gamma$ . To investigate the

TABLE IX
COMPARISON OF DIFFERENT DOWNSAMPLING METHODS AND SCALES IN
GENERATING MULTIMODAL VISUAL PROMPTS WITH LLAMA-7B AS THE
LANGUAGE MODEL.

Down sample	Size	Average
None	256	93.70
Avg. pooling	64	92.82
Avg. pooling	16	91.65
Avg. pooling	concat(64,16)	93.24
Avg. pooling	concat(256,16)	93.65
Avg. pooling	concat(256,64)	93.85
Avg. pooling	concat(256,64,16)	93.59
Max pooling	concat(256,64)	93.55

TABLE X INTEGRATION WITH DIFFERENT INPUT-STAGE FUSION SCHEMES WITH LLAMA-7B AS THE LANGUAGE MODEL.

Visual i #Visual tokens	Average	
0	×	92.97
0	✓	93.85
64	X	92.47
64	✓	92.86
256	X	89.86
256	✓	90.17

best configurations of these parameters, we compare different settings, and the results are shown in Figure 2. The results indicate that the best performance is achieved when  $\alpha=0.1$  and  $\beta=0.01.$  Regarding the drop ratio  $\gamma,$  the performance improves as  $\gamma$  increases, reaching its peak when  $\gamma=0.2.$  However, further increasing  $\gamma$  shows marginal benefits for adaptive fusion.

Compatibility with input-stage fusion methods. Our method paves the way for building efficient VL models by focusing on feature fusion at intermediate layers. To examine its compatibility with input-stage fusion methods, we conducted experiments comparing the introduction of visual features at different scales during the input stage. As shown in Table X, the best performance is achieved when only the [cls] token is used as the extra input token alongside text tokens, which is the default setting in our framework. However, when additional visual features are introduced at the input stage, there is a significant performance drop, regardless of the scale of visual features adopted. We speculate that this information has already been incorporated during the intermediate-layer fusion process, and further incorporating it at the input stage may interfere with the fusion, leading to a decline in performance. Additionally, extending the input length increases computational complexity quadratically. Thus, integrating our ADEM-VL framework directly with inputstage fusion approaches may not be an optimal solution.

#### D. Qualitative results

**Image captioning.** To better understand the adaptive multimodal fusion module in our ADEM-VL framework, we provide quantitative results obtained with our trained model on the

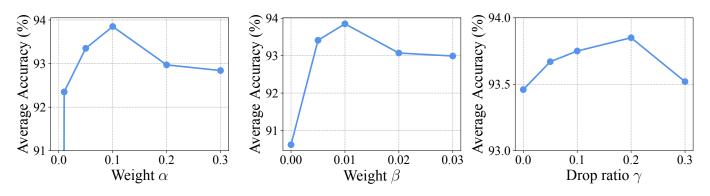


Fig. 2. Comparison of different hyperparameter settings in the ADEM-VL with LLaMA-7B as the language model.



a herd of sheep grazing in the snow near a house

Fig. 3. Visualization of image captioning results with LLaMA-7B. In each row, the left figure is the original image, while the middle and right figures demonstrate the dropping decisions for features at two different scales.

image captioning task. The results are presented in Figure 3. Alongside the original input image, we also visualize the feature-dropping decisions made by the model. Specifically, for each image feature, each text token has an individual decision on whether to use it at each layer. We collect these decisions and count the frequency of usage for each token. The normalized frequency value is used as the alpha channel for each corresponding patch in the original image, with more transparent patches indicating less importance. Since our framework involves two scales of visual features, we provide the dropping decisions for each scale separately. As shown in the figure, most of the dropped features are located in the background of the image, such as the tree in the bottom example. However, there are also several patches corresponding to the

subject of the image that have been frequently dropped. We speculate that this discrepancy is caused by the different visual perception mechanisms between VL models and humans and will investigate this in depth in future work.

**Instruction following.** We further evaluate the instruction-following data trained model through multimodal chat, where the model responds to an image and a corresponding instruction. Figure 4 presents the results. As shown, the model can recognize the subject in each image and respond correctly to the instruction. Additionally, the model can make proper inferences about the emotions depicted in the image, such as in the last two examples, even when these emotions are not directly demonstrated. These results showcase that our framework is not only efficient in both training and inference



**Instruction**: Describe this image.

**Response**: The image features a white airplane parked on a runway next to a pile of cargo containers. The airplane is a small jet, and it appears to be a passenger plane. The runway is surrounded by several trees, giving the scene a natural and serene atmosphere. The airplane is positioned near the center of the image, with the cargo containers located on the left side.



**Instruction**: what is in the center of this image?

**Response**: In the center of this image, there is a large clock tower with a cross on top. The clock tower is white and has a green dome on top, making it a prominent feature in the scene. The clock is visible on the tower, indicating the time for those around it. The clock tower is surrounded by trees, adding to its picturesque appearance.



**Instruction**: Write a diary after visiting this place.

Response: I visited a beautiful marina filled with numerous boats of various sizes. The boats were docked in the water, and the marina was surrounded by a lush green hillside. The scene was picturesque, and I felt like I was in a peaceful and serene environment. I walked around the marina, admiring the boats and the surrounding landscape. I noticed that some boats were docked closer to the shore, while others were further out in the water. I also noticed that some boats had names, and I wondered about the stories behind them. I felt a sense of calm and relaxation while I was there, and I'm sure I'll return to this marina in the future to enjoy the scenery and take a break from the hustle and bustle of daily life.



**Instruction**: Write an advertisement for this sport.

**Response:** Surfing is a thrilling and exhilarating sport that offers a unique experience for those who dare to ride the waves. It requires skill, balance, and strength, as well as a good understanding of the ocean's conditions. The image shows a man riding a surfboard on a wave, demonstrating the excitement and adrenaline that surfers experience while surfing. The sport is not only physically challenging but also mentally stimulating, as surfers must constantly adapt to changing wave conditions and the unpredictability of the ocean. Surfing is a sport for people of all ages and skill levels, and it can be enjoyed by individuals, families, and friends alike.

Fig. 4. Examples of zero-shot instruction-following tasks with LLaMA-7B.

but also highly competitive in practical scenarios.

# V. CONCLUSION

Building VL models upon pretrained LLMs has been proven effective but requires considerable computational resources. Facing the absence of efficient VL tuning methods, we propose a framework named ADEM-VL in this paper. The proposed framework consists of three main modules: parameter-free cross-attention for efficient multimodal fusion by removing most trainable parameters, multiscale visual prompts for ample visual information by downsampling, and adaptive multimodal fusion for target-focused learning by dynamically dropping useless features. We conduct experiments on visual question answering, image captioning, and instruction-following tasks, where quantitative results demonstrate the superiority of our proposed method over existing approaches. Additionally, we

provide visualization results obtained by our trained model, further demonstrating its effectiveness. One possible limitation of the proposed framework lies in the adaptive fusion module. The visualization of dropped features does not always align with the order of importance as perceived by humans. Further study should be conducted to improve this module or investigate the core differences between VL models and human perception to further enhance the performance of these models.

# ACKNOWLEDGEMENTS

This paper is supported by National Key Research and Development Program of China under No. 2021YFC3300200, and Joint Funds of the National Natural Science Foundation of China No. U2336211. This paper is supported by National Key Research and Development Program of China under No.

2021YFC3300200, and Joint Funds of the National Natural Science Foundation of China No. U2336211.

# REFERENCES

- J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023. 1, 6, 7
- [2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar et al., "Llama: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023. 1, 6, 7
- [3] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds et al., "Flamingo: a visual language model for few-shot learning," Advances in neural information processing systems, vol. 35, pp. 23716–23736, 2022. 1, 2, 3, 6
- [4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763. 1, 2, 6
- [5] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," Advances in neural information processing systems, vol. 36, 2024. 1, 2, 4, 5, 6, 7, 8
- [6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," arXiv preprint arXiv:2106.09685, 2021. 1, 3
- [7] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26296–26306. 2, 5
- [8] R. Xu, Y. Yao, Z. Guo, J. Cui, Z. Ni, C. Ge, T.-S. Chua, Z. Liu, M. Sun, and G. Huang, "Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images," arXiv preprint arXiv:2403.11703, 2024. 2, 5
- [9] 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 arXiv:2303.16199, 2023. 2, 3, 7, 8
- [10] G. Luo, Y. Zhou, T. Ren, S. Chen, X. Sun, and R. Ji, "Cheap and quick: Efficient vision-language instruction tuning for large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 2, 3, 6, 7, 8, 9
- [11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [12] 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 *European conference on computer vision*. Springer, 2020, pp. 104–120, 2
- [13] Z. Gan, Y.-C. Chen, L. Li, C. Zhu, Y. Cheng, and J. Liu, "Large-scale adversarial training for vision-and-language representation learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6616–6628, 2020.
- [14] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "Hero: Hierarchical encoder for video+ language omni-representation pre-training," arXiv preprint arXiv:2005.00200, 2020.
- [15] 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 Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16. Springer, 2020, pp. 121–137.
- [16] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances* in neural information processing systems, vol. 32, 2019.
- [17] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, "Flava: A foundational language and vision alignment model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15638–15650.
- [18] L. Zhu and Y. Yang, "Actbert: Learning global-local video-text representations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8746–8755.
- [19] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VI-bert: Pre-training of generic visual-linguistic representations," arXiv preprint arXiv:1908.08530, 2019.

- [20] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," arXiv preprint arXiv:1908.07490, 2019. 2
- [21] H. Zhang, L. Xu, S. Lai, W. Shao, N. Zheng, P. Luo, Y. Qiao, and K. Zhang, "Open-vocabulary animal keypoint detection with semanticfeature matching," *International Journal of Computer Vision*, pp. 1–18, 2024. 2
- [22] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," in *Proceedings* of the IEEE/CVF international conference on computer vision, 2021, pp. 1728–1738.
- [23] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*. PMLR, 2021, pp. 4904–4916.
- [24] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," *Advances in neural information processing systems*, vol. 34, pp. 9694–9705, 2021. 2
- [25] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9879–9889.
- [26] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li et al., "Florence: A new foundation model for computer vision," arXiv preprint arXiv:2111.11432, 2021.
- [27] B. Li, Y. Zhang, L. Chen, J. Wang, F. Pu, J. Yang, C. Li, and Z. Liu, "Mimic-it: Multi-modal in-context instruction tuning," arXiv preprint arXiv:2306.05425, 2023. 2
- [28] J. Cho, J. Lei, H. Tan, and M. Bansal, "Unifying vision-and-language tasks via text generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1931–1942.
- [29] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue *et al.*, "Llama-adapter v2: Parameter-efficient visual instruction model," *arXiv preprint arXiv:2304.15010*, 2023. 2, 3, 7
- [30] 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 arXiv:2311.03079, 2023.
- [31] D. Jia, J. Guo, K. Han, H. Wu, C. Zhang, C. Xu, and X. Chen, "Geminifusion: Efficient pixel-wise multimodal fusion for vision transformer," arXiv preprint arXiv:2406.01210, 2024.
- [32] X. Dong, P. Zhang, Y. Zang, Y. Cao, B. Wang, L. Ouyang, X. Wei, S. Zhang, H. Duan, M. Cao et al., "Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model," arXiv preprint arXiv:2401.16420, 2024. 2, 8
- [33] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu et al., "Palm-e: An embodied multimodal language model," arXiv preprint arXiv:2303.03378, 2023.
- [34] J. Chen, D. Zhu, X. Shen, X. Li, Z. Liu, P. Zhang, R. Krishnamoorthi, V. Chandra, Y. Xiong, and M. Elhoseiny, "Minigpt-v2: large language model as a unified interface for vision-language multi-task learning," arXiv preprint arXiv:2310.09478, 2023.
- [35] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping languageimage pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19730–19742. 2, 7
- [36] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," arXiv preprint arXiv:2304.10592, 2023. 2, 8
- [37] B. McKinzie, Z. Gan, J.-P. Fauconnier, S. Dodge, B. Zhang, P. Dufter, D. Shah, X. Du, F. Peng, F. Weers et al., "Mm1: Methods, analysis & insights from multimodal llm pre-training," arXiv preprint arXiv:2403.09611, 2024.
- [38] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi et al., "mplug-owl: Modularization empowers large language models with multimodality," arXiv preprint arXiv:2304.14178, 2023.
- [39] Y. Mu, Q. Zhang, M. Hu, W. Wang, M. Ding, J. Jin, B. Wang, J. Dai, Y. Qiao, and P. Luo, "Embodiedgpt: Vision-language pre-training via embodied chain of thought," Advances in Neural Information Processing Systems, vol. 36, 2024.
- [40] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, "Video-chatgpt: Towards detailed video understanding via large vision and language models," arXiv preprint arXiv:2306.05424, 2023. 2
- [41] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," arXiv preprint arXiv:2308.12966, 2023.

- [42] Y. Jin, K. Xu, L. Chen, C. Liao, J. Tan, B. Chen, C. Lei, A. Liu, C. Song, X. Lei et al., "Unified language-vision pretraining with dynamic discrete visual tokenization," arXiv preprint arXiv:2309.04669, 2023.
- [43] J. Lu, C. Clark, R. Zellers, R. Mottaghi, and A. Kembhavi, "Unified-io: A unified model for vision, language, and multi-modal tasks," in *The Eleventh International Conference on Learning Representations*, 2022.
- [44] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun *et al.*, "Mme: A comprehensive evaluation benchmark for multimodal large language models," *arXiv preprint arXiv:2306.13394*, 2023. 2, 6
- [45] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu *et al.*, "Mmbench: Is your multi-modal model an all-around player?" *arXiv preprint arXiv:2307.06281*, 2023. 2, 6
- [46] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun et al., "Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 9556–9567. 2, 6
- [47] K. Ying, F. Meng, J. Wang, Z. Li, H. Lin, Y. Yang, H. Zhang, W. Zhang, Y. Lin, S. Liu et al., "Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi," arXiv preprint arXiv:2404.16006, 2024.
- [48] H. Zhang, W. Shao, H. Liu, Y. Ma, P. Luo, Y. Qiao, and K. Zhang, "Avibench: Towards evaluating the robustness of large vision-language model on adversarial visual-instructions," arXiv preprint arXiv:2403.09346, 2024.
- [49] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameterefficient fine-tuning for transformer-based masked language-models," arXiv preprint arXiv:2106.10199, 2021. 3
- [50] D. Guo, A. M. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," arXiv preprint arXiv:2012.07463, 2020.
- [51] Y.-L. Sung, V. Nair, and C. A. Raffel, "Training neural networks with fixed sparse masks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24193–24205, 2021.
- [52] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799. 3
- [53] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," arXiv preprint arXiv:2110.04366, 2021. 3
- [54] R. Karimi Mahabadi, J. Henderson, and S. Ruder, "Compacter: Efficient low-rank hypercomplex adapter layers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1022–1035, 2021. 3
- [55] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1950–1965, 2022. 3
- [56] Y.-L. Sung, J. Cho, and M. Bansal, "Lst: Ladder side-tuning for parameter and memory efficient transfer learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12991–13005, 2022.
- [57] —, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5227–5237. 3
- [58] Z.-Y. Hu, Y. Li, M. R. Lyu, and L. Wang, "VI-pet: Vision-and-language parameter-efficient tuning via granularity control," in *Proceedings of* the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3010–3020.
- [59] S. Jie, Y. Tang, N. Ding, Z.-H. Deng, K. Han, and Y. Wang, "Memory-space visual prompting for efficient vision-language fine-tuning," arXiv preprint arXiv:2405.05615, 2024. 3, 6, 7, 8, 9
- [60] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*. PMLR, 2022, pp. 12 888–12 900. 3, 7
- [61] Y.-H. H. Tsai, S. Bai, M. Yamada, L.-P. Morency, and R. Salakhutdinov, "Transformer dissection: a unified understanding of transformer's attention via the lens of kernel," arXiv preprint arXiv:1908.11775, 2019.
- [62] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International conference on machine learning*. PMLR, 2020, pp. 5156– 5165. 4
- [63] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a

- cortex-inspired silicon circuit," *nature*, vol. 405, no. 6789, pp. 947–951, 2000. 4
- [64] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," arXiv preprint arXiv:1606.08415, 2016. 4
- [65] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural networks*, vol. 107, pp. 3–11, 2018. 4
- [66] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [67] Z. Hao, Y. Luo, Z. Wang, H. Hu, and J. An, "Cdfkd-mfs: Collaborative data-free knowledge distillation via multi-level feature sharing," *IEEE Transactions on Multimedia*, vol. 24, pp. 4262–4274, 2022. 5
- [68] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan, "Learn to explain: Multimodal reasoning via thought chains for science question answering," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2507–2521, 2022. 6, 7
- [69] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola, "Multimodal chain-of-thought reasoning in language models," arXiv preprint arXiv:2302.00923, 2023. 7
- [70] 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 arXiv:1504.00325, 2015.
- [71] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference* on computer vision and pattern recognition, 2015, pp. 3128–3137.
- [72] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," https://github.com/tatsu-lab/stanford\_alpaca, 2023. 6
- [73] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020. 6
- [74] 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 *Proceedings of the IEEE conference on* computer vision and pattern recognition, 2017, pp. 6904–6913. 6
- [75] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings* of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 6700–6709. 6
- [76] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020. 6
- [77] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," arXiv preprint arXiv:2111.09734, 2021. 7
- [78] W. Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu, J. Zhou, Y. Qiao et al., "Visionllm: Large language model is also an open-ended decoder for vision-centric tasks," Advances in Neural Information Processing Systems, vol. 36, 2024.
- [79] Y. Zhang, H. Fan, and Y. Yang, "Prompt-aware adapter: Towards learning adaptive visual tokens for multimodal large language models," arXiv preprint arXiv:2405.15684, 2024.
- [80] B. Zhao, H. Tu, C. Wei, J. Mei, and C. Xie, "Tuning layernorm in attention: Towards efficient multi-modal llm finetuning," arXiv preprint arXiv:2312.11420, 2023. 8
- [81] Q. Ye, H. Xu, J. Ye, M. Yan, A. Hu, H. Liu, Q. Qian, J. Zhang, and F. Huang, "mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 040–13 051. 8
- [82] B. Lin, Z. Tang, Y. Ye, J. Cui, B. Zhu, P. Jin, J. Zhang, M. Ning, and L. Yuan, "Moe-llava: Mixture of experts for large vision-language models," arXiv preprint arXiv:2401.15947, 2024.
- [83] S. N. Wadekar, A. Chaurasia, A. Chadha, and E. Culurciello, "The evolution of multimodal model architectures," arXiv preprint arXiv:2405.17927, 2024. 9