WINGS: Learning Multimodal LLMs without Text-only Forgetting

Yi-Kai Zhang[1,2,3*] **Shiyin Lu**[3] **Yang Li**[3] **Yanqing Ma**[3] **Qing-Guo Chen**[3]
Zhao Xu[3] **Weihua Luo**[3] **Kaifu Zhang**[3] **De-Chuan Zhan**[1,2] **Han-Jia Ye**[1,2†]
[1]School of Artificial Intelligence, Nanjing University
[2]National Key Laboratory for Novel Software Technology, Nanjing University
[3]Alibaba International Digital Commerce

Abstract

Multimodal large language models (MLLMs), initiated with a trained LLM, first align images with text and then fine-tune on multimodal mixed inputs. However, during the continued training, the MLLM catastrophically forgets the text-only instructions that the initial LLM masters. In this paper, we present WINGS, a novel MLLM that excels in both text-only and multimodal instructions. By examining attention across layers of MLLM, we find that *text-only forgetting* is related to the attention shifts from pre-image to post-image text. From that, we construct an additional Low-Rank Residual Attention (LoRRA) block that acts as the "modality learner" to expand the learnable space and compensate for the attention shift. The complementary learners, like "wings" on either side, are connected in parallel to each layer's attention block. The LoRRA mirrors the structure of attention but utilizes low-rank connections to ensure efficiency. Initially, image and text inputs are aligned with visual learners operating alongside the main attention, balancing focus on visual elements. Later, textual learners are integrated with token-wise routing, blending the outputs of both modality learners collaboratively. Our experimental results demonstrate that WINGS outperforms equally-scaled MLLMs in both text-only and visual question-answering tasks. WINGS with *compensation of learners* addresses text-only forgetting during visual modality expansion in general MLLMs.

1 Introduction

Large Language Models (LLMs) [34, 54, 93, 115] are making significant strides toward Artificial General Intelligence (AGI) systems. Multimodal Large Language Models (MLLMs), as a visual expansion of LLMs, have demonstrated astonishing performance in vision-related captioning [14, 16, 68], understanding [7, 33, 122], and reasoning [117, 127, 133]. Common MLLMs build upon powerful pre-trained LLMs that take mixed textual and visual tokens as inputs. The visual ones are acquired using an image encoder and a projector. We describe instructions processed by the LLM without images as *text-only instructions*. In comparison, *multimodal instructions* incorporate visual feature tokens into text-only sequences. Modality fusing at the token level provides a flexible and effective pipeline for training MLLMs to comprehend visual information [77, 80, 81]. However, training on multimodal instructions seems to impair the pre-existing profound knowledge, especially making MLLM forget how to respond to text-only instructions like the initial LLM [86, 90]. MLLM experiences a drastic performance decline on text-only evaluation. We term it as the *text-only forgetting* of MLLM.

*Work done during the internship at Alibaba International Digital Commerce.
†Corresponding author, email: yehj@lamda.nju.edu.cn.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

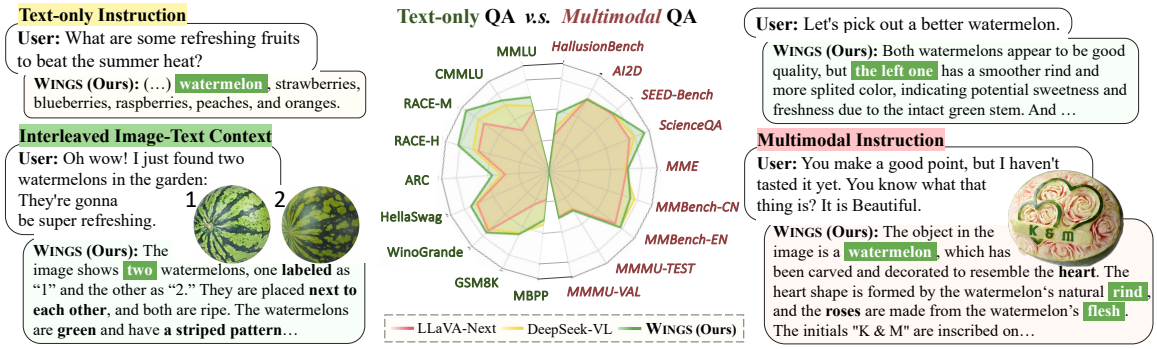


Figure 1: **Examples of text-only and multimodal conversations.** From left to right: Interacting with MLLM through *text-only* and *interleaved instructions*; Performance radar charts for WINGS, LLaVA-Next [81], and DeepSeek-VL [86] in *text-only* and *multimodal* QA tasks, with dark green indicating WINGS with the comprehensive performance; Interacting with *multimodal instructions*.

In practical applications, MLLMs also require engaging in text-only or interleaved conversations. As demonstrated in Figure 1, users often start with text-only inquiries and then, if not fully satisfied with the response, proceed to supplement questions with visual content. For multimodal instructions, MLLMs still rely on text to capture critical elements, as images may offer redundant information [15, 17, 85]. The first existing approaches replay extensive text-only or interleaved [61, 151] training data to mitigate catastrophic forgetting in MLLMs [72, 86, 90]. However, increasing training data incurs additional computational overhead and data collection challenges. Secondly, some applications [40] switch between LLM and MLLM based on whether images are included. This intuitive solution inevitably demands more deployment memory [1, 2] and is less cache-friendly in long vision-and-language interleaved conversations [42, 76, 101]. Therefore, it is crucial to train MLLM while preserving the text-only performance efficiently.

Given that the visual input tokens can be inserted at any position within the text sequence, we begin by examining the text before and after the inserted position to mark the impact of the visual part. Considering that MLLM's attention weights reflect the focus on tokens and influence the decision-making process, we first analyze the attention weights across each layer of the MLLM. Specifically, for each layer, we compute the attention weight proportion on all text tokens before and after the inserted image, termed as Layer-level Attention Weights (LAWS) of the before and after image text. From this, we examine the dynamic of attention across all layers as MLLM-Laws. Through training and sampling over 100 diverse MLLMs, we find that a well-trained model with superior text-only performance shows a positive correlation of MLLM-LAWS between the text segments before and after the image. This suggests that in a well-structured feature space, the main branch attention on text exhibits similar trends across layers, which is statistically linked to the semantic similarity of the text around the visual part. A closer similarity indicates minor disruption to MLLM's core attention, while a negative correlation shows that excessive focus on visual tokens shifts attention away from the text, significantly impacting MLLM-Laws.

Based on this observation, we propose WINGS, which introduces an extra module that acts as the boosted learner to compensate for the attention shift. We integrate complementary visual and textual learners in parallel at each layer's attention block, with visual learners enhancing focus on visual tokens and textual learners on text, respectively. In the first stage, visual features align with textual feature tokens, with all visual learners operating parallel to the main branch attention. The visual learners allocate some attention to visual tokens, mitigating the attention shift in the main branch. Subsequently, textual learners are integrated in parallel. We implement token-wise soft-routing based on shifted attention weights to harmonize the learning on visual and textual tokens. We design the Low-Rank Residual Attention (LoRRA) as the architecture for learners to ensure high efficiency. Figure 3 shows that the visual and textual learners on either side, like light feathers woven into "wings". Experiments show that our WINGS comprehensively achieves superior performance in text-only under the same training condition and exceeds other equal-level MLLMs on multimodal benchmarks. In addition, we construct the Interleaved Image-Text (IIT) benchmark with multi-turn

evaluations towards a general mixed-modality scenario. The samples are from text-only questions to strongly image-related conversations. WINGS achieve leading performance across various vision-relevance partitions. Overall, our contributions are as follows: (**1**) We claim and verify the text-only forgetting phenomenon of MLLM is related to the attention shift of cross-layer MLLM-LAWS before and after the image. (**2**) WINGS construct the visual and textual learners and introduce a router based on shifted attention weights for collaborative learning to compensate for attention shifts. (**3**) Experiments on text-only, visual-question-answering, and newly constructed Interleaved Image-Text (IIT) benchmarks demonstrate the comprehensive and versatile performance of WINGS.

2 A Closer Look at Attention Shift in Multimodal LLMs

In this section, we introduce the development from initialized LLM to MLLM. Next, we devise the MLLM-LAWS metric for representing attention shift and discuss the insights in building WINGS.

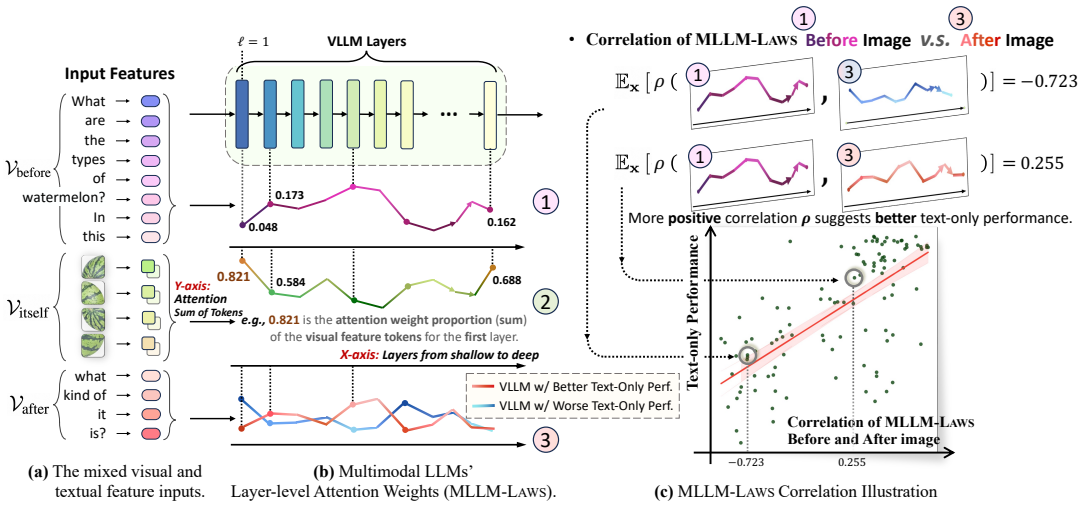
2.1 Granting Sight to Large Language Models

Large Language Models (LLMs). Even though existing Transformer-based [120] models [20, 83, 100, 134] like BERT [58] and OPT [142] have demonstrated profound language understanding capabilities, there has been a recent surge in powerful Generative Pre-trained Transformers (GPT) [10] under the auto-regressive language modeling paradigm. Both public [54, 55, 115, 116] and private [3, 93, 95, 112] solutions show remarkable progress in language comprehension and generation [91, 126]. These LLMs generally exceed a billion parameters, including pre-training [22, 32, 50, 56], supervised fine-tuning with instructions [26, 104, 110, 125], and reinforcement learning from human feedback [23, 96, 107, 152] on massive training data.

Multimodal LLMs (MLLMs). Integrating visual inputs into foundational LLMs to create MLLMs is becoming increasingly popular [18, 19, 63, 72, 136]. Unlike vision-centric multimodal frameworks [69, 137] such as CLIP series [99], MLLMs aim to align new modality features as the input of LLM with an additional encoder and perform multimodal question-answering [75, 80, 81, 128, 141, 150]. As illustrated in Figure 2 (a), it enables the combined training of mixed multimodal tokens, facilitating rapid deployment across various applications [24, 25, 45, 82, 122, 145]. One example of this pipeline is the LLaVA [80] series, which integrates a CLIP vision encoder with a linear projection to LLM Vicuna [21] and innovatively introduces instruction-following training data. Following this, some methods consider the richness of the vision-related training context [14, 46, 62], the scaled visual backbone [52, 73, 79], or the enhanced connectors [11, 124] to boost the visual effectiveness of MLLMs. Additionally, some works introduce monolithic multimodal solutions [30, 88, 111, 123]. Recently, some work has focused on the general capabilities of MLLM, specifically their performance on new modalities without suffering catastrophic forgetting of the text-only question-answering skills initially mastered by LLM [38, 74, 90]. For example, DeepSeek-VL [86] suggests that supplementing additional text-only training data can mitigate this forgetting. Others [78, 90, 114] try to incorporate interleaved visual-textual data into training to retain language knowledge. However, these methods are limited by training resources and data collection costs. We aim to preserve or even boost performance with text-related training data as little as possible. Some studies [66, 77, 106, 113, 135, 143] also consider expanding the scalability of LLM, such as using Mixture-of-Expert (MoE) with numerous parallel FFNs in the Transformer block alongside a sparse gating network for efficient selection [108, 148, 149]. There are some methods to configure effective information and feedback examples to enhance in-context learning abilities [131, 132]. These methods, however, require a massive increase in training parameters or inference costs. In WINGS, the newly designed parallel learners of Low-Rank Residual Attention (LoRRA) are similar to *MoE on attention block*, but with at least three orders of magnitude less in resource consumption. Compared to some LoRA-related methods [44, 47], WINGS focuses on parallel processing within the attention block rather than in certain in-block linear mappings [89, 121], particularly addressing the issue of capability forgetting in existing architectures.

2.2 Capturing the Attention Shift with MLLM-LAWS

The significant decline in text-only performance is closely linked to the observed related shift during the training process. Research on cross-modal learning [35, 67, 74] shows that transferring to new modalities affects feature distribution, output values, and activation levels. Considering attention



(a) The mixed visual and textual feature inputs. **(b)** Multimodal LLMs' Layer-level Attention Weights (MLLM-LAWS). **(c)** MLLM-LAWS Correlation Illustration

Figure 2: **Illustration of mixed visual-and-textual inputs and the Layer-level Attention Weights (LAWS) with its properties.** (a) The visual feature tokens from the visual encoder and projector are inserted into the textual feature sequence. (b) The attention weight proportion on textual tokens before-image, image-itself, and after-image across layers. The red curve is from the superior text-only MLLM, while the blue curve is from the inferior one. (c) Experiments on over 100 MLLMs show a positive correlation from the $\boldsymbol{\rho}$ for MLLM-LAWS before and after the visual tokens (x-axis) to the text-only performance of the MLLM (y-axis).

weights highlight where MLLM's focus depends on visual or textual tokens for decision-making [98], we investigate how attention shifts among *different parts of the sequences*, mainly where divided by the visual feature tokens. Specifically, we study over 100 diverse MLLMs to uncover how attention is allocated to each part for a text-only better MLLM. We take a closer look at the cross-layer dynamic curve of attention proportion on all text tokens *before* and *after* the inserted image.

For a instruction $\mathbf{x}$ and its hidden states in MLLM as $\mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_s]$ consisting of s mixed visual and textual tokens. Let a_{ij}^l represent the attention weight between the i^{th} and j^{th} tokens in the l^{th} of the L-layers MLLM. We have, for $\forall i$, $\sum_{j=0}^{s} \mathrm{a}_{ij}^l \left(\mathbf{h}^{l-1}\right) = 1$. As shown in Figure 2 (a), since the sequence of flattened visual tokens is continuously interleaved with the textual sequence, we denote the index set of the visual tokens as $\mathcal{V}_{\text{itself}} = \{v_{\text{start}}, v_{\text{start}} + 1, \cdots, v_{\text{end}}\}$. We refer to the textual sequence before the visual tokens as $\mathcal{V}_{\text{before}}$, and similarly, after the visual part as $\mathcal{V}_{\text{after}}$. For an MLLM with L layers, we define the Layer-level Attention Weights (MLLM-LAWS) as:

$$\text{LAWS}_{\mathcal{V}_*} = \left[\mathrm{a}_{\mathcal{V}_*}^1, \mathrm{a}_{\mathcal{V}_*}^2 \cdots, \mathrm{a}_{\mathcal{V}_*}^L\right] \ , \ \ \mathrm{a}_{\mathcal{V}_*}^l = \sum_{i=0}^{s} \sum_{j \in \mathcal{V}_*} \mathrm{a}_{ij}^l \left(\mathbf{h}^{l-1}\right) \ , \tag{1}$$

where token index set $\mathcal{V}_*$ can be $\mathcal{V}_{\text{itself}}$, $\mathcal{V}_{\text{before}}$, or $\mathcal{V}_{\text{after}}$ as mentioned above, and for simplicity, we omit $\mathbf{h}^{l-1}$ in $\mathrm{a}_{\mathcal{V}_*}^l(\cdot)$ of $\text{LAWS}_{\mathcal{V}_*}$. In practice, $\text{LAWS}_{\mathcal{V}_*}$ characterizes the MLLM's attention on the current sequence $\mathcal{V}_{\text{itself}}$, $\mathcal{V}_{\text{before}}$, or $\mathcal{V}_{\text{after}}$ regarding the dynamic curve over all MLLM-layers. As shown in Figure 2 (b), the attention to the textual part initially increases and then decreases as the layers progress, while the trend for the visual one is often the opposite. We find that when the MLLM forgets the text-only instructions, the LAWS of the textual sequence after the visual ones show a deviation from the initial trend of rising and then declining. This implies a shift of layer-level attention in the text following the image $\mathcal{V}_{\text{after}}$ compared to that preceding the image $\mathcal{V}_{\text{before}}$. The dynamics labeled as ③ in Figure 2 (b) show the red curve for better text-only performance towards the worse blue one. To quantify this, we compute the Pearson Correlation Coefficient [92] between LAWS before and after the visual sequence. Formally,

$$\text{Attention Shift} = \mathbb{E}_{\mathbf{x}}\left[-\rho\left(\text{LAWS}_{\mathcal{V}_{\text{before}}}, \text{LAWS}_{\mathcal{V}_{\text{after}}}\right)\right] + 1 \ .$$

Studying the attention shift of over 100 diverse MLLMs, we find a positive correlation between the shift and the text-only performance degradation. In Figure 2 (c), each point represents a trained

MLLM. We find that one reason for the poor text-only performance of MLLMs is the misalignment of textual LAWS before and after the visual sequence, which largely stems from the main branch's attention block lacking sufficient capacity for continued fine-tuning [130, 147]. Next, We focus on how to mitigate the shifted attention weights. Starting with LAWS we give the MLLM "wings".

3 WINGS: Flying to Generality with Low-Rank Residual Attention Learners

In this section, we explore a sufficiently reliable and convenient mechanism to alleviate attention shifts. Specifically, we introduce an additional attention extension structure to assist in learning the main branch's attention. The WINGS architecture operates intuitively by incorporating visual and textual learners designed to mitigate shifts in attention. A dynamic attention-weighted router, guided by negative feedback from biased attention weights, adjusts the outputs of these visual and textual learners. WINGS aims to excel in text-only and visual question-answering tasks with high generality. We start with the typical training pipeline for MLLM like LLaVA [80] (subsection 3.1). Following this, we explore the motivation behind employing parallel modality learners and explain its implementation (subsection 3.2). Finally, we describe the training process for WINGS (subsection 3.3).

3.1 Revisit the Training Pipeline of the MLLM

Following the mainstream architecture of visual-encoder-based MLLM, we take mixed visual and textual features as inputs. For a one-turn conversation, the sequence of the visual feature tokens may appear at any position in the input $\mathbf{x}$. We represent the feature tokens as:

$$\mathbf{x} = [\mathbf{x}_{\mathtt{V}}, \mathbf{x}_{\mathtt{T}}] = \left[\underbrace{\mathbf{h}_1, \cdots,}_{\text{textual features}} \underbrace{\mathbf{h}_{v_{\text{start}}}, \mathbf{h}_{v_{\text{start}}+1}, \cdots, \mathbf{h}_{v_{\text{end}}},}_{\text{visual features}} \underbrace{\cdots, \mathbf{h}_s}_{\text{textual features}} \right] , \tag{2}$$

where we omit the superscript of layer-index l for the 0^{th} layer. The v_{start} and v_{end} represent the starting and ending indices of the visual feature tokens, usually obtained through the vision encoder $\boldsymbol{\psi}$ and projector $\mathbf{W}_{\text{proj}}$, as $\mathbf{x}_{\mathtt{V}} = \mathbf{W}_{\text{proj}} \cdot \boldsymbol{\psi}\left(\mathbf{x}_{\text{image}}\right)$. Correspondingly, $\mathbf{x}_{\mathtt{T}} =$ the remaining 0 to v_{start} and v_{end} to length s denote features of the textual system prompt or user instructions. We consider the posterior of the ground-truth answer as:

$$\Pr\left(\mathbf{x}_{\mathtt{a}} \mid \mathbf{x}\right) = \prod_{i=1}^{s} \mathbb{1}_{[1, v_{\text{start}}) \cup (v_{\text{end}}, s]} \cdot \boldsymbol{\varphi}\left(\mathbf{h}_i \mid [\mathbf{h}_1, \cdots, \mathbf{h}_{i-1}]\right) \ . \tag{3}$$

Here, $\boldsymbol{\varphi}$ represents the main branch LLM, which consists of Transformer decoder layers [119]. Considering the interleaved image tokens, we omit the loss calculation for the "next visual token."

3.2 Visual and Textual Learners Weave WINGS

Motivation: Learning to mitigate the attention shift with modality-specific auxiliary structures. As mentioned in subsection 2.2, MLLM-Laws demonstrates the attention shift in the sequence following the visual features. The shift results from excessive dependency on visual features. This issue may stem from the insufficient alignment within mixed inputs [7, 15, 130], or the main branch struggles to accommodate capacity expansion during continual learning, where new modalities can obscure existing knowledge. It suggests adding a small, adjustable factor to the shifted mixed modality features and regulating unnecessary fluctuations in MLLM-LAWS. Consequently, we aim to adopt an efficient, learnable module as the visual "wing". Compared to the image-text mixed feature inputs of the main branch, it should focus specifically on extracting visual information to share the burden of overly shifted attention. The interaction between the current hidden state and visual features is conducted within this module. Similarly, to balance the auxiliary function of the visual learner, we also construct a symmetrical textual learner. Moreover, it is crucial to appropriately distribute the two learners across both modalities to ensure they function collaboratively.

Structure: parallel learner of attention & token-wise router of attention outputs. To capture key information in shifted modalities while ensuring efficiency, we design a multihead Low-Rank Residual Attention (LoRRA) learner at every layer. It takes input from the hidden state and interacts with the initial visual or text-only feature. The learner facilitates cross-cascading with the initial

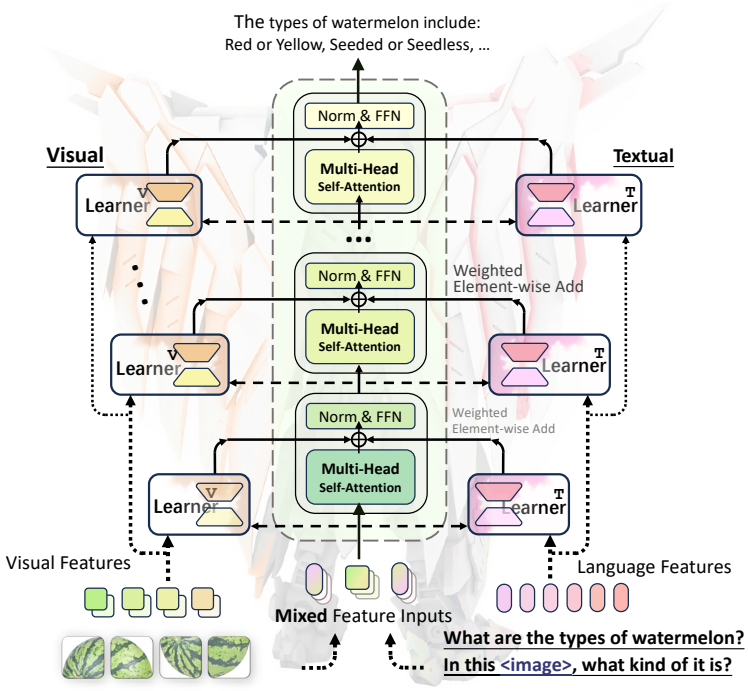


Figure 3: **The WINGS - model architecture.** We introduce extra modules parallel to the main attention, serving as boosted learners to compensate for the attention shift. We train the visual learners on one side, alleviating some shifted attention. Then, we collaboratively learn visual and textual learners based on routing shifted attention weights. They are like light feathers woven "wings".

projected information. Specifically, for the l^{th} layer, the visual/text-only learner is formulated as:

$$\text{Learner}^{*}\left(\text{Q}{=}\mathbf{h}^{l},\text{K},\text{V}{=}\mathbf{x}_{*}\right)_{*\in\{\text{V},\text{T}\}}=\text{Softmax}\left(\frac{\mathbf{h}^{l}\left(\mathbf{1}+\mathbf{W}^{\text{Q}}\right)\cdot\left(\mathbf{x}_{*}\left(\mathbf{1}+\mathbf{W}^{\text{K}}\right)\right)^{\top}}{\sqrt{d_{\text{head}}}}\right)\mathbf{x}_{*}\left(\mathbf{1}+\mathbf{W}^{\text{V}}\right)\mathbf{W}^{\text{O}}\ , \tag{4}$$

where the matrix $\mathbf{W}^{\text{Q}}$, $\mathbf{W}^{\text{K}}$, $\mathbf{W}^{\text{V}}$, and $\mathbf{W}^{\text{O}}$ is low-rank and is obtained by the dot product of $\mathbf{W}_a \in \mathbb{R}^{d\times\underline{d}}$ and $\mathbf{W}_b \in \mathbb{R}^{\underline{d}\times d}$, and $\underline{d}$ is relatively small enough. The symbol $\mathbf{1}$ is represented as the identity matrix. The structure of multihead LoRRA preserves the effectiveness of the cross-attention structure and employs efficient low-rank mapping to reduce computational demands. Following LoRA [47], LoRRA learners also employ random Gaussian initialization for $\mathbf{W}_a$ and sets $\mathbf{W}_b$ to zero. Since $\mathbf{W}^{\text{O}}$ lacks a residual, the output of LoRRA is zero at the beginning of training. As shown in Figure 3, the visual and textual features are fed into their respective side learners, like two "wings" woven together. The outputs of two learners from each layer are then weighted sum to the attention of the main branch. As illustrated in the left part of Figure 4, a router receives attention weights to generate the balance weights of visual and textual learners for each token. In summary, we formulate the WINGS block as:

$$\text{Att}^{\text{WINGS}}=\text{Att}^{\text{main}}+\sum_{*\in\{\text{V},\text{T}\}}\text{Router}\left(\mathbf{a}\right)\cdot\text{Learner}^{*}\left(\mathbf{h}^{l},\mathbf{x}_{*}\right)\ , \tag{5}$$

where $\mathbf{a}\in\mathbb{R}^{s\times s}$ represents the attention weights of the current main branch. The router is formalized as $\text{Router}\left(\mathbf{a}\right)=\mathbf{W}[:,:s]\cdot\mathbf{a}^{\top}$, which is implemented by a single-layer dynamic MLP, $\mathbf{W}\in\mathbb{R}^{2\times s_{\max}}$. It receives the attention weights $\mathbf{a}$ and processes them using $\mathrm{Softmax}$ on two modality learners.

3.3 Stable Training Recipe

The architecture of WINGS comprises four elements: vision encoder, projector, initialized LLM, and the learners with a router. During the training process, the vision encoder is consistently fixed. Firstly,

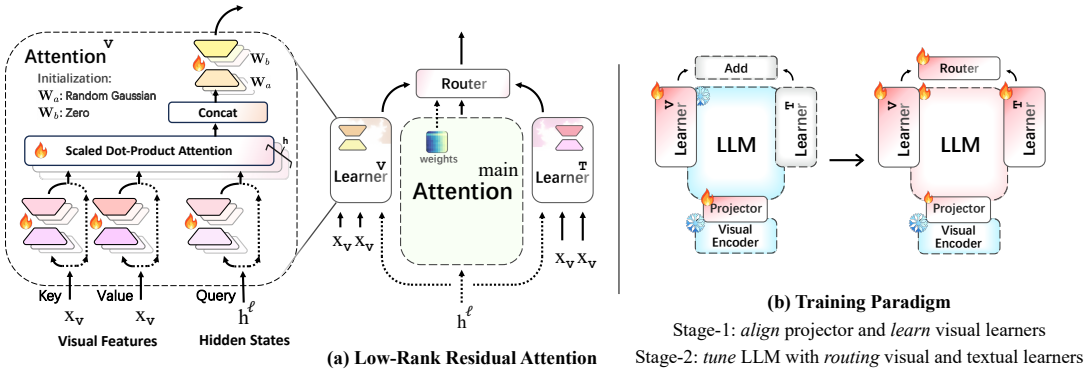


Figure 4: **Illustrations of the detailed WINGS structure, and training strategies.** WINGS is constructed by the Low-Rank Residual Attention (LoRRA) module where the previous hidden state acts as the `query` and the visual/textual features serve as the `key` and `value`. Training starts with visual learners and projectors, followed by the dynamic attention-based routing.

we only fine-tune the projector and visual learners. We primarily employ image-text pairs for visual alignment, while the outputs of visual learners are directly added to the main branch. For this part, the visual learners primarily handle the visual focus, minimizing disturbances to the main branch during continued learning. Subsequently, textual learners are paralleled with visual learners on the attention block of LLMs. The router begins by learning to allocate visual and textual learners from the attention weights of the main branch. At this stage, both types of learners work more effectively together to focus attention on key tokens. To summarize, WINGS prioritizes enhancing visual learners first. Subsequently, it "spreads its wings" by concurrently learning and routing visual and textual learners based on shifted attention weights. During inference, the routed weights of the visual wings branch are deactivated for text-only instructions, while multimodal instructions activate both wings.

4 Experiments

In this section, we first introduce the benchmarks for evaluating WINGS, including Table 1: text-only forgetting on the same multimodal training data, Table 2: comparison with general MLLMs, and Figure 5: analysis on the Interleaved Image-Text (IIT) benchmark with varying levels of vision-related conversation. Following that, we outline the training details and configurations of the WINGS, and delve into experimental analysis across each benchmark. Following that, we perform an ablation study on various learning rates with different training parts. Finally, we provide supplementary descriptions regarding WINGS' overhead compared to general MLLMs and how its innovative compensatory learners help effectively mitigate attention issues.

Evaluation Setups. We aim to assess through MLLM how much visual information is required for evaluation. For example, generic multimodal instructions require MLLMs to strongly capture image aspects, whereas text-only instructions focus on the text. We introduce three types of benchmarks:

- **Standard text-only benchmarks.** We are particularly interested in the text-only performance improvement of WINGS under the same training data and resource conditions. Different datasets including *interdisciplinary exams* like MMLU [43], CMMLU [65], ARC-Easy, ARC-Challenge [27], language *understanding* and *knowledge* such as WinoGrande [103], OpenbookQA [8], Race-Middle, Race-High [60], WSC [129], CHID [144], *reasoning* such as HellaSwag [139], SIQA [105], PIQA [9], OCNLI [48], and *math* and *code*-related tasks such as GSM8K [28] and MBPP [4] are comprehensively evaluated.
- **General multimodal benchmarks.** We evaluate on MMMU [138], MME [37], MMBench [84] (MMB) in English (EN) and Chinese (CN), ScienceQA [87] for test (SciQA), SEED-Bench [64] for image part (SEED), AI2D [57] for test, and HallusionBench [41] (HallB).
- **Our Interleaved Image-Text (IIT) benchmark** with diverse text-only, interleaved, and image-related multi-turn conversations. It includes sampling for MMLU, CMMLU, OpenbookQA, HellaSwag, MMMU, MMBench, SEED-Bench, and AI2D datasets.

Dataset	Model	Vicuna LLM	Vicuna + CLIP	LoRA$_{\text{Vicu.}}$ + CLIP	Vicuna + SigLIP	Qwen LLM	Qwen + CLIP	LoRA$_{\text{Q}}$ + CLIP	Qwen + SigLIP	WINGS (Ours)	Text-only Forgetting (-)	Our Impro. (-)
Exam	MMLU	51.18	51.12	48.89	50.63	**60.86**	50.83	59.67	51.16	60.53	9.70	9.37
	CMMLU	38.60	38.29	37.24	38.73	69.37	62.58	67.87	60.46	**69.82**	8.91	9.36
	ARC-E	57.62	53.63	55.82	53.95	**59.96**	56.93	59.35	55.87	54.29	4.09	-1.58
	ARC-C	33.75	34.60	34.68	35.17	38.90	39.14	38.64	39.50	**43.39**	-0.60	3.89
Under-standing	Winogrande	68.01	64.97	67.83	65.21	**71.38**	69.82	71.03	69.05	69.28	2.33	0.23
	OpenbookQA	77.10	73.28	77.15	72.12	**81.73**	78.31	81.29	77.51	81.05	4.22	3.54
	Race-Middle	63.99	60.10	62.84	59.45	**74.82**	68.25	72.06	68.34	74.24	6.48	5.90
	Race-High	58.74	53.24	54.91	52.69	**71.05**	59.20	65.67	57.72	69.62	13.33	11.90
	WSC	51.30	47.21	51.06	47.72	56.17	54.18	57.30	55.23	**66.35**	0.94	11.12
	CHID	39.05	49.66	45.26	53.49	71.94	71.82	72.92	**74.29**	74.06	-2.35	-0.23
Reasoning	HellaSwag	63.11	63.08	62.58	63.02	**65.70**	61.90	64.32	63.24	65.12	2.46	1.88
	SIQA	42.37	44.06	43.27	44.52	45.57	50.20	46.83	**51.71**	49.64	-6.14	-2.07
	PIQA	71.92	71.95	70.35	71.84	76.59	74.60	73.77	75.19	**78.06**	1.40	2.87
	OCNLI	33.89	37.74	39.41	40.46	49.73	48.31	48.07	50.29	**50.39**	-0.56	0.10
Math	GSM8K	25.19	23.72	22.68	23.05	56.77	50.10	54.25	51.37	**52.08**	5.40	0.71
Code	MBPP	13.80	11.29	13.92	10.80	37.50	34.82	36.72	33.20	**38.92**	4.30	5.72
Multimodal	MMMU-VAL	–	35.67	30.78	35.56	–	34.56	32.33	35.11	**39.89**	–	4.78
	MMMU-TEST	–	34.40	30.90	35.33	–	34.90	31.80	35.10	**37.30**	–	2.20
	MMBench	–	63.18	59.83	65.14	–	66.05	62.84	**70.94**	70.53	–	-0.41
	ScienceQA	–	67.72	64.49	71.50	–	74.26	69.09	74.89	**78.76**	–	3.87

Table 1: **Performance comparisons of WINGS and the baseline MLLMs under the same training data**. We consider 8 baseline MLLMs, including LLMs as Vicuna$_{\text{v1.5}}$ & Qwen1.5, visual encoders as CLIP [99] & SigLIP [140], and training strategies as full-parameter & LoRA fine-tuning. The first entry represents the initial LLM, upon which each MLLM is trained. Our evaluation spans 6 domains with 20 datasets. WINGS is based on the Qwen1.5 and SigLIP, and the column "Our Improvement" highlights how much WINGS surpasses its baseline with the same backbones.

Model Summaries & Implementation Details. We release the WINGS$_{\texttt{base}}$ and WINGS$_{\texttt{pro}}$, with Qwen1.5-7B LLM [6] and SigLIP [140] visual encoder as the foundations. We also introduce the WINGS$_{\text{1.8B}}$ version, adapted to Qwen1.5-1.8B LLM for edge device compatibility. As illustrated in Figure 4, we only optimize the projector and the image learners of WINGS for the first alignment stage. The LLM branch adaptation is incorporated during the second instruction tuning stage. We train for 1 epoch with the AdamW optimizer and the Cosine learning schedule. Typically, the learning rates for the first and second stages are set at $1e^{-3}$ and $2e^{-6}$ (with the projector part as $1e^{-5}$), respectively. For WINGS$_{\texttt{base}}$, approximately 1m training data to align image learners and about 0.6m supervised fine-tuning instructions for the next stage (the same as LLaVA$_{\text{v1.5}}$ [80]). In the WINGS$_{\texttt{pro}}$, we use the same aligned data and approximately 2m training data for learning image-text learners. These two types of MLLM require about 1.5 and 6 days of training on 8× A100 GPUs, respectively. The training datasets for WINGS$_{\texttt{mini}}$ are consistent with the WINGS$_{\texttt{pro}}$. It takes approximately 5 days to run on 4× A100 GPUs.

Details in Figure 2. We adopt various multimodal to text-only sample ratios (`25:1`, `20:1`, `10:1`, `5:1`, `2:1`, `1:1`, `1:2`, ..., `1:25`) plus an `all:0` setup (12 combinations total) to ensure sufficient scenarios. The learning rate is kept consistent with the setup described above. We sample 5 models per epoch, excluding 12 failed ones due to issues like gradient explosion, resulting in 108 for analysis.

4.1 Toward Comprehensive Text-only and Multimodal Performance

Text-only Comparison in Fair Data and Resource Environments. As shown in Table 1, "Vicuna-v1.5 + CLIP" corresponds to LLaVA$_{\text{v1.5}}$, and "Qwen1.5 + SigLIP" serves as the foundation for WINGS. When comparing LLM itself and the rest of MLLMs, we observe that fine-tuning with multimodal instructions, compared to the "Qwen LLM", there is text-only forgetting in 12 out of 16 datasets, with notable decreases of up to 9.70, 8.91, and 13.33 in MMLU, CMMLU, and RACE-High, respectively. WINGS significantly improve performance on datasets such as MMLU, CMMLU, RACE-High, and WSC, despite the potential for severe text-only forgetting on baselines. Additionally, we find that

Method \ Dataset	Text-Only QAs							Multimodal QAs						
	MMLU/C*	RACE-M/H	ARC	HellaSwag	Winog.	GSM8K	MBPP	MMMU-V/T	MMB-EN/CN	MME	SciQA	SEED	AI2D	HallB
Equal-Scale Open-Source 7B Multimodal LLMs														
O-Flamingo$_{v2}$ [5]	26.9 / 27.1	40.3 / 32.6	31.0	55.4	58.3	10.2	9.1	29.1 / 28.7	10.9 / 13.3	803.9	55.8	30.2	32.6	30.4
IDEFICS [51]	33.0 / 26.4	38.2 / 36.9	33.2	58.9	60.2	11.7	8.1	17.6 / 20.2	49.6 / 27.3	1239.3	62.4	44.8	43.4	24.6
InstructBLIP [29]	43.2 / 35.7	52.8 / 49.7	39.5	55.7	54.9	18.3	10.3	32.7 / 32.1	38.5 / 26.8	1425.6	61.3	45.7	41.1	33.3
ShareGPT4V [14]	47.6 / 36.9	55.9 / 51.0	41.6	54.7	60.1	18.0	8.9	35.5 / 35.2	67.4 / 63.1	**1915.3**	68.9	68.1	58.2	26.6
Qwen-VL [7]	49.7 / 58.3	65.2 / 64.8	34.4	58.2	61.0	49.0	34.6	36.4 / 35.9	60.3 / 57.4	1806.2	69.6	62.0	61.9	34.1
Monkey [73]	52.8 / 66.9	65.6 / 62.1	38.2	60.6	59.3	51.8	37.1	**40.3** / 37.1	71.9 / 67.8	1815.4	78.3	69.1	62.5	42.1
LLaVA$_{v1.5}$ [80]	51.1 / 38.3	60.1 / 53.2	34.6	63.1	65.0	23.7	11.3	35.7 / 34.4	63.2 / 57.7	1518.6	67.7	63.7	56.4	29.7
LLaVA$_{Next}$ [81]	50.2 / 39.7	65.1 / 58.3	36.0	63.7	68.9	30.3	23.0	37.6 / 35.8	67.8 / 61.8	1760.3	70.1	69.1	66.4	29.6
DeepSeek-VL [86]	53.9 / 64.0	70.6 / 63.8	39.2	65.1	67.2	55.3	**43.1**	37.6 / 35.3	72.7 / **72.5**	1716.8	80.6	70.0	**66.5**	36.2
WINGS (Ours)	60.5 / **69.8**	74.2 / 69.6	43.4	65.1	69.3	52.1	38.9	39.9 / **37.3**	70.5 / 68.3	1753.8	78.8	69.5	62.7	45.8
WINGS$_{pro}$ (Ours)	**61.3** / 68.5	**82.8 / 76.3**	**46.3**	**69.2**	**70.9**	**56.3**	39.3	38.2 / 36.9	**73.1** / 69.0	1786.1	**83.1**	**70.2**	65.8	**47.3**
Advanced Private Multimodal LLMs														
GPT-4 [95]	83.5 / 71.2	93.2 / 87.8	93.6	88.4	75.6	91.6	56.2†	–	–	–	–	–	–	–
GPT-4V [94]	79.3 / 69.4	93.7 / 89.2	92.9	84.7	76.1	88.4	72.4	58.9 / 56.8	77.0 / 74.4	2153.6	68.4	73.7	75.5	46.5
Gemini$_{pro\ vision}$ [102]	85.9 / 73.7	88.9 / 83.2	85.0	78.8	71.5	86.4	61.5	60.6 / 62.2	73.6 / 74.3	2193.2	58.3	70.8	70.2	45.2
Efficient Multimodal LLMs with WINGS$_{1.8B}$														
DeepSeek-VL$_{1.3B}$ [86]	31.7 / 38.2	63.6 / 58.4	35.8	**52.9**	45.7	17.6	16.3	33.8 / 32.3	65.1 / 60.7	1483.4	65.4	63.3	50.1	25.0
MiniCPM-V$_{2.4B}$ [49]	42.4 / 40.9	**68.8** / 62.6	37.0	48.3	51.7	32.5	24.2	**37.2 / 34.4**	**65.7 / 64.1**	**1584.1**	64.9	**64.7**	54.9	**31.8**
WINGS$_{1.8B}$ (Ours)	**44.9 / 50.9**	68.5 / **63.2**	**37.1**	50.5	**53.0**	**40.6**	**28.5**	35.7 / 33.9	64.2 / 61.2	1527.3	**67.5**	62.8	**55.2**	30.2

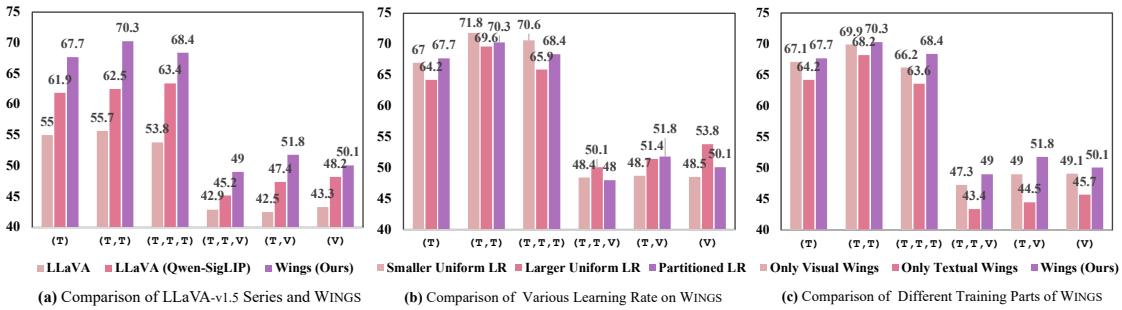
Table 2: **Performance comparisons of the equal-scale MLLMs and the efficient multimodal LLMs** on text-only and multimodal datasets. We evaluate the open-source, efficient, and private API MLLMs. We select 18 representative evaluation datasets. C* represents the CMMLU dataset.

the forgetting effects of CLIP and SigLIP are similar. In contrast, parameter-efficient fine-tuning methods like LoRA result in less text-only forgetting but underperform on multimodal questions. Overall, WINGS' visual and textual learners are credibly demonstrated to retain performance on text-only tasks while also performing well on visual-related questions. In datasets like CHID, OCNLI, and SIQA, MLLMs show improved text-only performance due to increased language diversity (*e.g.*, Chinese context) or semantic similarity in their fine-tuning data.

General Evaluation in Text-Only and Multimodal Tasks. We present the performance of 9, roughly 8B open-source MLLMs, 2 roughly 2B, and 2 private API ones evaluated in the general text-only and multimodal tasks. Table 2 shows that WINGS series can perform better on text-only and multimodal question-answering datasets. It achieves state-of-the-art performance on 13 out of 18 datasets, significantly surpassing LLaVA$_{v1.5}$ with the same architecture. We find that WINGS is equally effective for more efficient foundations, as shown in the "Efficient Multimodal LLMs" parts. WINGS can still capture key elements and demonstrate good scalability as the parameter increases. Although WINGS$_{base}$ does not receive additional training for the text-only component, it is still able to achieve comparable performance.

4.2 Interleaved Image-Text (IIT) Benchmark

To finely evaluate MLLMs, we construct a series of text-only and multimodal mixed multi-turn conversations. We extract instructions from MMLU, CMMLU, OpenbookQA, HellaSwag, MMMU, MMBench, SEED-Bench, and AI2D datasets with similar semantics by chroma [39]. We then polish the connection between some instructions using GPT-3.5 Turbo to make them closer to real-world conversations. We set up 6 vision-content configurations, categorized by the multi-turn content as: (T), (T, T), (T, T, T), (T, T, V), (T, V), and (V). For instance, (T, T, V) indicates two consecutive text-only queries followed by a visual question requiring a response.



(a) Comparison of LLaVA-v1.5 Series and WINGS **(b)** Comparison of Various Learning Rate on WINGS **(c)** Comparison of Different Training Parts of WINGS

Figure 5: **Performance comparison** on the newly constructed **Interleaved Image and Text (IIT) Benchmark** of the **LLaVA series**, **different learning rate** and **fine-tuning parts**. The horizontal axis represents different multimodal question settings. The horizontal axis shows different multimodal setups, *e.g.*, (T, T, I) represents a visual question after two text-only QAs. The three subfigures represent different ablation settings, with the violet color representing our WINGS.

4.3 Ablation Studies

Referencing Figure 5, we address three questions to comprehensively analyse WINGS:

- Can WINGS sustain performance with interleaved evaluation? We find that part (a) highlights WINGS surpassing $\text{LLaVA}_{\text{v1.5}}$ and the same-backbone as $\text{LLaVA}_{\text{v1.5}}$ (Qwen-SigLIP) for each multi-turn setting, especially in text-centric dialogues.
- How do WINGS fare with different learning rate settings? Part (b) demonstrates that using a lower learning rate maintains proficiency in text-only tasks but falls short in multimodal questions, while a higher rate boosts multimodal abilities but not text-only. Applying a higher learning rate to the projector and a lower one to the others achieves the optimal.
- Are all components of WINGS equally effective? In part (c), we examine that incorporating visual learners alone slightly preserves text-only abilities, likely by minimizing disruption to the LLM, but diminishes performance on multimodal tasks.

In the diverse IIT bench, which ranges from text-rich to multimodal contexts, the effectiveness of WINGS is particularly evident. As shown in Figure 1, within real-world applications, textual content offers insights for following visual tasks. WINGS excels in handling text-only tasks while improving performance on visual-related instructions.

5 Conclusion

We propose WINGS, which includes visual and textual learners, to alleviate text-only forgetting. The learner is composed of efficient Low-Rank Residual Attention (LoRRA). We start by considering the shifted attention weights in MLLM and, in the first stage, focus on learning the visual learner. Then, we co-train the visual and textual learners with routing based on the shifted attention weights. WINGS demonstrates remarkable performance on text-only, visual-question-answering, and newly constructed Interleaved Image-Text (IIT) benchmarks. WINGS allows for maintaining text-only performance with limited resources and further enhances performance in well-resourced settings.

Acknowledgments and Disclosure of Funding

This research was supported by National Science and Technology Major Project (2022ZD0114805), NSFC (61773198, 62376118,61921006), Collaborative Innovation Center of Novel Software Technology and Industrialization, CCF-Tencent Rhino-Bird Open Research Fund (RAGR20240101)

References

[1] Keivan Alizadeh, Iman Mirzadeh, Dmitry Belenko, Karen Khatamifard, Minsik Cho, Carlo C. Del Mundo, Mohammad Rastegari, and Mehrdad Farajtabar. LLM in a flash: Efficient large language model inference with limited memory. *CoRR*, abs/2312.11514, 2023.

[2] Reza Yazdani Aminabadi, Samyam Rajbhandari, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Olatunji Ruwase, Shaden Smith, Minjia Zhang, Jeff Rasley, and Yuxiong He. Deepspeed- inference: Enabling efficient inference of transformer models at unprecedented scale. In *SC22*, pages 46:1–46:15, 2022.

[3] Anthropic. Introducing Claude, 2023. URL `https://www.anthropic.com/index/introducing-claude`.

[4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[5] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Yitzhak Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *CoRR*, abs/2308.01390, 2023.

[6] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[7] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[8] Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. Careful selection of knowledge to solve open book question answering. In *ACL*, pages 6120–6129. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1615. URL `https://doi.org/10.18653/v1/p19-1615`.

[9] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *AAAI*, pages 7432–7439, 2020.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

[11] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. *arXiv preprint arXiv:2312.06742*, 2023.

[12] Yingshan Chang, Guihong Cao, Mridu Narang, Jianfeng Gao, Hisami Suzuki, and Yonatan Bisk. Webqa: Multihop and multimodal QA. In *CVPR*, pages 16474–16483. IEEE, 2022.

[13] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024.

[14] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023.

[15] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[16] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015.

[17] Yunkai Chen, Qimeng Wang, Shiwei Wu, Yan Gao, Tong Xu, and Yao Hu. Tomgpt: Reliable text-only training approach for cost-effective multi-modal large language model. *ACM Trans. Knowl. Discov. Data*, 2024.

[18] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[19] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

[20] Po-Han Chi, Pei-Hung Chung, Tsung-Han Wu, Chun-Cheng Hsieh, Yen-Hao Chen, Shang-Wen Li, and Hung-yi Lee. Audio albert: A lite bert for self-supervised learning of audio representation. In *IEEE SLT*, pages 344–350, 2021.

[21] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

[22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023.

[23] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, pages 4299–4307, 2017.

[24] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023.

[25] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024.

[26] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022.

[27] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018.

[28] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[29] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *NeurIPS*, 36, 2024.

[30] Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models. *CoRR*, abs/2406.11832, 2024.

[31] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[32] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey for in-context learning. *CoRR*, abs/2301.00234, 2023.

[33] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024.

[34] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *ACL*, pages 320–335, 2022.

[35] Haoyi Duan, Yan Xia, Mingze Zhou, Li Tang, Jieming Zhu, and Zhou Zhao. Cross-modal prompts: Adapting large pre-trained models for audio-visual downstream tasks. In *NeurIPS*, 2023.

[36] Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. Mixture-of-loras: An efficient multitask tuning for large language models. *CoRR*, abs/2403.03432, 2024.

[37] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[38] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. In *NeurIPS*, 2022.

[39] Chroma Group. Chroma - the open-source embedding database. https://github.com/chroma-core/chroma, 2017.

[40] Poe Group. Poe - fast, helpful ai chat. https://poe.com, 2024.

[41] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: An advanced diagnostic suite for entangled language hallucination & visual illusion in large vision-language models. *arXiv preprint arXiv:2310.14566*, 2023.

[42] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *CVPR*, 2024.

[43] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[44] Zhang Hengmin, Yang Jian, Du Wenli, Zhang Bob, Zha Zhiyuan, and Wen Bihan. Enhanced acceleration for generalized nonconvex low-rank matrix learning. *Chinese Journal of Electronics*, 34(1):1–16, 2025. doi: 10.23919/cje.2023.00.340.

[45] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

[46] Anwen Hu, Yaya Shi, Haiyang Xu, Jiabo Ye, Qinghao Ye, Ming Yan, Chenliang Li, Qi Qian, Ji Zhang, and Fei Huang. mplug-paperowl: Scientific diagram analysis with the multimodal large language model. *arXiv preprint arXiv:2311.18248*, 2023.

[47] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[48] Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S. Moss. OCNLI: original chinese natural language inference. In *EMNLP*, volume EMNLP 2020 of *Findings of ACL*, pages 3512–3526, 2020.

[49] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[50] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *ACL*, 2023.

[51] IDEFICS. Introducing idefics: An open reproduction of state-of-the-art visual language model. `https://huggingface.co/blog/idefics`, 2023.

[52] Jitesh Jain, Jianwei Yang, and Humphrey Shi. Vcoder: Versatile vision encoders for multimodal large language models. *CoRR*, abs/2312.14233, 2023.

[53] Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases. *arXiv preprint arXiv:2303.14742*, 2023.

[54] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[55] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[56] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020.

[57] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, pages 235–251, 2016.

[58] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

[59] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *ECCV*, 2022.

[60] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In *EMNLP*, pages 785–794, 2017.

[61] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023.

[62] Bo Li, Peiyuan Zhang, Jingkang Yang, Yuanhan Zhang, Fanyi Pu, and Ziwei Liu. Otterhd: A high-resolution multi-modality model. *arXiv preprint arXiv:2311.04219*, 2023.

[63] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023.

[64] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[65] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. *arXiv preprint arXiv:2306.09212*, 2023.

[66] Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. *arXiv:*, 2024.

[67] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, pages 9694–9705, 2021.

[68] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pages 12888–12900, 2022.

[69] Junnan Li, Silvio Savarese, and Steven C. H. Hoi. Masked unsupervised self-training for zero-shot image classification. *CoRR*, abs/2206.02967, 2022.

[70] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *CoRR*, abs/2403.00231, 2024.

[71] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL/IJCNLP*, pages 4582–4597, 2021.

[72] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.

[73] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023.

[74] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y. Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *NeurIPS*, 2022.

[75] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[76] Bin Lin, Tao Peng, Chen Zhang, Minmin Sun, Lanbo Li, Hanyu Zhao, Wencong Xiao, Qi Xu, Xiafei Qiu, Shen Li, Zhigang Ji, Yong Li, and Wei Lin. Infinite-llm: Efficient LLM service for long context with distattention and distributed kvcache. *CoRR*, abs/2401.02669, 2024.

[77] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024.

[78] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. VILA: on pre-training for visual language models. *CoRR*, abs/2312.07533, 2023.

[79] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

[80] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36, 2023.

[81] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

[82] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang, Jianfeng Gao, and Chunyuan Li. Llava-plus: Learning to use tools for creating multimodal agents. *arXiv:2311.05437*, 2023.

[83] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[84] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

[85] Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*, 2024.

[86] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

[87] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *NeurIPS*, 35:2507–2521, 2022.

[88] Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model. *CoRR*, abs/2405.20797, 2024.

[89] Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jifeng Dai, Yu Qiao, and Xizhou Zhu. Mono-internvl: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training, 2024.

[90] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[91] Shervin Minaee, Tomás Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *CoRR*, abs/2402.06196, 2024.

[92] Cuong V Nguyen, Tal Hassner, Cedric Archambeau, and Matthias Seeger. Leep: A new measure to evaluate transferability of learned representations. In *ICML*, 2020.

[93] OpenAI. Chatgpt. `https://openai.com/blog/chatgpt`, 2022.

[94] OpenAI. Gpt-4v(ision) system card. `https://cdn.openai.com/papers/GPTV_System_Card.pdf`, 2023.

[95] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[96] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[97] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[98] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, pages 3942–3951, 2018.

[99] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[100] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The J. Mach. Learn. Res.*, 21(1):5485–5551, 2020.

[101] Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. Zero-infinity: breaking the GPU memory wall for extreme scale deep learning. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2021, St. Louis, Missouri, USA, November 14-19, 2021*, page 59. ACM, 2021.

[102] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[103] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.

[104] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. In *ICLR*, 2022.

[105] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. SocialIQA: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728, 2019.

[106] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017.

[107] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[108] Zhi-Hao Tan, Jian-Dong Liu, Xiao-Dong Bi, Peng Tan, Qin-Cheng Zheng, Hai-Tian Liu, Yi Xie, Xiao-Chuan Zou, Yang Yu, and Zhi-Hua Zhou. Beimingwu: A learnware dock system. In *KDD*, 2024.

[109] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[110] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

[111] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *CoRR*, abs/2405.09818, 2024.

[112] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[113] The Mosaic Research Team. Introducing dbrx: A new state-of-the-art open llm, March 2024. URL `https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm`.

[114] Fu Tianhao, Yang Zehua, Ye Zhisheng, Ma Chenxiang, Han Yang, Luo Yingwei, Wang Xiaolin, and Wang Zhenlin. A survey on the scheduling of dl and llm training jobs in gpu clusters. *Chinese Journal of Electronics*, 34:1–25, 2024. doi: 10.23919/cje.2024.00.070.

[115] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[116] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[117] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34: 200–212, 2021.

[118] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*, 2022.

[119] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[121] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models. *CoRR*, abs/2311.03079, 2023.

[122] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[123] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need. *CoRR*, abs/2409.18869, 2024.

[124] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Vary: Scaling up the vision vocabulary for large vision-language models. *arXiv preprint arXiv:2312.06109*, 2023.

[125] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022.

[126] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022.

[127] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

[128] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023.

[129] Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A chinese language understanding evaluation benchmark. In *COLING*, pages 4762–4772, 2020.

[130] Shipeng Yan, Jiangwei Xie, and Xuming He. DER: dynamically expandable representation for class incremental learning. In *CVPR*, pages 3014–3023, 2021.

[131] Xu Yang, Yongliang Wu, Mingzhuo Yang, Haokun Chen, and Xin Geng. Exploring diverse in-context configurations for image captioning. In *NeurIPS*, 2023.

[132] Xu Yang, Yingzhe Peng, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. Lever lm: Configuring in-context sequence to lever large vision language models, 2024.

[133] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

[134] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

[135] Chao Yi, De-Chuan Zhan, and Han-Jia Ye. Bridge the modality and capacity gaps in vision-language model selection. *CoRR*, abs/2403.13797, 2024.

[136] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

[137] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[138] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

[139] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *ACL*, pages 4791–4800, 2019.

[140] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023.

[141] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*, 2024.

[142] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv:2205.01068*, 2022.

[143] Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model spider: Learning to rank pre-trained models efficiently. In *NeurIPS*, 2023.

[144] Chujie Zheng, Minlie Huang, and Aixin Sun. Chid: A large-scale chinese idiom dataset for cloze test. In *ACL*, pages 778–787, 2019.

[145] WANG Zhifang, ZHEN Jiaqi, LI Yanchao, LI Guoqiang, and HAN Qi. Multi-feature multimodal biometric recognition based on quaternion locality preserving projection. *Chinese Journal of Electronics*, 28(4):789–796, 2019. doi: 10.1049/cje.2019.05.006.

[146] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: less is more for alignment. In *NeurIPS*, 2023.

[147] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *CVPR*, pages 9036–9046. IEEE, 2022.

[148] Zhi-Hua Zhou. Learnware: on the future of machine learning. *Frontiers Computer Science*, 10(4), 2016.

[149] Zhi-Hua Zhou and Zhi-Hao Tan. Learnware: Small models do big. *CoRR*, abs/2210.03647, 2022.

[150] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, Hongfa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Wancai Zhang, Zhifeng Li, Wei Liu, and Li Yuan. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *CoRR*, abs/2310.01852, 2023.

[151] Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal C4: an open, billion-scale corpus of images interleaved with text. In *NeurIPS*, 2023.

[152] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593, 2019.

Supplementary Material

A Experimental Setups and Implementation Details

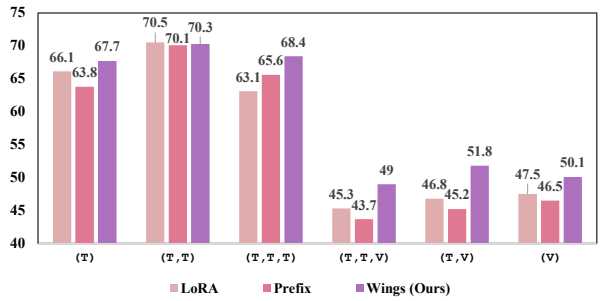
Training Datasets. The training datasets for the first and second stage of WINGS$_{\texttt{base}}$ are consistent with LLaVA$_{\text{v1.5}}$ [80]. For the second stage, WINGS$_{\texttt{pro}}$ extends the training dataset to include some visual QA datasets as ALLaVA [13], SynthDog [59], and ArXivQA [70], and text-only QA datasets as Stanford Alpaca [109], Alpaca GPT-4 [97], LIMA [146], UltraChat [31], WebQA [12], and BELLE-0.5M [53]. WINGS$_{\text{1.8B}}$ shares the same training set as WINGS$_{\texttt{pro}}$.

Model Structures. We employ Qwen1.5 [6] and SigLIP [140] as our foundations.

Training Hyperparameters. We utilize a batch size of 32, along with the AdamW optimizer and a cosine schedule. For all WINGS-series, the learning rate is set at $1e^{-3}$ for the first stage and adjusts to $2e^{-6}$ for the second stage, except for the projector as $1e^{-5}$.

Training Environment. WINGS$_{\texttt{base}}$ and WINGS$_{\texttt{pro}}$ are trained over approximately 1.5 or 6 days on $8\times$ A100 GPUs. WINGS$_{\text{1.8B}}$ require approximately 5 days of training on $4\times$ A100 GPUs.

B Additional Experimental Results



(a) Comparison of Parameter Efficient Modules and WINGS

Figure 6: **Performance comparison** on the newly constructed **Interleaved Image and Text (IIT) Benchmark** of the **Parameter Efficient Modules**. The horizontal axis represents different multimodal question settings. The horizontal axis shows different multimodal setups, *e.g.*, (`T`, `T`, `I`) represents a visual question after two text-only QAs.

Should we only add additional modules on top of an LLM branch or, like WINGS, create two distinct learners for visual and textual modalities? We delve into the low-rank adaptation (LoRA) [47] and Prefix-tuning [71] for minimally adapt to the LLM component. These techniques introduce optimization parameters beyond the primary branch. These lightweight adjustments align with extensive modifications, effectively minimizing text-only forgetting but concurrently curbing cross-modal positive transfer.

C Discussion

WINGS is a universal plugin that can be integrated with any multimodal mixed-input MLLMs. Notably, it introduces a new concept of competitive reuse among multiple expert groups: we may not require the experts to the Transformer block's MLP layer at a scale three orders of magnitude larger; instead, a minor update in the attention for better allocation may suffice. This idea is also found in some variants of LoRA [36, 118]. In the future, we will gradually explore the future of MLLMs.

Regarding the extent to which WINGS alleviates attention shifts, we acknowledge that the main branch of WINGS (without visual and textual learners) still exhibits attention shifts. However, since the outputs from the visual and textual learners compensate within the hidden state, we extract portions of the attention weight matrices from both learners and add them to the main branch's weights. Results

show that throughout the training process, the attention shift phenomenon is continually mitigated under the influence of these learners. Essentially, the primary branch's attention ideally retains its original text-only capabilities, while the visual solutions are implemented within the auxiliary learner.

D Limitation & Broader Impact

Despite WINGS' strong adaptability for embedding auxiliary attention learners in various MLLMs, integrating visual learners requires restarting the feature alignment training, incurring extra costs. Additionally, its deployment on edge devices faces limitations, with $\text{WINGS}_{1.8B}$ offering a solution at the expense of performance. Furthermore, WINGS still requires some text-only data to replay and enhance overall performance, aiming for integration into more generic AI systems in the future.

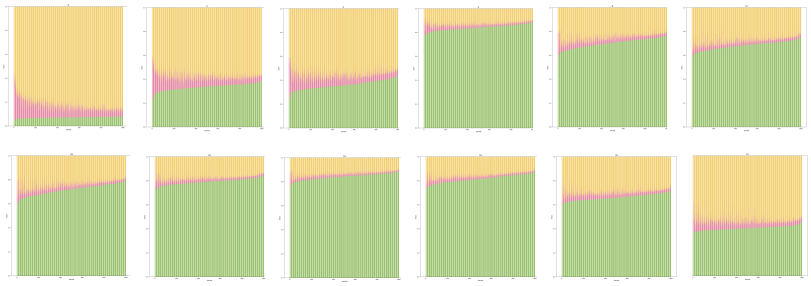


Figure 7: **Dynamics of Attention Weights from Shallow to Deep Layers.** We calculate the proportion of attention weights for the image-before (yellow), the image-itself (red), and the image-after (green) in each layer. From left to right, top to bottom, from shallow to deep layers.

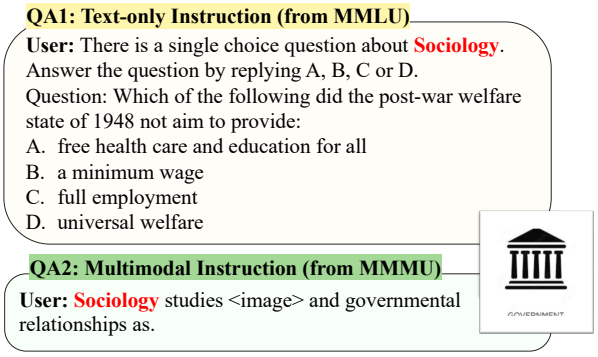


Figure 8: **An Example of an Interleaved Image-Text Benchmark.** This dialogue is represented as (T, V), consisting of a text-only QA from MMLU [43] and a visual QA from MMMU [138]. It can be observed that, due to the sampling, both include questions from the *Sociology* category.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract reflects the contribution and scope of our work and illustrates our model and experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Appendix D.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We show the architecture in Figure 3 and Figure 4. Additionally, we provide the experimental details in section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The datasets and benchmarks used in our work are open source. The code of our proposed method will be released upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide training and test details in section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: During the training process and evaluation, we use methods such as fixed random seeds to obtain certain and consistent results, so our work do not focus on the significance of the experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources for the experiments is reported in section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and strictly followed the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts in the Appendix D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no such risk in this work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have listed and cited relevant datasets and models in detail in the paper to ensure that the license and terms are met.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: Our work uses public datasets for training and evaluation, and does not involve or open source new datasets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.