AdaDARE-γ: Balancing Stability and Plasticity in Multi-modal LLMs through Efficient Adaptation

Jingyi Xie[1*], Jintao Yang[1*], Zhunchen Luo[1], Yunbo Cao[1], Qiang Gao[1]
Mengyuan Zhang[2†], Wenpeng Hu[1†]
[1]Information Research Center of Military Science, AMS
[2]School of Cyber Science and Technology, Beihang University
{jingyi_xie96, jintaoyang2023}@163.com, huwenp@outlook.com, myzhang18@buaa.edu.cn

Abstract

Adapting Multi-modal Large Language Models (MLLMs) to target tasks often suffers from catastrophic forgetting, where acquiring new task-specific knowledge compromises performance on pre-trained tasks. In this paper, we introduce AdaDARE-γ, an efficient approach that alleviates catastrophic forgetting by controllably injecting new task-specific knowledge through adaptive parameter selection from fine-tuned models without requiring retraining procedures. This approach consists two key innovations: (1) an adaptive parameter selection mechanism that identifies and retains the most task-relevant parameters from fine-tuned models, and (2) a controlled task-specific information injection strategy that precisely balances the preservation of pre-trained knowledge with the acquisition of new capabilities. Theoretical analysis proves the optimality of our parameter selection strategy and establishes bounds for the task-specific information injection factor. Extensive experiments on InstructBLIP and LLaVA-1.5 across image captioning and visual question answering tasks demonstrate that AdaDARE-γ establishes new state-of-the-art results in balancing model performance. Specifically, it maintains 98.2% of pre-training effectiveness on original tasks while achieving 98.7% of standard fine-tuning performance on target tasks.

1. Introduction

With the rapid development of artificial intelligence [7, 33, 41], multi-modal large language models (MLLMs) have gained significant attention due to their ability to capture rich representations from various data modalities [1, 5, 10, 28, 40, 45, 46]. Pioneering works such as GPT-4V [1], Qwen2-VL [45], InstructBLIP [10] and LLaVA-1.5 [28] have played pivotal roles in shaping the landscape of general-purpose AI assistants, significantly enhancing their capabilities and versatility.

The typical architecture of MLLMs consists of two key components: modality-specific sub-modules and a lightweight connector [46]. The modality-specific sub-modules, such as vision encoders and language models, are designed to process and extract information from different types of input data. The lightweight connector plays a crucial role in aligning these diverse representations, enabling the model to comprehend multi-modal inputs and thus generate coherent outputs [5, 27, 28]. This structure allows MLLMs to effectively combine information from text, images, and potentially other modalities, leading to more sophisticated and versatile AI systems.

Despite their advanced capabilities, MLLMs often struggle with performance on unseen tasks [17, 31, 49] due to their unique architectural complexity and the inherent disparities in data characteristics across modalities. Consequently, continual fine-tuning [36] remains a crucial approach for rapidly adapting these models to specific downstream tasks and enhancing their performance. Nevertheless, this process frequently leads to catastrophic forgetting, a phenomenon where the model's performance on pre-trained tasks deteriorates as it acquires new, task-specific knowledge, as shown in Fig. 1. This highlights a crucial challenge: how to efficiently strike a balance between enhancing MLLMs' performance on target tasks (plasticity) while preserving their effectiveness on original tasks (stability)?

Current research on mitigating catastrophic forgetting predominantly focuses on traditional continual learning settings with smaller-scale or uni-modal models [2, 34, 42, 44], which exhibit inherent limitations in both representation capacity and cross-modal understanding. In context of MLLMs, Model-Tailor [61] made the first attempt to preserve pre-trained task capabilities during fine-tuning on im-

*Equal contribution.
†Corresponding authors.

age captioning and visual question answering tasks. However, this approach, as illustrated in Tab. 1 and Tab. 2, achieves only sub-optimal performance in balancing pre-trained and target task capabilities. Furthermore, it incurs substantial computational overhead due to the requirement of layer-wise Hessian inverse calculations.

In response to these challenges, we introduce Adaptive Drop And REscale with a γ factor (AdaDARE-γ): an efficient approach that alleviates catastrophic forgetting by controllably injecting new task-specific knowledge through adaptive parameter selection from fine-tuned models without requiring retraining procedures. Building upon the DARE framework, AdaDARE-γ employs a probabilistic dropout mechanism for each delta parameter. This adaptive process selectively retains the most task-salient parameters. Furthermore, inspired by the Task-Arithmetic paradigm [20], which demonstrates the efficacy of simple arithmetic operations on delta parameters for model editing, we incorporate a γ factor to characterize the intensity of task-specific information injection. This approach enables fine-grained control over the balance between preserving pre-trained knowledge and adapting to new tasks, effectively creating a dynamic equilibrium between generalization and specialization in MLLMs.

In summary, our work makes several significant contributions:

- We propose AdaDARE-γ, an efficient framework that addresses catastrophic forgetting in MLLMs by combining principled adaptive parameter selection with controlled knowledge injection, enabling flexible trade-off between preserving pre-trained capabilities and acquiring task-specific adaptations.
- We provide theoretical analysis that establishes optimality guarantees for parameter selection and derives bounds for the task-specific information injection factor, offering theoretical insights into the trade-off between stability and plasticity.
- We demonstrate through comprehensive experiments that our method establishes new state-of-the-art performance in balancing the target task performance and pre-trained knowledge retention, with maintaining $\sim$98.2% effectiveness on original tasks versus pre-training and achieving $\sim$98.7% performance on the target task compared to standard fine-tuning, providing a more effective solution to the stability-plasticity dilemma in MLLMs.

2. Related Work

2.1. Multi-modal Large Language Models

Multi-modal Large Language Models formally refers to the LLM-based model with the ability to receive, reason and generate with multi-modal information [46]. Ever since the release of GPT-4 [1], MLLMs have gained much attention because of the amazing performance it shows. Recent advancements [4, 5, 24, 27, 45, 60] have emerged with stronger ability of understanding and reasoning, benefiting from the rapid development of LLMs [7, 9, 41]. Drawing inspiration of the success of instruction tuning in LLMs [33], researchers have extended this crucial technique to MLLMs [10, 28, 45]. This technique enables MLLMs to generate responses to instructions conditioned on both visual and textual inputs, thereby adapting to various tasks and meeting diverse user needs. For instance, LLaVA-1.5 [28] has built on LLaVA framework [27], improving performance by instruction tuning on a wide range of additional academic-task-oriented visual question answering datasets. Moreover, recently, there has been a growing focus on enhancing In-Context Learning (ICL) and Chain-of-Thought (CoT) performance in various multi-modal scenarios [14, 23, 37, 55, 56]. Representative works include: MIMIC-IT [23], which combines in-context learning with instruction tuning by building an instruction dataset formatted with multi-modal context; Multimodal-CoT [55], which extended the chain-of-thought prompting to multi-modal scenarios, enhancing reasoning capabilities across different modalities. These advancements demonstrate the ongoing efforts to improve the versatility and reasoning capabilities of MLLMs, pushing the boundaries of multi-modal AI systems.

2.2. Trade-off between Stability and Plasticity

The trade-off between stability (the retention of previously learned knowledge) and plasticity (the ability to acquire new information) first emerged in continual learning, where models must learn new tasks sequentially without forgetting previously learned knowledge. This has been a long-standing challenge in machine learning [42]. With the rise of pre-trained language models, this challenge has evolved to focus on mitigating catastrophic forgetting during adaptation, specifically balancing pre-training stability with task-specific plasticity [11, 22]. **(1) Continual Learning:** Over the past few decades, numerous studies have focused on improving continual learning performance, primarily in the context of smaller models [2, 6, 21, 34]. Traditional approaches to continual learning typically fall into three categories: weight regularization [21]), structure expansion [34], and data replay [6]. However, these methods have become less suitable for large pre-trained models due to their computational costs, storage overheads and inability to leverage the unique properties of these pre-trained models, such as loss landscape flatness [53] and extensive world knowledge. Recent studies have explored the potential of leveraging pre-trained models for continual learning [12, 13, 43, 44, 50]. These approaches have significantly improved continual learning performance by utilizing the rich knowledge embedded and properties of flatness in pre-

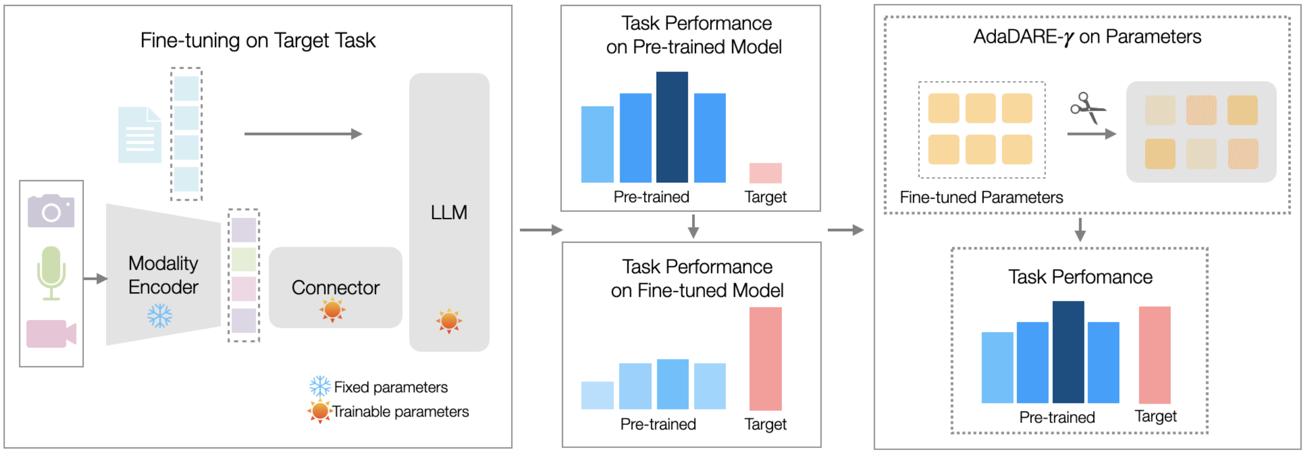

Figure 1. **Fine-tuning MLLMs on Target Task.** Fine-tuning MLLMs on the target task often suffers from catastrophic forgetting, where acquiring new task-specific knowledge compromises performance on pre-trained tasks.

trained models. While these methods have shown promise, they primarily focus on employing either full fine-tuning or PEFT techniques on uni-modal models. However, the application of these approaches to multi-modal scenarios, particularly in the context of MLLMs, remains limited, with only a few studies [8, 17, 31, 39, 49, 54, 57]. **(2) Adaptation in Pre-trained Models:** The challenge of mitigating catastrophic forgetting first gained prominence in language models, where studies revealed that fine-tuning on small datasets often degrades generalization[11, 22, 52]. In NLP filed, to address this challenge, Dong et al. [11] proposed an adversarial training method from an information-theoretical perspective and Lee et al. [22] proposed a mixout regularization technique to control parameter deviation during fine-tuning. In context of MLLMs, Model Tailor [61] addresses the same adaptation settings as our work and explores multi-modal generation and reasoning tasks in MLLMs. However, it requires substantial computational overhead for layer-wise Hessian inverse calculation and lacks flexibility in balancing stability and plasticity.

3. Problem Formulation

We consider a multi-modal large language model $\mathcal{M}$ with initial parameters Θ_{pre}, obtained through pre-training on a diverse set of tasks $\mathcal{P} = \{\mathcal{D}_1, ..., \mathcal{D}_n\}$. When adapting $\mathcal{M}$ to a novel target task $\mathcal{T}$ through fine-tuning, we obtain a new set of parameters Θ_{sft}. This adaptation process can be represented as:

$$\Theta_{\text{sft}} = \Theta_{\text{pre}} + \Delta\Theta, \tag{1}$$

where $\Delta\Theta$ denotes the change in parameters (referred to as delta parameters) between pre-trained and fine-tuned states. To mitigate the performance degradation on pre-trained tasks while optimizing for the target task, we propose a fusion strategy $\mathcal{F}$ that processes Θ_{pre} and Θ_{sft} to derive Θ_{fusion}:

$$\Theta_{\text{fusion}} = \mathcal{F}(\Theta_{\text{pre}}, \Theta_{\text{sft}}). \tag{2}$$

The objective is to find an optimal Θ_{fusion} that satisfies:

$$\begin{aligned} ||\mathcal{L}_{\mathcal{P}}(\Theta_{\text{fusion}}) - \mathcal{L}_{\mathcal{P}}(\Theta_{\text{pre}})|| \leq \epsilon_{\text{pre}}, \\ ||\mathcal{L}_{\mathcal{T}}(\Theta_{\text{fusion}}) - \mathcal{L}_{\mathcal{T}}(\Theta_{\text{sft}})|| \leq \epsilon_{\text{target}}, \end{aligned} \tag{3}$$

where $\mathcal{L}_{\mathcal{P}}(\cdot)$ represents the aggregate loss on the pre-training tasks $\mathcal{P}$, $\mathcal{L}_{\mathcal{T}}(\cdot)$ denotes the loss on the target task $\mathcal{T}$. And $\epsilon_{\text{pre}} > 0$ is the maximum allowable performance degradation on pre-training tasks.

4. Method

4.1. Overview of AdaDARE-γ

In this section, we introduce AdaDARE-γ, our novel approach to efficient adaptation in MLLMs. AdaDARE-γ builds upon the foundation of DARE while introducing layer-wise optimization for adaptive parameter selection and a γ factor for controlled task-specific information injection. The overall framework is illustrated in Fig. 2. The following subsections detail each component of our method, starting with our objective and a review of the DARE approach, followed by our enhancements and theoretical analysis.

4.2. Layer-wise Objective

Given a pre-trained model with parameters Θ_{pre} and its fine-tuned version Θ_{sft}, our goal is to find optimal fusion parameters Θ_{fusion} that maintain performance on both pre-trained and target tasks. We formulate this as a layer-wise optimiza-

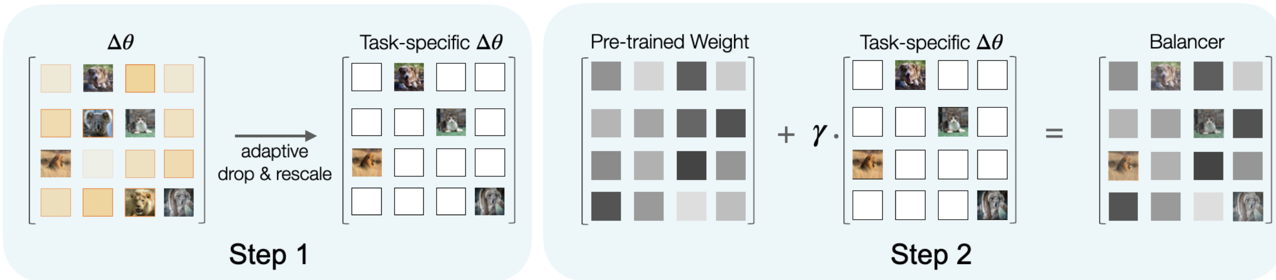

Figure 2. **Illustration of Adaptive-DARE with γ factor.** Two-step parameter adaptation for task-specific learning: (1) Adaptive drop and rescale of task-specific delta parameters ($\Delta\theta$), selectively removing parameters based on varying probabilities to eliminate redundant or task-irrelevant information. (2) Balancing pre-trained knowledge with adapted task-specific information using a scaling factor γ to achieve optimal performance across general and target tasks.

tion problem:

$$\begin{aligned} &\arg\min_{\Theta^{\ell}_{\text{fusion}}} \mathcal{L}\left(f_\ell(\mathcal{X}^{\ell}_{\mathcal{T}}, \Theta^{\ell}_{\text{sft}}), f_\ell(\mathcal{X}^{\ell}_{\mathcal{T}}, \Theta^{\ell}_{\text{fusion}})\right), \\ &\text{s.t. } \mathbb{E}\left[\mathcal{L}(f_\ell(\mathcal{X}^{\ell}_{\mathcal{T}}, \Theta^{\ell}_{\text{pre}}), f_\ell(\mathcal{X}^{\ell}_{\mathcal{T}}, \Theta^{\ell}_{\text{fusion}}))\right] \le \epsilon. \end{aligned} \tag{4}$$

However, in practical scenarios, access to pre-training data is often restricted due to proprietary or privacy considerations. Therefore, we approximate the constraint by measuring the parameter distance between Θ_{fusion} and Θ_{pre}:

$$\begin{aligned} &\arg\min_{\Theta^{\ell}_{\text{fusion}}} \left\|\Theta^{\ell}_{\text{sft}}\mathbf{X}^{\ell}_{\mathcal{T}} - \Theta^{\ell}_{\text{fusion}}\mathbf{X}^{\ell}_{\mathcal{T}}\right\|_2^2, \\ &\text{s.t. } \left\|\Theta^{\ell}_{\text{fusion}} - \Theta^{\ell}_{\text{pre}}\right\|_1 < \eta. \end{aligned} \tag{5}$$

4.3. Preliminary: Drop And REscale

The Drop And REscale (DARE) method forms the basis of our approach. DARE introduced a groundbreaking insight: in Supervised Fine-Tuned (SFT) language models, a significant portion of delta parameters can be zeroed out while preserving model capabilities. This finding suggests that model adaptation often involves substantial parameter redundancy, which can be efficiently managed through randomly dropping and rescaling.

The DARE process can be formally described by the following equations:

$$\begin{aligned} M^{\mathcal{T}} &\sim \text{Bernoulli}(p), \\ \tilde{\Delta\Theta}^{\mathcal{T}} &= (1 - M^{\mathcal{T}}) \odot \Delta\Theta^{\mathcal{T}}, \\ \hat{\Delta\Theta}^{\mathcal{T}} &= \tilde{\Delta\Theta}^{\mathcal{T}} / (1-p), \end{aligned} \tag{6}$$

where $M^{\mathcal{T}}$ is a binary mask with each element sampled from a Bernoulli distribution with probability $p \in [0, 1)$, $\Delta\Theta^{\mathcal{T}}$ represents the delta parameters, $\odot$ denotes element-wise multiplication and / denotes element-wise division.

Notably, the drop rate p directly corresponds to the sparsity ratio, where a drop rate of p yields a $(100p)\%$ sparsity of delta parameters. The rescaling factor $1/(1-p)$ ensures that the expected values of the remaining parameters compensate for the dropped ones.

To understand the simple theoretical foundation of DARE, we can examine its effect on the expectation of embeddings. Consider the i-th dimension:

$$\begin{aligned} \mathbb{E}[\hat{h}_i] &= \mathbb{E}\left[\sum_{j=1}^{n}(w_{ij} + \Delta\hat{w}_{ij})x_j + (b_i + \Delta\hat{b}_i)\right] \\ &= \sum_{j=1}^{n} x_j \mathbb{E}[w_{ij}] + \mathbb{E}[b_i] + \sum_{j=1}^{n} x_j \mathbb{E}[\Delta\hat{w}_{ij}] + \mathbb{E}[\Delta\hat{b}_i] \\ &= \sum_{j=1}^{n} w_{ij}x_j + b_i + \sum_{j=1}^{n} x_j \left(\frac{(1-p)\cdot\Delta w_{ij}}{1-p} + p\cdot 0\right) \\ &\quad + \left(\frac{(1-p)\cdot\Delta b_i}{1-p} + p\cdot 0\right) \\ &= h_i^{\text{pre}} + \Delta h_i. \end{aligned} \tag{7}$$

This derivation shows that DARE approximates the fine-tuned embeddings while incorporating a scaled version of the changes introduced by fine-tuning.

4.4. AdaDARE-γ: Methodology

Our method introduces a principled framework for efficient adaptation in MLLMs through two key innovations: adaptive parameter-wise probabilities and a γ factor for controlled task-specific information injection.

4.4.1. Adaptive Parameter Selection

We propose an adaptive parameter selection mechanism that leverages the properties of parameter importance to determine optimal retention strategies. This approach extends beyond simple drop and rescale by introducing parameter-wise probabilistic selection. The process can be formalized as:

$$\begin{aligned} M^{\mathcal{T},\ell} &\sim \text{Bernoulli}(P^{\ell}), \\ \tilde{\Delta\Theta}^{\mathcal{T},\ell} &= (1 - M^{\mathcal{T},\ell}) \odot \Delta\Theta^{\mathcal{T},\ell}, \\ \hat{\Delta\Theta}^{\mathcal{T},\ell} &= \tilde{\Delta\Theta}^{\mathcal{T},\ell} / (1 - P^{\ell}), \end{aligned} \tag{8}$$

where P^{ℓ} is a probability matrix with the same shape as $\Delta\Theta^{\mathcal{T},\ell}$ with each element in P^{ℓ} specifying the drop probability for the corresponding delta parameter in layer ℓ. $M^{\mathcal{T},\ell}$ is a binary mask, where each element is sampled from a Bernoulli distribution with the probability defined by P^{ℓ}. And $/$ denotes element-wise division. By replacing p with p_j (j-th dimension of P^{ℓ}) in Eq. (7), it's easy to show this process effectively drops parameters with their respective probabilities and rescales the remaining ones, maintaining the expected value of the embeddings for each dimension.

4.4.2. Controlled Task-specific Information Injection

We introduce a γ factor to regulate the injection of task-specific information into pre-trained knowledge:

$$\begin{aligned} \Theta_{\text{fusion}}^{\mathcal{T},\ell} &= \mathcal{F}(\Theta_{\text{pre}}^{\mathcal{T},\ell}, \Theta_{\text{sft}}^{\mathcal{T},\ell}) \\ &= \Theta_{\text{pre}}^{\mathcal{T},\ell} + \gamma \cdot \hat{\Delta\Theta}^{\mathcal{T},\ell}. \end{aligned} \tag{9}$$

This equation combines the pre-trained parameters with a scaled version of the adaptive selected delta parameters. The γ factor allows us to balance between retaining old knowledge (when γ is closer to 0) and incorporating new information (when γ is closer to 1).

4.5. Theoretical Analysis

4.5.1. Optimizing Drop Probabilities

To determine the optimal drop probabilities, we formulate an optimization problem that minimizes the expected layer-wise loss while maintaining a desired ratio of sparsity. This leads to our key theorem:

Theorem 1 (Optimal Probabilities) *Consider the optimization problem of minimizing the expected layer-wise loss:*

$$\arg\min_{P^{\ell}} \mathbb{E}\left[\left\|\Theta_{sft}^{\ell}\mathcal{X}_{\mathcal{T}}^{\ell} - \Theta_{fusion}^{\ell}\mathcal{X}_{\mathcal{T}}^{\ell}\right\|_2^2\right], \tag{10}$$

subject to the constraints:

$$\frac{\sum_{i=1}^{n} p_i}{n} >= p, \quad and \quad 0 \le p_i < 1 \quad \forall i. \tag{11}$$

where p represents the desired sparsity ratio. Then, the optimal probabilities p_i^* that minimize the expected loss are given by:

$$p_i^* = \max(0, 1 - \frac{n(1-p)\sqrt{H_{ii}\delta_i^2}}{\sum_{j=1}^{n}\sqrt{H_{jj}\delta_j^2}}) \quad \forall i. \tag{12}$$

Here, H_{ii} represents the i-th diagonal element of the Hessian matrix, and δ_i is the i-th element of $\Delta\Theta^{\mathcal{T},\ell}$.

The detailed proof of this theorem can be found in Appendix A. This theorem provides us with a way to adaptively determine which parameters are more important to retain based on their contribution to the loss and the model's sensitivity to their changes.

4.5.2. Theoretical Bounds on γ

Theorem 2 (upper bound of γ) *Given the constraint $\mathbb{E}\left[\left\|\Theta_{fusion}^{\ell} - \Theta_{pre}^{\ell}\right\|_1\right] < \eta$, there exists an upper bound for γ:*

$$\gamma \le \frac{\eta}{\sum_{i=1}^{n}|\delta_i|}, \tag{13}$$

where δ_i represents the i-th element of the delta parameters $\Delta\Theta^{\mathcal{T},\ell}$.

The detailed proof of this theorem can be found in Appendix A. This bound ensures that parameter changes remain within the constraint specified in Eq. (5), maintaining the optimization objective due to its monotonic relationship with γ. The relationship between γ and η provides a theoretical guarantee for controlled information injection, enabling precise balance between knowledge retention and task adaptation.

4.6. Pratical Implications of γ Bounds

As illustrated in Theorem 2, $\gamma \le \frac{\eta}{\sum_{i=1}^{n}|\delta_i|}$, where η represents our desired pre-trained performance constraint and δ_i reflects parameter changes. This relationship has important practical implications. The η parameter controls how much pre-trained task performance we aim to preserve. Similar to neural network training where absolute loss values don't directly map to dataset performance, we validate γ's effectiveness on validation sets, which represents generalization ability, rather than explicitly setting η. This approach allows practical tuning while respecting theoretical bounds.

5. Experiments

5.1. Experiments Setup

Architectures and Datasets. Our experiments focus on two representative multi-modal models: InstructBLIP [10] and LLaVA-1.5 [28], both based on Vicuna-7B [58]. These models differ in their fine-tuning approaches: InstructBLIP adjusts only the Q-Former, whereas LLaVA-1.5 fine-tunes the language model with an optional projector update [59]. For InstructBLIP, we evaluated image captioning datasets including COCO Caption [26] and NoCaps (with in-domain, near-domain and out-of-domain subsets) [3], as well as VQA tasks such as OKVQA [32], AOKVQA [35], and VQAv2 [15]. We fine-tuned InstructBLIP on Flickr30k [47] for image captioning and GQA [19] for VQA, both of which were unseen during InstructBLIP's pre-training phase. In the case of LLaVA-1.5, our evaluation encompassed a diverse range of datasets: VQAv2, GQA, Vizwiz [16], SQA [30], TextVQA [38], POPE [25], MM-Bench [29] and MM-Bench-CN [51]. For LLaVA-1.5, we conducted fine-tuning on Flickr30k for image captioning and OKVQA for VQA, both distinct from LLaVA-1.5's pre-trained datasets.

Compared Baselines. We compare our method against three existing approaches. The first is **Fine-Tuning**, where for InstructBLIP, the process targets the Q-Former, a connector between the vision encoder and the LLM, involving approximately 188M parameters. For LLaVA-1.5, fine-tuning is applied to 11 layers, uniformly sampling from the 32-layer LLM at equal intervals, totaling 2.5 billion parameters (constrained by GPU memory limits). The second approach is **Model-Tailor** [61], which employs a two-step strategy: initially, it retains only about 10% of delta parameters based on a specific selection strategy, discarding the rest; subsequently, it compensates the remaining delta parameters to mitigate performance loss on the target task, then fuses the selected and decorated parameters with pre-trained parameters. The third method is **DARE** [48], which randomly selects and rescales delta parameters, then fuses them with pre-trained parameters to maintain the model's performance on the target task. Similar to our method, both Model-Tailor and DARE achieve parameter-efficient adaptation without requiring model re-training.

Evaluation Metrics. We employ two complementary metrics to evaluate MLLM performance. The first is arithmetic mean across both pre-trained and target tasks, providing an overall performance measure:

$$\text{Avg} = \text{Mean}(\text{score}(\mathcal{P}), \text{score}(\mathcal{T})). \tag{14}$$

The second is the H-score, defined as the harmonic mean between pre-trained and target task performances:

$$\text{H-score} = \frac{2 * \text{Mean}(\text{score}(\mathcal{P})) * \text{Mean}(\text{score}(\mathcal{T}))}{\text{Mean}(\text{score}(\mathcal{P})) + \text{Mean}(\text{score}(\mathcal{T}))}, \tag{15}$$

where $\text{score}(\mathcal{P})$ and $\text{score}(\mathcal{T})$ represent performance scores on pre-trained and target tasks respectively. The H-score is particularly sensitive to performance imbalances, providing a rigorous measure of the trade-off between retaining pre-trained knowledge and adapting to new tasks.

Implementation Details. All experiments were conducted using 8 A100 GPUs, each with 40GB memory. For InstructBLIP fine-tuning on both Flickr30k and GQA, we performed a grid search over learning rates in the range [1e-6, 1e-5], identifying 1e-5 as optimal. The training utilized a batch size of 16 per GPU for 5 epochs. For LLaVA-1.5, we explored learning rates in the range [1e-6, 1e-4], with optimal values of 5e-6 for both Flickr30k and OKVQA. Similarly, we maintained a batch size of 16 per GPU and trained for 5 epochs in both cases. When combined with LoRA, we set the LoRA rank to 128 and learning rate to 5e-6, maintaining the same batch size and training epochs as in the standard fine-tuning experiments. Specifically, we tune γ within the range [0.4, 0.8], assessing specialization on the target validation dataset and generalization ability on the VQAv2 validation dataset.

5.2. Experimental Results

Now, we present the results of our empirical investigation, including the overall performance of our method and compared baselines, an ablation study of adaptive selection mechanism and the trade-off between stability and plasticity of tuning γ and a complexity analysis.

5.2.1. Overall Performance:

Tab. 1 and Tab. 2 present the experimental results for fine-tuning InstructBLIP and LLaVA-1.5 across different datasets. For both models, we maintain a 90% sparsity ratio across all parameter-efficient methods and set $\gamma = 0.5$ for AdaDARE-γ. The results demonstrate that AdaDARE-γ consistently achieves the best H-score across all experimental settings, achieving optimal trade-off between model stability and plasticity. Notably, while maintaining strong performance on pre-trained tasks, AdaDARE-γ achieves comparable or even better performance on target tasks compared to full fine-tuning (99.7 vs. 101.3, 57.22 vs. 60.53 on InstructBLIP and 90.1 vs. 93.2, 57.21 vs. 55.72 on LLaVA-1.5). In contrast, although DARE achieves almost the same target performance as full fine-tuning, it shows significant performance degradation on pre-trained tasks. Model-Tailor, when only fine-tuning the Q-former in InstructBLIP, exhibits sub-optimal target task performance (e.g., 81.8 vs. 101.1 on Flickr30k and 54.36 vs. 60.53 on GQA).

5.2.2. Combination with LoRA

AdaDARE-γ is compatible with existing parameter-efficient fine-tuning (PEFT) methods such as LoRA [18]. While LoRA dramatically reduces trainable parameters by freezing pre-trained weights and introducing low-rank decomposition matrices, our experiments fine-tuning LLaVA-1.5 with LoRA on Flickr30K and OKVQA demonstrate that it still encounters catastrophic forgetting. As shown in Fig. 3, incorporating AdaDARE-γ into LoRA-based fine-tuning yields significant performance gains across multiple datasets, outperforming existing methods on most datasets as well as in Average and H-score metrics. This confirms AdaDARE-γ's effectiveness as a complementary technique to PEFT approaches.

5.2.3. Ablation Study

Adaptive Selection Enhances Target Task Performance and Drop Rate Robustness. We conduct ablation studies comparing our adaptive selection strategy with DARE's

Table 1. **InstructBLIP Fine-tuning on Flickr30k and GQA.** All parameter-efficient methods maintain 90% sparsity ratio with $\gamma = 0.5$ for AdaDARE-γ. Results are reported on both pre-trained tasks (COCO, NoCaps, OKVQA, AOKVQA, VQAv2) and the target task (Flickr30k, GQA). #Params denotes the number of trainable parameters. The optimal and sub-optimal results are denoted by boldface and underlining.

Method	#Params	Pre-trained tasks								Target task	Metrics	
		COCO	NoCaps-in	NoCaps-near	NoCaps-out	OKVQA	AOKVQA	VQAv2	GQA	Flickr30k	Avg	H-score
Zero-shot	-	143.0	116.9	124.0	122.6	57.26	60.75	76.65	49.19	83.4	92.64	88.29
Fine-tune	188M	123.2	109.0	120.0	120.0	45.54	55.31	61.03	43.91	101.3	86.58	92.28
DARE	18.8M	123.6	110.1	120.3	119.9	45.59	55.34	60.99	43.98	101.1	86.76	92.33
Model-Tailor	18.8M	139.7	116.5	125.0	125.9	56.50	60.75	70.84	49.15	96.4	<u>93.41</u>	<u>94.69</u>
AdaDARE-γ	18.8M	137.6	116.2	125.4	126.6	55.71	60.29	70.46	48.95	99.7	**93.43**	**96.05**
Method	#Params	Pre-trained tasks								Target task	Metrics	
		COCO	NoCaps-in	NoCaps-near	NoCaps-out	OKVQA	AOKVQA	VQAv2	Flickr30k	GQA	Avg	H-score
Zero-shot	-	143.0	116.9	124.0	122.6	57.26	60.75	76.65	83.4	49.19	**92.64**	65.51
Fine-tune	188M	123.6	101.4	109.2	110.7	34.95	44.1	64.17	72.5	60.53	80.13	69.85
DARE	18.8M	125.6	102.1	110.7	112.2	35.84	45.32	64.31	74.2	60.28	81.17	<u>70.11</u>
Model-Tailor	18.8M	141.5	116.5	122.7	121.4	55.23	59.79	71.04	81.8	54.36	<u>91.59</u>	69.47
AdaDARE-γ	18.8M	139.6	114.1	122.0	120.3	51.56	57.75	70.14	81.1	57.22	90.42	**71.30**

Table 2. **LLaVA-1.5 Fine-tuning on Flickr30k and OKVQA.** All parameter-efficient methods maintain 90% sparsity ratio with $\gamma = 0.5$ for AdaDARE-γ. Results are reported on both pre-trained tasks (VQAv2, GQA, VizWiz, SQA, TextVQA, POPE, MM-Bench, MM-Bench-CN) and the target task (Flickr30k, OKVQA). #Params denotes the number of trainable parameters. The optimal and sub-optimal results are denoted by boldface and underlining.

Method	#Params	Pre-trained tasks								Target task	Metrics	
		VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	Flickr30k	Avg	H-score
Zero-shot	-	78.52	61.94	50.06	70.22	58.21	86.1	64.6	58.1	12.2	59.99	20.59
Fine-tune	2.5B	75.27	60.05	25.63	66.28	52.07	87.7	56.7	52.92	93.2	63.31	72.68
DARE	248M	76.7	60.8	27.05	66.89	52.78	86.2	57.3	53.86	91.9	63.72	72.74
Model-Tailor	248M	77.74	61.46	38.41	69.02	57.17	87.0	65.12	57.64	91.9	<u>67.27</u>	<u>75.58</u>
AdaDARE-γ	248M	77.89	61.61	47.83	69.28	57.53	87.1	65.21	57.99	90.1	**68.28**	**75.89**
Method	#Params	Pre-trained tasks								Target task	Metrics	
		VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	OKVQA	Avg	H-score
Zero-shot	-	78.52	61.94	50.06	70.22	58.21	86.1	64.6	58.1	0.02	58.64	0.04
Fine-tune	2.5B	70.24	51.79	42.37	52.06	53.16	65.6	45.79	57.98	55.72	54.96	55.29
DARE	248M	71.35	53.08	42.87	52.25	53.92	69	42.52	57.64	56.45	55.45	55.88
Model-Tailor	248M	75.01	59.1	48.33	69.16	55.96	85	64.17	58.67	54.86	<u>63.36</u>	<u>59.25</u>
AdaDARE-γ	248M	75.52	59.51	48.81	69.49	56.42	85	64.78	58.33	57.21	**63.90**	**60.74**

Note: We observed that fine-tuning only the last 12 layers of the LLM leads to substantially lower performance compared to PEFT methods, as reported in [59]. In response, we developed a more reasonable fine-tuning approach that, under limited GPU resources, achieves superior performance compared to both the baseline fine-tuning results and the Model-Tailor approach reported in Model-Tailor's paper.

random selection approach. As shown in Fig. 3, adaptive selection not only improves target task performance but also significantly enhances model robustness under high drop rates.

Performance Trade-off with γ. Performance Trade-off with γ. We investigate the impact of different γ values on model performance. As illustrated in Fig. 4, γ regulates the injection of task-specific information, creating a balance of stability and plasticity. As established in Theorem 2, $\gamma \le \frac{\eta}{\sum_{i=1}^{n} |\delta_i|}$, where η represents our desired pre-trained performance constraint and δ_i reflects parameter changes. This relationship manifests clearly in our experiments: (1) For full fine-tuning across datasets, we set $\gamma = 0.5$ as used in Tab. 1 and Tab. 2; (2) For parameter-efficient fine-tuning (PEFT) with LoRA shown in Tab. 3, we set $\gamma = 0.7$. This difference arises because PEFT induces smaller parameter changes (δ_i), allowing for larger γ values while maintaining the theoretical bound. As shown in Fig. 4, the method remains highly stable for $\gamma \in [0.5, 0.8]$. This stability across a range of γ values demonstrates the robustness of our approach in practical settings.

5.2.4. Computational Complexity Analysis

The computational complexity of our method consists of two main components: (1) Initial Hessian Computation:

Table 3. **Combination with LoRA on LLaVA.** Performance comparison of AdaDARE-γ ($\gamma = 0.7$, sparsity ratio = 90%) combined with LoRA-based LLaVA-1.5 fine-tuning and Model-Tailor across multiple datasets. Results demonstrate significant improvements in maintaining both pre-trained and target task performance.

Method	Pre-trained tasks								Target task	Metrics	
	VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	Flickr30k	Avg	H-score
LoRA	72.76	57.45	8.29	62.44	40.44	80.70	51.63	33.76	89.50	53.03	62.33
Model-Tailor	78.04	61.26	26.79	68.03	53.96	86.50	60.22	53.78	89.70	64.25	72.67
AdaDARE-γ	78.36	61.65	32.45	68.92	55.82	86.10	62.97	56.44	89.20	**65.77**	**73.73**
Method	Pre-trained tasks								Target task	Metrics	
	VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	OKVQA	Avg	H-score
LoRA	25.51	53.25	43.11	62.18	54.47	70.05	50.25	56.78	58.11	52.63	54.85
Model-Tailor	74.15	57.09	48.09	68.52	55.45	81.49	63.14	58.24	59.28	62.82	61.21
AdaDARE-γ	74.66	58.19	47.76	68.69	55.87	84.20	64.43	58.67	58.97	**63.49**	**61.41**

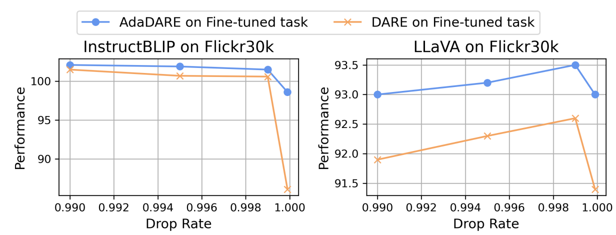

Figure 3. Performance comparison between AdaDARE-γ ($\gamma = 1$) and DARE on Flickr30k across various drop rates. Results demonstrate that AdaDARE-γ consistently outperforms DARE, particularly showing enhanced robustness at higher drop rates.

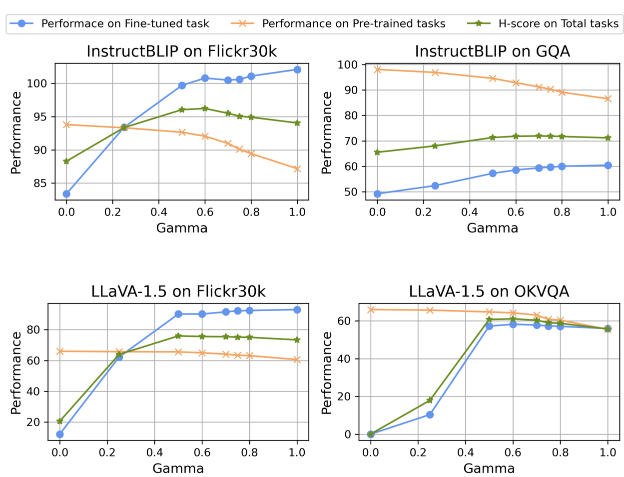

Figure 4. Performance trade-off across different γ values on InstructBLIP (Flickr30k, GQA) and LLaVA-1.5 (Flickr30k, OKVQA). The plots demonstrate that γ effectively controls the stability-plasticity trade-off.

Computing the initial Hessian matrix ($H = 2XX^T + \lambda I$, where X represents input samples) has a time complexity of $\mathcal{O}(nd_{col}^2)$, where n denotes the number of input samples

Table 4. Computational complexity comparison of different methods. n denotes the number of input samples, d_{col} and d_{row} represent the column and row dimensionality of the weight matrix, respectively.

Method	Computational Complexity
DARE	$\mathcal{O}(1)$
AdaDARE-γ	$\mathcal{O}(nd_{col}^2 + d_{row}d_{col})$
Model-Tailor	$\mathcal{O}(nd_{col}^2 + d_{col}^3 + d_{col}d_{row}^2)$

(set to 128 in our experiments) and d_{col} represents the column dimensionality of the matrix. (2) Adaptive Drop and Rescale: This process primarily involves calculating optimal probabilities with a time complexity of $\mathcal{O}(d_{row}d_{col})$, where d_{row} denotes the row dimensionality of the matrix.

6. Conclusion and Limitation

In this paper, we present AdaDARE-γ, demonstrating effective MLLM adaptation through principled parameter selection and controlled knowledge injection. Our method addresses the stability-plasticity dilemma by identifying crucial parameters and regulating their integration with pre-trained knowledge, achieving state-of-the-art performance across multi-modal tasks with computational efficiency. While promising, our approach relies on manual tuning of γ, and theoretical bounds may not directly align with optimal practical values. Future work could explore automatic optimization of γ and better connections between theory and practice for optimal balance between adaptation and knowledge preservation. Our framework provides valuable insights for efficient multi-modal adaptation.

Acknowledgements

This work was supported by the NSFC under grant No. 62401029 and No. 62206308.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 2

[2] Tameem Adel, Han Zhao, and Richard E. Turner. Continual learning with adaptive weights (claw), 2020. 1, 2

[3] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019. 5

[4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. 2

[5] 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. 1, 2

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem, 2019. 2

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023. 1, 2

[8] Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. Ratt: Recurrent attention to transient tasks for continual image captioning, 2020. 3

[9] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. 2

[10] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 2, 5

[11] Xinhsuai Dong, Luu Anh Tuan, Min Lin, Shuicheng Yan, and Hanwang Zhang. How should pre-trained language models be fine-tuned towards adversarial robustness?, 2021. 2, 3

[12] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin, 2024. 2

[13] Enrico Fini, Victor G. Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners, 2022. 2

[14] Jiaxin Ge, Hongyin Luo, Siyuan Qian, Yulu Gan, Jie Fu, and Shanghang Zhang. Chain of thought prompt tuning in vision language models, 2023. 2

[15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering, 2017. 5

[16] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people, 2018. 6

[17] Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models, 2023. 1, 3

[18] 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, 2021. 6

[19] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. 5

[20] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023. 2

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2

[22] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models, 2020. 2, 3

[23] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning, 2023. 2

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 2

[25] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. 6

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva

Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 5

[27] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2

[28] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024. 1, 2, 5

[29] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024. 6

[30] 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. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 6

[31] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2024. 1, 3

[32] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge, 2019. 5

[33] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 1, 2

[34] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks, 2022. 1, 2

[35] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge, 2022. 5

[36] Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners, 2022. 1

[37] Dianmo Sheng, Dongdong Chen, Zhentao Tan, Qiankun Liu, Qi Chu, Jianmin Bao, Tao Gong, Bin Liu, Shengwei Xu, and Nenghai Yu. Towards more unified in-context visual understanding, 2024. 2

[38] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read, 2019. 6

[39] James Seale Smith, Paola Cascante-Bonilla, Assaf Arbelle, Donghyun Kim, Rameswar Panda, David Cox, Diyi Yang, Zsolt Kira, Rogerio Feris, and Leonid Karlinsky. Construct-vl: Data-free continual structured vl concepts learning, 2023. 3

[40] 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. 1

[41] 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. 1, 2

[42] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. 1, 2

[43] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning, 2022. 2

[44] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning, 2022. 1, 2

[45] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 1, 2

[46] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023. 1, 2

[47] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 5

[48] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2024. 6

[49] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models, 2023. 1, 3

[50] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model, 2023. 2

[51] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Haodong Duan, Songyang Zhang, Shuangrui Ding, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition, 2023. 6

[52] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning, 2021. 3

[53] Xingxuan Zhang, Renzhe Xu, Han Yu, Yancheng Dong, Pengfei Tian, and Peng Cui. Flatness-aware minimization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5189–5202, 2023. 2

[54] Xi Zhang, Feifei Zhang, and Changsheng Xu. Vqacl: A novel visual question answering continual learning setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19102–19112, 2023. 3

[55] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models, 2024. 2

[56] Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. Mmicl: Empowering vision-language model with multi-modal in-context learning, 2024. 2

[57] Junhao Zheng, Qianli Ma, Zhen Liu, Binquan Wu, and Huawen Feng. Beyond anti-forgetting: Multimodal continual instruction tuning with positive forward transfer, 2024. 3

[58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 5

[59] Xiongtao Zhou, Jie He, Yuhua Ke, Guangyao Zhu, Víctor Gutiérrez-Basulto, and Jeff Z. Pan. An empirical study on parameter-efficient fine-tuning for multimodal large language models, 2024. 5, 7

[60] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. 2

[61] Didi Zhu, Zhongyi Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Kun Kuang, and Chao Wu. Model tailor: Mitigating catastrophic forgetting in multi-modal large language models, 2024. 1, 3, 6