Merge then Realign: Simple and Effective Modality-Incremental Continual Learning for Multimodal LLMs

Dingkun Zhang[1], **Shuhan Qi**[1,2*], **Xinyu Xiao**[1], **Kehai Chen**[1], **Xuan Wang**[1],
[1]Harbin Institute of Technology, Shenzhen
[2]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
dingkunzhang0xffff@gmail.com, {shuhanqi,wangxuan}@cs.hitsz.edu.cn,
23b951021@stu.hit.edu.cn, chenkehai@hit.edu.cn

Abstract

Recent advances in Multimodal Large Language Models (MLLMs) have enhanced their versatility as they integrate a growing number of modalities. Considering the heavy cost of training MLLMs, it is efficient to reuse the existing ones and extend them to more modalities through Modality-incremental Continual Learning (MCL). The exploration of MCL is in its early stages. In this work, we dive into the causes of performance degradation in MCL. We uncover that it suffers not only from forgetting as in traditional continual learning, but also from misalignment between the modality-agnostic and modality-specific components. To this end, we propose an elegantly simple MCL paradigm called "MErge then ReAlign" (MERA) to address both forgetting and misalignment. MERA avoids introducing heavy model budgets or modifying model architectures, hence is easy to deploy and highly reusable in the MLLM community. Extensive experiments demonstrate the impressive performance of MERA, holding an average of 99.84% Backward Relative Gain when extending to four modalities, achieving nearly lossless MCL performance. Our findings underscore the misalignment issue in MCL. More broadly, our work showcases how to adjust different components of MLLMs during continual learning.

1 Introduction

With the recent trend of developing general-purpose any-modality Multimodal Large Language Models (MLLMs) (Panagopoulou et al., 2023; Chen et al., 2023a; Wu et al., 2024; Han et al., 2024; Zhan et al., 2024; Shuhan et al., 2023), MLLMs are evolving towards integrating more modalities. The typical MLLM architecture includes modality-specific encoders, modality-specific connectors, and a shared Large Language Model (LLM). A standard process of training MLLMs involves aligning modality-specific components with LLM through modality-text paired data and then fine-tuning on modality-text instruction data (Rao et al., 2024). Such architecture and training strategy have been successfully applied to a wide range of modalities, i.e., image (Liu et al., 2024b,a; Rao et al., 2023), video (Lin et al., 2024a; Maaz et al., 2024), audio (Li et al., 2024; Wu et al., 2024), point cloud (Chen et al., 2024), etc, equipping MLLMs with the ability to understand a growing number of modalities. Existing methods (Wu et al., 2024; Zhan et al., 2024; Panagopoulou et al., 2023; Fu et al., 2024) typically employ a joint training strategy, where the MLLM is jointly trained on datasets of all pre-defined modalities (Xin et al., 2024). However, it is challenging to extend an existing MLLM to new modalities as it requires another round of joint training on the previous modalities and the new modalities.

To reuse the existing models and adapt them to new data, Continual Learning (CL) is proposed to learn from a stream of data. During continual learning, performance degradation in previously learned tasks often occurs. The degradation is generally attributed to catastrophic forgetting (McCloskey and Cohen, 1989; Goodfellow et al., 2014; Rao et al., 2025), i.e., the model forgets the previously learned knowledge. To this end, many CL methods (Kirkpatrick et al., 2017; Yu et al., 2024b; Scialom et al., 2022; Wang et al., 2024b) have been proposed to alleviate catastrophic forgetting.

In addition to traditional CL, Modality-incremental Continual Learning (MCL) (Yu et al., 2024a) focuses on the particular scenario of incrementally extending MLLMs to new modalities. The exploration of MCL is in its early stages. In this work, we first analyze the causes of performance degradation in MCL. Unlike traditional CL, the performance degradation encountered in MCL comes not only from forgetting but also from the misalignment between modality-agnostic and

*Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 13148–13164
November 4-9, 2025

modality-specific components.

To address both forgetting and misalignment, we propose a simple yet effective two-stage MCL paradigm called "MErge then ReAlign" (MERA).

The first stage of MERA aims at addressing the forgetting problem. To this end, we introduce model merging to our MCL framework. We focus on the simplest model merging method, i.e., weight averaging, and revise it into an MCL form. We achieve this by associating its merging coefficients with the progress of CL stages and only merging the modality-agnostic components.

The second stage of MERA aims at addressing the misalignment problem. We leverage a small subset of data from each learned modality to realign the modality encoders with the LLM backbone. In this stage, modality encoders and LLM backbone are both frozen, only the lightweight connectors are updated to enable an efficient realignment between them. Further experiments show that the realigning stage can significantly narrow the gap between the incrementally learned MLLM and the individually trained expert MLLMs on each modality.

In summary, the contributions of this paper are threefold:

- We analyze the causes of degradation in Modality-incremental Continual Learning (MCL). We uncover that it suffers not only from forgetting as in traditional continual learning, but also from misalignment.
- We propose "MErge then ReAlign" (MERA), an elegantly simple and effective two-stage MCL paradigm, to address both forgetting and misalignment.
- Extensive experiments show that our MERA significantly outperforms other representative continual learning methods including the state-of-the-art MCL method, and **even achieves nearly lossless MCL performance**.

2 Related Work

2.1 Multimodal Large Language Models

Recent advances (Panagopoulou et al., 2023; Chen et al., 2023a; Wu et al., 2024; Han et al., 2024; Zhan et al., 2024) in MLLM have extended LLMs to perceive multimodal inputs such as image, video, audio, point cloud, etc. Among all the MLLMs, the most influential one is LLaVA (Liu et al., 2024b,a), which utilizes a simple MLP connector to project visual information encoded by the pre-trained vision encoder into the language embedding space. Due to its simplicity and effectiveness, LLaVA-like architecture is widely adopted by a wide range of subsequent MLLMs (Lin et al., 2024b,a; Maaz et al., 2024; Wu et al., 2024; Chen et al., 2024) and accounts for the majority of current MLLMs. In this paper, we assume that the MLLM has a LLaVA-like architecture that includes modality-specific encoders and connectors, and a shared modality-agnostic LLM backbone.

The rapid development of MLLMs demands high efficiency in their training process. It is efficient to reuse the existing MLLMs and extend them to more modalities. However, directly fine-tuning MLLMs on new modalities often results in significant performance degradation in previously learned modalities. In this work, we leverage the continual learning technique to tackle this problem.

2.2 Traditional Continual Learning

Continual Learning (CL) (Van de Ven and Tolias, 2019; Wang et al., 2024a) aims to continually acquire new knowledge with minor forgetting of previously learned knowledge. Existing CL methods mainly fall into the following four categories: **Regularization-based methods** (Kirkpatrick et al., 2017; Huszár, 2017; Schwarz et al., 2018) seek to protect the parameters that store important knowledge. However, storing an importance matrix during training requires extra memory with the same scale as the trainable parameters. **Architecture-based methods** (Yu et al., 2024a,b; Zadouri et al., 2024; Srinivasan et al., 2023) add task-specific parameters to the base model for each new task. This category requires modifications to the model architecture, harming its reusability. For many methods in this category, the model scale grows linearly as tasks increase, introducing extra memory overhead. **Replay-based methods** (Rolnick et al., 2019; Scialom et al., 2022; Wang et al., 2024b) leverage a small subset of historical data and replay it when learning on new data. This category requires access to partial data from previous tasks or distributions. However, this drawback is relatively minor in real applications since the replay data is often accessible. **Merging-based methods** (Wortsman et al., 2022; Marczak et al., 2024; Xiao et al., 2024; Zhu et al., 2024) edit models in parameter space to integrate the previously learned knowledge into the fine-tuned models by model merging (Yadav et al., 2024a; Yu et al., 2024c; Yang et al.,

	Extra-Train-Memory-Free	Arch-Modification-Free	Replay-Data-Free
Regularization-Based	✘	✔	✔
Architecture-Based	•	✘	✔
Replay-Based	✔	✔	✘
Merging-Based	✔	✔	✔
MERA (Ours)	✔	✔	✘

Table 1: Characteristics of different CL categories and our proposed MERA. Extra-train-memory-free: does not introduce extra GPU memory overhead at training time. Arch-modification-free: does not modify the architecture of the model or add auxiliary components. Replay-data-free: does not require access to partial data from the previous tasks or distributions. • denotes that some methods of this category don't satisfy the property. ✘ denotes that this drawback is relatively minor in real applications.

2024b,a).

Although many traditional CL methods have been proposed, they are not specifically designed for Modality-incremental Continual Learning (MCL). In this work, we propose a simple MCL paradigm tailored for MLLMs. Table 1 summarizes the characteristics of different CL categories and our proposed method.

2.3 Continual Learning for MLLMs

Aside from traditional continual learning, there are methods tailored for MLLMs (Srivastava et al., 2024; Zeng et al., 2024; Maharana et al., 2024; Gao et al., 2024; Cao et al., 2024; Yu et al., 2024a). However, most of these works are specific to vision-language-only MLLMs, and incompatible with MCL where MLLMs can extend to arbitrarily more modalities. To the best of our knowledge, PathWeave (Yu et al., 2024a) is the most relevant work to MCL, and is so far the only work on MCL for MLLMs. It is an architecture-based method that uses an adapter-in-adapter mechanism to alleviate forgetting in previous modalities.

In this work, we dive into the causes of degradation in MCL, uncovering that it suffers not only from forgetting but also from misalignment. To this end, we propose our two-stage MCL paradigm to address both forgetting and misalignment.

3 Dual Causes of Degradation in MCL

3.1 Preliminary

We define the Modality-incremental Continual Learning (MCL) problem as follows. Given a sequence of m modalities $\{M_1, M_2, \ldots, M_m\}$ and their corresponding datasets $\{D_1, D_2, \ldots, D_m\}$, MCL sequentially learns on each modality M_i to obtain the model θ_i. We denote the MLLM as $\theta = \{\theta^{Enc}, \theta^{Conn}, \theta^{LLM}\}$, where $\theta^{Enc}, \theta^{Conn}, \theta^{LLM}$ denote the modality encoders, the connectors, the LLM backbone, respectively. Notably, the θ^{Enc} and θ^{Conn} are modality-specific components and the θ^{LLM} is a modality-agnostic component. Further, we denote the feature distribution of the outputs of each modality connector θ_i^{Conn} as ϕ_i.

3.2 Forgetting and Misalignment in MCL

MCL is a special scenario of CL, where models incrementally learn on new modalities. However, MCL faces a more severe problem. During continual learning, models would suffer from performance degradation in previously learned domains, tasks, or modalities. In traditional CL, degradation comes from the forgetting of previously learned knowledge. However, in MCL, it comes from two significant aspects: forgetting and misalignment.

Forgetting: the modality-agnostic θ^{LLM} forgets the knowledge of old modalities. It is associated with various factors, including representation drift (Caccia et al.), gradient interference (Wang et al., 2023), learning dynamics (Ren and Sutherland, 2024), distribution shift, etc.

Misalignment: the modality-agnostic θ^{LLM} is misaligned with the modality-specific θ^{Enc}. When incrementally learning on a new modality M_i, the θ^{LLM} is updated along with the modality connector of M_i, while other old connectors are kept frozen. Therefore, θ^{LLM} adapts to the new feature distribution ϕ_i and drifts away from the original multimodal distribution $\phi_1 \cup \phi_2 \cup \cdots \cup \phi_{i-1}$[1]. Hence, there exists a misalignment in feature mapping between the old modality encoders and the θ^{LLM}, leading to the breakdown of the "encoder-connector-LLM" collaboration chains for old modalities. The illustration of misalignment is

[1]For our proposed method, the θ^{LLM} also drifts away from ϕ_i due to the use of model merging.

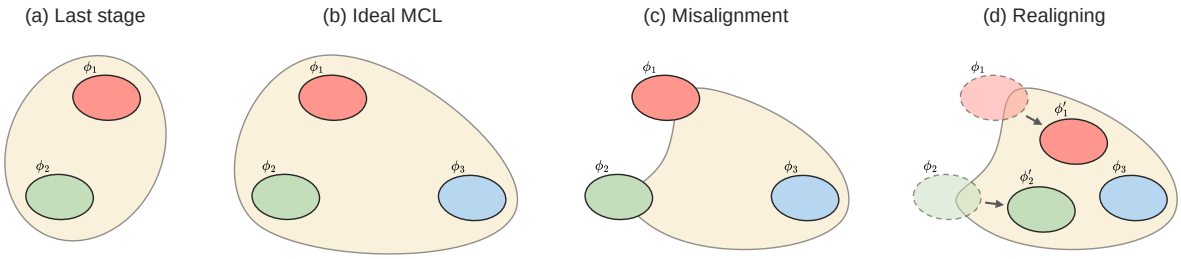

Figure 1: Illustration of misalignment and the mechanism of our proposed realigning. ϕ_i is the feature distribution of the i-th modality. Regions in yellow represent the LLM's expected distribution of the connector's output. (a) and (b) are the states of the last learning stage and the ideal MCL after learning on a new modality. (c) shows the actual misalignment after learning on a new modality. (d) demonstrates the mechanism of our proposed realigning.

sketched in Figure 1. Experiments in Section 6.1 also provide empirical evidence of the existence of misalignment.

Above, we analyzed the dual causes of performance degradation in MCL. Due to the existence of misalignment, MCL problem requires special treatments in contrast to traditional CL problems.

4 Method

To tackle the dual causes of performance degradation in MCL, we propose a two-stage MCL paradigm called "MErge then ReAlign" (MERA). In each stage of MCL, MERA executes the following two stages: merging and realigning, to address forgetting and misalignment respectively.

4.1 Stage 1: Merging

Model merging is efficient in integrating the previously learned knowledge into the fine-tuned models. Moreover, Yadav et al. (2024b) finds that model merging is more effective with larger models. Therefore, applying model merging to large-scale models such as MLLMs is inherently beneficial. Inspired by these, we introduce model merging to our MCL framework to mitigate forgetting.

The first step before model merging is to perform the standard MLLM training, which often encompasses a pre-training and a fine-tuning phase. After the standard training step, we get the vanilla model $\theta_{i,vanilla}$ that inevitably suffers from forgetting the knowledge of previous modalities.

The second step is to perform model merging to mitigate forgetting. In this work, we only focus on the simplest model merging method, i.e., weight averaging, *aiming only to provide a basic framework*. To adapt weight averaging to MCL, we associate its merging coefficients with the progress of MCL stages. At the i-th training stage, the merged model is calculated by:

$$\theta_{i,merged} = \frac{i-1}{i}\theta_{i-1} + \frac{1}{i}\theta_{i,vanilla}$$

Notably, we only merge the modality-agnostic component θ^{LLM} and ensemble the modality-specific components θ^{Enc} and θ^{Conn}. After merging, we obtain the $\theta_{i,merged}$, whose knowledge of previous modalities is enhanced.

4.2 Stage 2: Realigning

To address the misalignment issue, we propose a lightweight realigning stage to update the multimodal distribution $\phi_1 \cup \phi_2 \cup \cdots \cup \phi_i$ to realign the "encoder-connector-LLM" chains for all the modalities.

The realigning stage simply leverages a small replay dataset[2] $R_i \leftarrow$ sample $r\%$ data from $\{D_1, D_2, \ldots, D_i\}$ to further fine-tune all the connectors of $\theta_{i,merged}$. This realigning process is formulated as:

$$\min_{\theta^{Conn}_{i,merged}} \mathbb{E}_{x\sim R_i}\mathcal{L}(\theta_{i,merged}, x)$$

where the $\mathcal{L}$ is the auto-regressive loss, unchanged from the original MLLM training loss. By fine-tuning the lightweight θ^{Conn} with only a small replay dataset, it efficiently realigns the θ^{Enc} with θ^{LLM}. After realigning, we can obtain the final model θ_i.

Differences between realigning and replay-based CL methods. The realigning stage resembles replay-based CL methods (Scialom et al., 2022; Wang et al., 2024b) in form as they both leverage replay data, however, they are essentially different. First, replay methods train on the joint

[2]It is different from replay data in replay-based continual learning methods, where their R_i is sampled from $\{D_1, D_2, \ldots, D_{i-1}\}$.

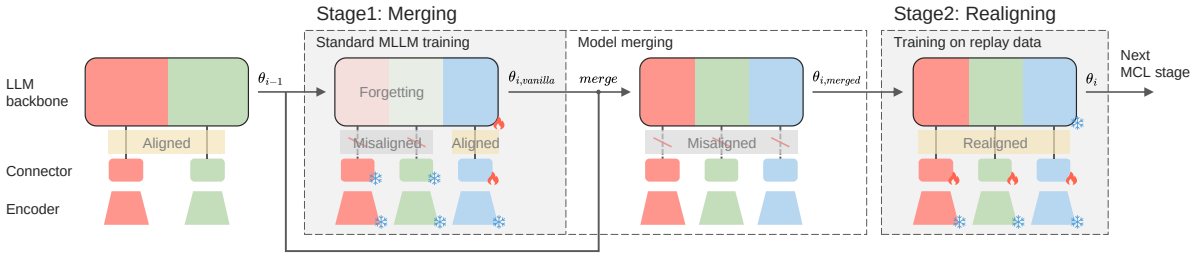

Figure 2: Pipeline of the proposed MERA. The procedures in gray boxes involve training. ❄ and 🔥 represent the frozen and trainable modules, respectively.

dataset of D_i and R_i, while our realigning stage trains solely on R_i. Second, replay methods update both the θ^{LLM} and θ^{Conn}, while the realigning stage only efficiently updates the θ^{Conn} to prevent the knowledge inside θ^{LLM} from being overwritten again.

4.3 Overall Pipeline

The overall pipeline of MERA is illustrated in Figure 2. In each stage of MCL, MERA goes through two stages. The first stage is merging, where we fine-tune θ_{i-1} on the incoming modality and merge the fine-tuned model $\theta_{i-1,vanilla}$ with the historical model θ_{i-1} in order to alleviate forgetting. The second stage is realigning, where we leverage a small set of replay data R_i to efficiently fine-tune the lightweight modality connectors θ^{Conn} to realign the θ^{Enc} with θ^{LLM}.

5 Experiments

5.1 Experimental Setup

We build our MCL experiments on four modalities: image, video, audio, and point cloud, with two different training orders. Based on the prevalence of different modalities, we determine the two orders as follows. **Sequential Order**: image → video → audio → point cloud. **Reverse Order**: point cloud → audio → video → image. On top of this, the adopted datasets, metrics, models, and baselines are detailed as follows.

Datasets. For each modality M_j, we leverage a dataset of Captioning (Cap) task and a dataset of Question Answering (QA) task to form the joint dataset $D_j = \{D_{j,Cap}, D_{j,QA}\}$. The Cap and QA datasets for each modality are listed respectively. For image modality, we use MSCOCO-2014 (Lin et al., 2014) and OK-VQA (Marino et al., 2019). For video modality, we use MSVD (Chen and Dolan, 2011) and MSVD-QA (Xu et al., 2017). For audio modality, we use AudioCaps (Kim et al., 2019) and Clotho-AQA (Lipping et al., 2022). For point cloud modality, we use a subset of Cap3D (Luo et al., 2024) and a subset of Cap3D-QA (Panagopoulou et al., 2023). More details of these datasets are in Appendix B.

Evaluation Metrics. First, we leverage Relative Gain (Scialom et al., 2022; Wang et al., 2024b) as a normalized metric across different tasks. We naively train (without using any continual learning methods) expert MLLMs individually on each single modality M_j and test with their respective holdout data, taking their scores on the k-th dataset $D_{j,k}$ as upper bound $S^{sup}_{j,k}$. In the incremental stage i, the Relative Gain of modality M_i with its dataset $D_j = \{D_{j,k}\}_{k=1}^{K}$ is calculated by:

$$\text{Relative Gain}^i_j = \frac{1}{K}\sum_{k=1}^{K}\frac{S^i_{j,k}}{S^{sup}_{j,k}}$$

where $S^i_{j,k}$ is the score on the test set of $D_{j,k}$ in the stage i. Here, we utilize CIDEr score (Vedantam et al., 2015) and prediction accuracy (Acc) for Cap and QA tasks respectively to calculate $S^{sup}_{j,k}$ and $S^i_{j,k}$. To evaluate the performance degradation of the previously learned modalities, we calculate the Backward Relative Gain in the stage i as:

$$\text{Bw Relative Gain}^i = \frac{1}{i-1}\sum_{j=1}^{i-1}\text{Relative Gain}^i_j$$

To measure the plasticity, i.e., the ability to adapt to new knowledge, we calculate the Forward Relative Gain in the stage i as:

$$\text{Fw Relative Gain}^i = \text{Relative Gain}^i_i$$

Model and Training Details. We leverage the mainstream MLLM architecture, i.e., LLaVA-like architecture with the Llama-3-8B-Instruct (Dubey et al., 2024) as its LLM backbone. The selections of modality encoders and connectors are detailed

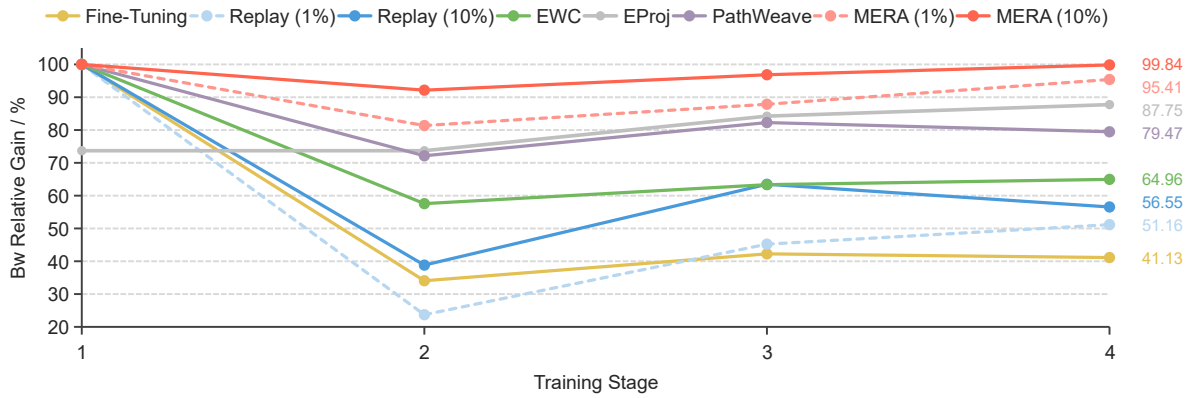

Figure 3: Progressive Backward Relative Gain in modality-incremental continual learning. For each stage i, we plot the average score of the corresponding Backward Relative Gain with two different training orders. We set Backward Relative Gain to 100% for the 1st stage, denoting the initial performance without degradation. Exceptionally, the initial Backward Relative Gain of EProj is not 100% since it only tunes the modality-specific components, causing an initial performance degradation.

in Appendix C.1. Trainings that involve updating the LLM backbone utilize LoRA (Hu et al., 2022) for parameter-efficient fine-tuning. The training process in our merging stage is the same as the standard MLLM training, i.e., in the first step, only the connector is updated with Cap datasets, then in the second step, the connector and the LLM backbone are updated with all the task-related datasets (the combination of Cap and QA datasets in our case). In our realigning stage, the replay datasets are randomly sampled from the joint datasets of Cap and QA tasks. For each training process, the hyperparameters are listed in Appendix C.1.

Baselines. In our experiments, we compare our MERA with non-CL fine-tuning, as well as the representative CL and MCL methods: **Fine-Tuning**: directly train MLLMs sequentially on each modality without applying any CL method. **Replay**: the vanilla replay-based CL method. During training on a new task, the model is updated with both samples from the current task and a set of randomly sampled replay data from previous tasks. **EWC** (Kirkpatrick et al., 2017): the most representative regularization-based CL method. EWC mitigates forgetting by restricting the updates of important weights during training on new tasks. It uses the Fisher information matrix to measure the importance of each weight. **EProj** (He et al., 2023): tuning only the modality-specific components to prevent forgetting. **PathWeave** (Yu et al., 2024a): an architecture-based CL method, also the state-of-the-art MCL method for MLLMs. PathWeave uses an adapter-in-adapter mechanism to memorize and extract knowledge from historical modalities to enhance the learning of the current modality. PathWeave is originally built on X-InstructBLIP (Panagopoulou et al., 2023). For a fair comparison, we implement PathWeave for our adopted MLLM architecture. Implementation details of each baseline method are in Appendix C.3.

5.2 Main Results

We conduct experiments under our MCL setting with both sequential and reverse orders. For Replay and our MERA, results using $r\%$, $r = \{1, 10\}$ replay data are reported, denoted by Replay ($r\%$) and MERA ($r\%$) respectively.

Evaluating Degradation. The progressive Backward Relative Gains averaged from different training orders are plotted in Figure 3. It is observed that our MERA demonstrates an impressive capability of mitigating performance degradation with consistent and promising Backward Relative Gains. When extending to all four modalities, MERA (10%) holds up to a 99.84% Backward Relative Gain, indicating that MERA can achieve nearly lossless MCL performance, with at least 12.09% absolute improvements of Backward Relative Gain compared with other baselines. Notably, when only leveraging 1% replay data, MERA (1%) can still achieve at least 7.66% absolute improvements over other baselines. Further, we calculated the mean and standard deviation of Backward Relative Gains in all training stages for each method,

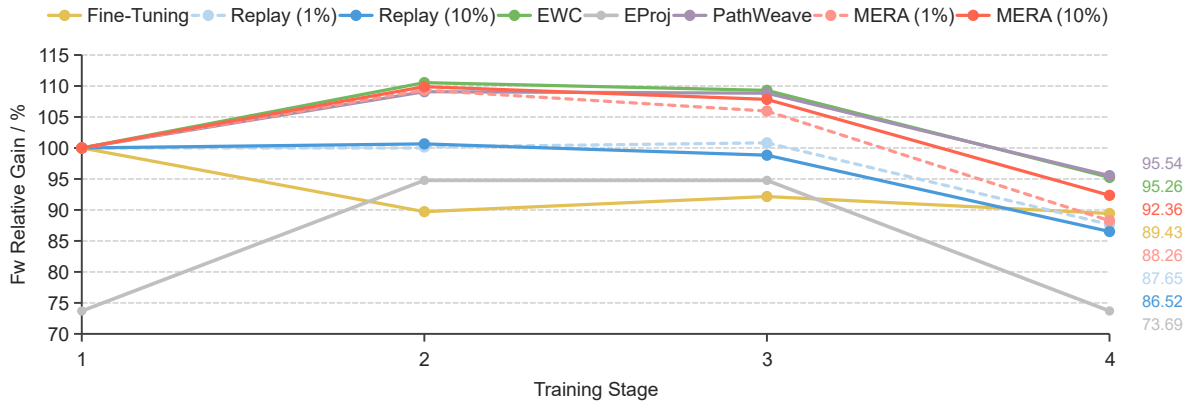

Figure 4: Progressive Forward Relative Gain in modality-incremental continual learning. For each stage i, we plot the average score of the corresponding Forward Relative Gain with two different training orders. We set Forward Relative Gain to 100% for the 1st stage, denoting the initial lossless plasticity. Exceptionally, the initial Forward Relative Gain of EProj is not 100% since it only tunes the modality-specific components, causing an initial loss of plasticity.

Method	Sequential		Reverse	
	Mean	Std	Mean	Std
Fine-Tuning	59.76	27.23	48.96	35.70
Replay (1%)	66.09	25.86	43.95	39.03
Replay (10%)	77.52	16.32	51.90	36.34
EWC	74.93	17.14	68.01	21.54
EProj	89.61	**2.63**	70.07	<u>11.86</u>
PathWeave	86.85	12.17	80.09	13.31
MERA (1%)	<u>97.90</u>	6.02	<u>84.42</u>	12.93
MERA (10%)	**101.00**	<u>3.90</u>	**93.42**	**6.25**

Table 2: The mean and standard deviation of Backward Relative Gains in all the training stages. Results are reported on different training orders. The best results are in **bold**, while the second-best are <u>underlined</u>.

Method	Sequential	Reverse
Fine-Tuning	38.50	36.14
Replay (1%)	30.61	33.02
Replay (10%)	56.68	32.50
EWC	25.79	24.94
EProj	55.61	<u>55.61</u>
PathWeave	36.32	46.39
MERA (1%)	<u>72.13</u>	**61.22**
MERA (10%)	**72.70**	54.04

Table 3: Accuracies on MCUB-4 benchmark. Results are reported on different training orders and are measured after continually learning on all four modalities.

in different training orders. Table 2 shows that our MERA (10%) achieves the highest mean in both training orders, indicating its superior performance. It also achieves the lowest and second-lowest standard deviation in sequential and reverse orders respectively, indicating its high stability. Notably, in sequential order, MERA (10%) performs even better than lossless MCL, with an over 100% average Backward Relative Gain, also at least 11.39% absolute improvements of average Backward Relative Gain over other baselines. In reverse order, MERA (10%) also achieves at least 13.33% absolute improvements. When with only 1% replay data, MERA (1%) still achieves at least 8.29% and 4.33% absolute improvements in sequential and reverse orders respectively.

Evaluating Degradation on Complex Multimodal Tasks. To evaluate MERA's performance in more complex multimodal tasks that involve multiple modalities at a time, we further conduct experiments on the MCUB-4 (Chen et al., 2024) benchmark that requires the model to simultaneously infer on all four modalities, i.e., image, video, audio, and pointcloud. Table 3 shows the accuracies of each method on MCUB-4 benchmark. It is observed that our proposed MERA also significantly excels in more complex multimodal tasks that involve multiple modalities at a time.

Evaluating Plasticity. Aside from alleviating performance degradation, the capability to adapt to new knowledge, i.e., plasticity, is also an important aspect. We use the Forward Relative Gain as the metric. The progressive Forward Relative Gains

Method	Sequential		Reverse	
	Mean	Std	Mean	Std
Fine-Tuning	59.76	27.23	48.96	35.70
+Merging	90.29	7.42	70.00	30.87
+Realigning	87.92	12.41	71.90	24.52
MERA	**101.00**	**3.90**	**93.42**	**6.25**

Table 4: Ablation study of different components in MERA. The realigning stage uses 10% replay data.

averaged from different training orders are plotted in Figure 4. It is observed that the most elastic CL methods are EWC and PathWeave, while our MERA (10%) demonstrates comparable plasticity.

5.3 Ablation Study

We conduct ablation studies to investigate the effectiveness of each stage of MERA. Results are shown in Table 4. Firstly, from Table 4 and Table 2, it is observed that the realigning stage that addresses the misalignment issue can already beat many other baselines designed to tackle the forgetting issue, achieving the second-best average Backward Relative Gain among baselines in both sequential and reverse orders. Secondly, combining both merging and realigning stages, MERA further narrows the gap between the incrementally learned models and the individually trained experts on each modality, even, surpassing the individually trained experts in sequential training order with over 100% Backward Relative Gain.

6 Discussions

6.1 Is Misalignment Common in MCL?

Since the realigning stage achieves great success on top of our proposed merging stage, we further ask another question: does realigning benefit other CL methods, or in other words, *is misalignment a common phenomenon in MCL*? To examine this, we perform the realigning stage at the end of every training stage for different CL or non-CL methods to observe whether there are performance improvements[3]. Table 5 shows that the additional realigning stage brings substantial performance improvements and increased stability for different CL or non-CL methods. Based on this observation, we can conclude that **misalignment is a common phenomenon in MCL**, and can be alleviated by our proposed realigning stage.

[3]We do not examine this for EProj. EProj is self-evidently misalignment-free, since it freezes the LLM backbone.

Method	Sequential		Reverse	
	Mean	Std	Mean	Std
Fine-Tuning	59.76	27.23	48.96	35.70
+Realigning	+28.16	-14.83	+22.93	-11.17
Replay (1%)	66.09	25.86	43.95	39.03
+Realigning	+20.64	-9.75	+24.35	-8.36
Replay (10%)	77.52	16.32	51.90	36.34
+Realigning	+14.21	-6.67	+23.71	-11.07
EWC	74.93	17.14	68.01	21.54
+Realigning	+19.54	-6.87	+23.02	-13.25
PathWeave	86.85	12.17	80.09	13.31
+Realigning	+6.22	-0.28	+2.40	-1.53
Merging	90.29	7.42	70.00	30.87
+Realigning	+10.71	-3.52	+23.42	-24.63

Table 5: Applying realigning to different CL methods can further improve their Backward Relative Gain and stability. The realigning stage uses 10% replay data. Improvements are colored in green.

6.2 Positive Backward Transfer and Positive Forward Transfer

From Figure 3 and its raw data shown in Appendix D, we observe a faint phenomenon of Positive Backward Transfer (Lin et al., 2022) that learning new knowledge improves the performance on previously learned tasks. For most methods, the Backward Relative Gain comes to a low level when incrementally learning the second modality, but starts to stabilize and even increase when incrementally learning more modalities.

From Figure 4, we observe a strong phenomenon of Positive Forward Transfer (Ke et al., 2021) that the knowledge acquired from earlier tasks improves the learning efficiency of new tasks. The Positive Forward Transfer exists before the 4-th incremental stage, when employing EWC, PathWeave, and MERA. This phenomenon is also reported in other MCL literature (Yu et al., 2024a). In contrast to Positive Forward Transfer, there is a gradual loss of plasticity (Dohare et al., 2024, 2023) as the model attempts to retain more knowledge. This explains the decreases in Forward Relative Gain across different CL methods in the 4-th stage, as the loss of plasticity comes to a dominant position.

7 Future Work

This work is one of the early attempts of MCL for MLLMs, focusing on the dual causes of its degradation. In addition to this, observation of Positive Backward Transfer and Positive Forward Transfer in Section 6.2 may imply the complex cross-modal interaction in multimodal learning, urging for future research on the mechanisms of modality interaction in the context of MCL.

8 Conclusion

In this paper, we first revisit MCL and uncover the dual causes of its degradation, i.e., forgetting and misalignment. Next, to address both forgetting and misalignment, we propose MERA, a simple yet effective MCL paradigm. Extensive experiments demonstrate that MERA significantly outperforms the state-of-the-art methods and even achieves nearly lossless MCL performance. Our findings underscore the misalignment issue in MCL. More broadly, our work showcases how to adjust different components of MLLMs during continual learning. Further, we observe signs of complex cross-modal interaction in MCL, providing a direction for future work.

9 Limitations

This work is restricted in the following aspects. First, our experiments are limited to four commonly used modalities due to the lack of resources for other less-studied modalities. Second, we limit this work to LLaVA-like architecture as it covers the majority of MLLMs. Third, this work is limited to any-to-text MLLMs while there is now a trend of exploring any-to-any MLLMs. However, the main idea of MERA is generic to them. Fourth, our work only explores the simplest model merging method in the context of MCL, aiming to provide a universal framework, leaving other model merging methods for MCL for future work.

10 Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62372139), the National Natural Science Foundation of China (2024A1515030024), Research Projects of Shenzhen (JCYJ20220818102414030), and Key Laboratory of Guangdong Province (2022B1212010005).

References

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*.

Meng Cao, Yuyang Liu, Yingfei Liu, Tiancai Wang, Jiahua Dong, Henghui Ding, Xiangyu Zhang, Ian Reid, and Xiaodan Liang. 2024. Continual llava: Continual instruction tuning in large vision-language models. *arXiv preprint arXiv:2411.02564*.

Chi Chen, Yiyang Du, Zheng Fang, Ziyue Wang, Fuwen Luo, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Maosong Sun, and Yang Liu. 2024. Model composition for multimodal large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. 2023a. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv preprint arXiv:2305.04160*.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. 2023b. Beats: audio pre-training with acoustic tokenizers. In *International Conference on Machine Learning*.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. 2024. Loss of plasticity in deep continual learning. *Nature*.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. 2023. Maintaining plasticity in deep continual learning. *arXiv preprint arXiv:2306.13812*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Shaoqi Dong, Xiong Wang, Di Yin, Long Ma, et al. 2024. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*.

Zijian Gao, Xingxing Zhang, Kele Xu, Xinjun Mao, and Huaimin Wang. 2024. Stabilizing zero-shot prediction: A novel antidote to forgetting in continual vision-language tasks. *Advances in Neural Information Processing Systems*.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *International Conference on Learning Representations*.

Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. 2024. Onellm: One framework to align all modalities with language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. 2023. Continual instruction tuning for large multimodal models. *arXiv preprint arXiv:2311.16206*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ferenc Huszár. 2017. On quadratic penalties in elastic weight consolidation. *arXiv preprint arXiv:1712.03847*.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. 2019. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*.

Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. 2024. Uni-moe: Scaling unified multimodal llms with mixture of experts. *arXiv preprint arXiv:2405.11273*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024a. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024b. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. 2022. Beyond not-forgetting: Continual learning with backward knowledge transfer. *Advances in Neural Information Processing Systems*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*.

Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. 2022. Clotho-aqa: A crowdsourced dataset for audio question answering. In *2022 30th European Signal Processing Conference*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in Neural Information Processing Systems*.

Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. 2024. Scalable 3d captioning with pretrained models. *Advances in Neural Information Processing Systems*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Adyasha Maharana, Jaehong Yoon, Tianlong Chen, and Mohit Bansal. 2024. Adapt-∞: Scalable lifelong multimodal instruction tuning via dynamic data selection. *arXiv preprint arXiv:2410.10636*.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. 2024. Magmax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision*.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*.

Openai. 2021. huggingface/openai/clip-vit-large-patch14.

Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. 2023. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *arXiv preprint arXiv:2311.18799*.

Jun Rao, Zepeng Lin, Xuebo Liu, Xiaopeng Ke, Lian Lian, Dong Jin, Shengjun Cheng, Jun Yu, and Min Zhang. 2025. APT: Improving specialist LLM performance with weakness case acquisition and iterative preference training. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. CommonIT: Commonality-aware instruction tuning for large language models via data partitions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Jun Rao, Xv Meng, Liang Ding, Shuhan Qi, Xuebo Liu, Min Zhang, and Dacheng Tao. 2023. Parameter-efficient and student-friendly knowledge distillation. *IEEE Transactions on Multimedia*.

Yi Ren and Danica J Sutherland. 2024. Learning dynamics of llm finetuning. *arXiv preprint arXiv:2407.10490*.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in Neural Information Processing Systems*.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*.

Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. 2023. Multimodal neurons in pretrained text-only transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Qi Shuhan, Cao Zhengying, Rao Jun, Wang Lei, Xiao Jing, and Wang Xuan. 2023. What is the limitation of multimodal llms? a deeper look into multimodal llms through prompt probing. *Information Processing & Management*.

Tejas Srinivasan, Furong Jia, Mohammad Rostami, and Jesse Thomason. 2023. I2i: Initializing adapters with improvised knowledge. In *Conference on lifelong learning agents*.

Shikhar Srivastava, Md Yousuf Harun, Robik Shrestha, and Christopher Kanan. 2024. Improving multimodal large language models using continual learning. *arXiv preprint arXiv:2410.19925*.

Gido M Van de Ven and Andreas S Tolias. 2019. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gaurav Verma, Minje Choi, Kartik Sharma, Jamelle Watson-Daniels, Sejoon Oh, and Srijan Kumar. 2024. Cross-modal projection in multimodal llms doesn't really project visual attributes to textual space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024a. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024b. Inscl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

wangleihitcs. 2019. github/wangleihitcs/captionmetrics.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2022. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2024. Next-gpt: Any-to-any multimodal llm. In *Forty-first International Conference on Machine Learning*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. 2024. Lm-cocktail: Resilient tuning of language models via model merging. In *Findings of the Association for Computational Linguistics ACL 2024*.

Yi Xin, Junlong Du, Qiang Wang, Zhiwen Lin, and Ke Yan. 2024. Vmt-adapter: Parameter-efficient transfer learning for multi-task dense scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. 2024. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024a. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*.

Prateek Yadav, Tu Vu, Jonathan Lai, Alexandra Chronopoulou, Manaal Faruqui, Mohit Bansal, and Tsendsuren Munkhdalai. 2024b. What matters for model merging at scale? *arXiv preprint arXiv:2410.03617*.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. 2024a. Representation surgery for multi-task model merging. *International Conference on Machine Learning*.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024b. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*.

Jiazuo Yu, Haomiao Xiong, Lu Zhang, Haiwen Diao, Yunzhi Zhuge, Lanqing Hong, Dong Wang, Huchuan Lu, You He, and Long Chen. 2024a. Llms can evolve continually on modality for x-modal reasoning. *arXiv preprint arXiv:2410.20178*.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. 2024b. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024c. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. 2024. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *International Conference on Learning Representations*.

Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. 2024. Modalprompt: Dual-modality guided prompt for continual learning of large multimodal models. *arXiv preprint arXiv:2410.05849*.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, Hang Yan, Jie Fu, Tao Gui, Tianxiang Sun, Yu-Gang Jiang, and Xipeng Qiu. 2024. AnyGPT: Unified multimodal LLM with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Didi Zhu, Zhongyi Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Chao Wu, and Kun Kuang. 2024. Model tailor: mitigating catastrophic forgetting in multi-modal large language models. In *International Conference on Machine Learning*.

Appendix

A Further Analysis and Discussions

A.1 Deeper Discussions on the Distinction Between Forgetting and Misalignment

Theoretically, forgetting is associated with various factors such as representation drift (Caccia et al.), gradient interference (Wang et al., 2023), learning dynamics (Ren and Sutherland, 2024), distribution shift (which causes misalignment), making it a comprehensive issue. For example, Schwettmann et al. (2023) finds that there exist multimodal neurons in the LLM backbone, each corresponds to certain domain concepts, and when breaking some of these neurons, the model's prediction is largely affected. However, addressing the misalignment issue by our proposed realigning will possibly not recover the model's performance since the connectors do not encode domain-specific concepts (Verma et al., 2024). This example showcases that the forgetting issue is not limited to misalignment. **Empirically**, our experiments in Table 5 have shown that, a) addressing only the misalignment issue, i.e., fine-tuning with realigning, does not outperform some methods that address the overall forgetting, e.g., PathWeave/merging without realigning, b) methods that address the overall forgetting significantly complement the realigning alone (fine-tuning with realigning). In terms of methodology, forgetting is a comprehensive issue that can be tackled mainly by heuristic methods such as EWC or replay, while misalignment is caused solely by distribution shift that can be directly addressed by realigning the LLM and modality-specific components. **In terms of methodology**, forgetting is a comprehensive issue that can be tackled mainly by heuristic methods such as EWC or replay, while misalignment is caused solely by distribution shift that can be directly addressed by realigning the LLM and modality-specific components.

A.2 Misalignment Also Exists In Replay-Based Methods

Theoretically, a potential exception to misalignment is when replay-based methods are applied, since the θ^{LLM} is trained on the joint distribution of all $\{M_1, M_2, \ldots, M_i\}$ modalities. However, Table 5 suggests that replay-based methods still suffer from misalignment and can be compensated for by our proposed realigning stage. We conjecture that it is due to the imbalanced distribution of its training data. Replay-based methods train the model on the joint dataset of its replay data and D_i, where the scale of replay data from each previous modality is insignificant to the scale of D_i. Therefore, it still suffers from a certain degree of distribution shift during its training process.

A.3 Easily Adapt MERA to Other MLLM Architectures

Although we limit our work to LLaVA-like architecture, our method can be easily adapted to other MLLM architectures. We present several naive ways to adapt MERA to some other MLLM architectures.

- For connector-free MLLMs, the last few layers of their encoders can be treated as connectors so that our method can be directly applied.
- For MLLMs that use a uni-connector for all modalities, we can treat them as the connector-free MLLMs.
- For encoder-free MLLMs, we can realign the raw multimodal input distributions instead of multimodal feature distributions with the LLM backbone by fine-tuning the embedding layers.

A.4 Efficiency Comparisons

We compare the efficiency of different baselines and our MERA, as shown in Table 6. It is observed that our MERA can achieve optimal results except that MERA (1%) and MERA (10%) introduce 2% and 15% extra training time-consuming respectively. However, we believe its trade-off between training time-consuming and performance is worthwhile, considering the impressive performance of MERA. In our experiments, EWC and PathWeave introduce marginal extra training memory overhead, as we employ parameter-efficient fine-tuning. However, for larger LoRA ranks or even full model fine-tuning, their extra training memory consumptions would be substantial, as they necessitate storing additional parameters whose sizes increase linearly with the trainable parameters.

A.5 Robustness to the Quality of Replay Data

Since the realigning stage of MERA relies on a small replay dataset, it is beneficial to understand how robust MERA is to the quality of replay data. To evaluate its robustness, we manually corrupt

Method	Training		Inference	
	Peak Memory	Time-Consuming	Peak Memory	Lantency per Token
Fine-Tuning	37.43 GB	53 h	17.71 GB	34 ms
Replay (1%)	37.43 GB	54 h	17.71 GB	34 ms
Replay (10%)	37.43 GB	59 h	17.71 GB	34 ms
EWC	38.73 GB	54 h	17.71 GB	34 ms
PathWeave	40.08 GB	81 h	20.32 GB	111 ms
MERA (1%)	37.43 GB	54 h	17.71 GB	34 ms
MERA (10%)	37.43 GB	61 h	17.71 GB	34 ms

Table 6: Training and inference overheads of different methods. The peak memories during training and inference are measured with batch sizes of 4 and 1 respectively. The time-consuming refers to the total GPU hours for continually learning the four modalities. All metrics are measured on a single NVIDIA RTX A6000 48G. The non-optimal results are colored in red.

Method	Sequential		Reverse	
	Mean	Std	Mean	Std
MERA	**97.90**	6.02	**84.42**	**12.93**
MERA w/ 10% noise	96.87	**4.97**	83.56	13.48
MERA w/ 50% noise	94.10	5.67	79.79	15.54

Table 7: Evaluations of MERA's robustness to noisy replay data. The realigning stage uses 10% replay data.

	#Training Set	#Test Set	License
MSCOCO-2014*	82K	1K	CC-BY 4.0
OK-VQA	26K	1K	CC-BY 4.0
MSVD	48K	670	-
MSVD-QA	30K	1K	-
AudioCaps	44K	1K	-
Clotho-AQA*	15K	1K	MIT License
Cap3D*	50K	1K	ODC-BY 1.0
Cap3D-QA*	30K	1K	-

Table 8: Statistics of the datasets. Datasets marked with * are filtered from their original ones.

$p\%$ samples in the replay dataset by mispairing their text-modality pair, and test MERA's performance. Results in Table 7 suggest that 10% noisy samples does not significantly corrupt MERA's performance, and it still shows decent performance even under the extreme condition of 50% noisy samples. This suggests that our MERA is robust to low-quality replay data.

A.6 Why is the Performance Degradation More Severe in Reverse Training Order

In Table 2, it is observed that the performance degradation is more severe in reverse training order than in sequential training order. This phenomenon occurs across different methods. We conjecture that training in sequential order is a sort of curriculum learning, while the reverse order corresponds to reversed curriculum learning. In our experiments, the sequential order is determined based on the prevalence of different modalities, and more prevalent modalities might be easier to learn. For example, the image modality is easier to understand than video, and the point cloud might be the most difficult modality to learn. Wang et al. (2024b) also observed that the performance degradation is less severe when continually learning in a curriculum learning order than in a reversed curriculum learning order. We conjecture that learning easy knowledge leads to the forgetting of harder ones, but learning hard knowledge might even consolidate the easier ones.

B Dataset Details

Table 8 details the statistics of each dataset. Some datasets are filtered from their original ones:

- MSCOCO-2014 (Lin et al., 2014): Each image has multiple captions, we only use its first caption to form the training set.
- Clotho-AQA (Lipping et al., 2022): Each sample is annotated with a confidence level, we only use the samples whose confidence levels are "yes" to form the training set and test set.
- Cap3D (Luo et al., 2024): Since the original

Hyperparameters	Pre-Training	Fine-Tuning	Realigning
Trainable Components	Connectors	LLM and Connectors	Connectors
Batch Size	128	16	16
Learning Rate of Connectors	1e-3	2e-5	2e-5
Learning Rate of LLM	-	2e-4	-
Learning Rate Schedule		Cosine Decay	
Warmup Ratio		0.03	
Epoch		1	

Table 9: Hyperparameters for each training process. Pre-Training and Fine-Tuning refer to the two stages of the standard MLLM training process.

dataset is huge in scale, we filter out the samples whose caption is longer than 100 letters. Then, we randomly sample a 50K subset as the training set.

- Cap3D-QA (Panagopoulou et al., 2023): Since the original dataset is huge in scale, we randomly sample a 30K subset as the training set.

For each dataset, we use a randomly sampled 1K subset of its holdout test set as the final test set, except for the MSVD (Chen and Dolan, 2011), since the size of its original test set is less than 1K.

C Experimental Details

C.1 Implementation Details

We build our experimental codebase on top of LLaVA (Liu et al., 2024b,a) and NExT-GPT (Wu et al., 2024). We detail the modality-specific components of each modality as follows:

- Image: We use CLIP-ViT-L-336px (Openai, 2021) as the pre-trained image encoder, a randomly initialized MLP as the connector.
- Video: We use CLIP-ViT-L-336px (Openai, 2021) as the pre-trained video encoder, a randomly initialized MLP as the connector. We uniformly sample 4 frames from a video as input frames. Then each frame is encoded by the video encoder separately. The output feature frames are downsampled by 2x using bilinear pooling before sending into the connector to improve efficiency.
- Audio: We use $\text{BEATs}_{\text{iter3+}}$(AS2M) (Chen et al., 2023b) as the pre-trained audio encoder, a Q-Former (Li et al., 2023) initialized from the pre-trained bert-base-uncased (Kenton and Toutanova, 2019) as the connector. The number of query tokens is set to 32.
- Point Cloud: We use Point-BERT-v1.2 (Xu et al., 2024) as the pre-trained point cloud encoder, a randomly initialized MLP as the connector.

We set the hyperparameters mainly following previous works (Liu et al., 2024b,a), as listed in Table 9. For training that involves updating the LLM backbone, we utilize parameter-efficient fine-tuning with LoRA (Hu et al., 2022) applied across all linear modules within the LLM, setting the LoRA rank to 128 and the alpha parameter to 128. All the experiments are conducted on a single NVIDIA RTX A6000 48G with FP16.

C.2 Evaluation Details

For the calculation of CIDEr scores (Vedantam et al., 2015), we utilized an open-sourced library CaptionMetrics (wangleihitcs, 2019). For the calculation of prediction accuracy, we leverage a GPT-based open-ended QA evaluation with GPT-4o mini as the judge model. The GPT is prompted to judge whether the generated prediction semantically matches the ground truth answer. The prompt template is shown in Table 10. All the reported experimental results are from single runs.

C.3 Implementation of Baselines

The implementation details of each CL baseline are listed as follows:

- **Replay** leverage a small replay dataset $R_i \leftarrow$ randomly sample $r\%$ data from $\{D_1, D_2, \ldots, D_{i-1}\}$. When training on a new modality M_i, it trains on the joint dataset of R_i and D_i.

System Prompt:
You are an intelligent chatbot designed for evaluating the correctness of generative outputs for question-answer pairs. Your task is to compare the predicted answer with the correct answer and determine if they match meaningfully. Here's how you can accomplish the task: ##INSTRUCTIONS: - Focus on the meaningful match between the predicted answer and the correct answer. - Consider synonyms or paraphrases as valid matches. - Evaluate the correctness of the prediction compared to the answer.
User Prompt:
Please evaluate the following question-answer pair: Question: <question> Correct Answer: <answer> Predicted Answer: <prediction> Provide your evaluation only as a yes/no and score where the score is an integer value between 0 and 5, with 5 indicating the highest meaningful match. Please generate the response in the form of a Python dictionary string with keys 'pred' and 'score', where value of 'pred' is a string of 'yes' or 'no' and value of 'score' is in INTEGER, not STRING.DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string. For example, your response should look like this: {'pred': 'yes', 'score': 4.8}.

Table 10: Prompts to query the GPT for open-ended QA evaluation. The placeholders in red boxes are filled according to each evaluated sample.

- **EWC** firstly estimates the Fisher information matrix F_{i-1} of the last training stage $i-1$ as:

$$F_{i-1} = \mathbb{E}_{x \sim D_{i-1}} \nabla_{\theta_{i-1}} \mathcal{L}(\theta_{i-1}, x) \cdot \nabla_{\theta_{i-1}} \mathcal{L}(\theta_{i-1}, x)^T$$

where $\mathcal{L}(\theta_{i-1}, x)$ denotes the auto-regressive loss of model θ_{i-1} on data $x \sim D_{i-1}$, which is sampled from a 1% size random subset of D_{i-1}. Then the loss function $\mathcal{L}^*(\theta_i, x)$ of stage i is:

$$\mathcal{L}^*(\theta_i, x) = \mathcal{L}(\theta_i, x) + \sum_{j=0}^{i-1} \frac{\lambda}{2} F_j (\theta_i - \theta_{i-1})^2$$

where the hyperparameter λ is set to 1 as default. This implementation of $\mathcal{L}^*(\theta_i, x)$ is known as Online EWC (Huszár, 2017; Schwarz et al., 2018).

- **PathWeave** leverages Adapter-in-Adapter (AnA) modules. In our implementation, the AnA modules are injected into the LLM rather than the connector for a fair comparison. The rank of AnA is consistent with the LoRA rank of other baselines, which is 128. In their original settings, PathWeave removes the newly added modules when testing the former modalities. However, this results in the inability to perform cross-modality tasks, which are common in real applications. Therefore, we do not remove them for a fair comparison.

For each baseline, all the common parameters about training MLLM itself are the same and set as default in Table 9.

D Complete Raw Data

Table 11 and Table 12 show the raw data of Figure 3 and Figure 4.

Method		Image		Video		Audio		Point Cloud	
		MSCOCO	OK-VQA	MSVD	MSVD-QA	AudioCaps	Clotho-AQA	Cap3D	Cap3D-QA
Individually Trained Experts		100.76	0.358	138.39	0.460	60.14	0.658	99.93	0.568
Fine-Tuning	Stage 1	100.76	0.358	-	-	-	-	-	-
	Stage 2	54.52	0.172	130.22	0.555	-	-	-	-
	Stage 3	34.87	0.303	12.78	0.292	43.17	0.590	-	-
	Stage 4	58.63	0.201	29.55	0.350	8.28	0.094	84.40	0.524
Replay (1%)	Stage 1	100.76	0.358	-	-	-	-	-	-
	Stage 2	41.45	0.125	137.07	0.569	-	-	-	-
	Stage 3	65.79	0.276	30.74	0.312	55.21	0.675	-	-
	Stage 4	59.94	0.225	102.16	0.469	22.17	0.490	81.43	0.508
Replay (10%)	Stage 1	100.76	0.358	-	-	-	-	-	-
	Stage 2	50.65	0.266	137.67	0.584	-	-	-	-
	Stage 3	83.32	0.318	33.87	0.381	44.82	0.651	-	-
	Stage 4	67.42	0.259	133.43	0.520	24.13	0.525	73.19	0.515
EWC	Stage 1	100.76	0.358	-	-	-	-	-	-
	Stage 2	64.84	0.208	155.09	0.595	-	-	-	-
	Stage 3	44.22	0.211	73.14	0.569	59.86	0.690	-	-
	Stage 4	56.54	0.227	36.72	0.564	26.64	0.651	96.40	0.551
EProj	Stage 1	92.16	0.298	-	-	-	-	-	-
	Stage 2	92.16	0.298	123.04	0.470	-	-	-	-
	Stage 3	92.16	0.298	123.04	0.470	54.85	0.637	-	-
	Stage 4	92.16	0.298	123.04	0.470	54.85	0.637	58.59	0.349
PathWeave	Stage 1	100.76	0.358	-	-	-	-	-	-
	Stage 2	78.06	0.234	158.51	0.606	-	-	-	-
	Stage 3	79.07	0.251	138.63	0.547	59.47	0.682	-	-
	Stage 4	66.92	0.255	123.39	0.536	38.53	0.639	97.32	0.554
MERA (1%)	Stage 1	100.76	0.358	-	-	-	-	-	-
	Stage 2	93.70	0.304	153.73	0.573	-	-	-	-
	Stage 3	90.42	0.316	147.42	0.567	57.09	0.678	-	-
	Stage 4	95.18	0.334	142.67	0.562	53.04	0.678	79.32	0.454
MERA (10%)	Stage 1	100.76	0.358	-	-	-	-	-	-
	Stage 2	98.30	0.340	152.20	0.579	-	-	-	-
	Stage 3	96.46	0.346	147.89	0.566	61.49	0.684	-	-
	Stage 4	98.05	0.338	141.25	0.560	56.79	0.678	87.59	0.468

Table 11: Raw data of sequential order training. Results that are better than the last stage are colored in green, indicating a Positive Backward Transfer.

Method		Point Cloud		Audio		Video		Image	
		Cap3D	Cap3D-QA	AudioCaps	Clotho-AQA	MSVD	MSVD-QA	MSCOCO	OK-VQA
Individually Trained Experts		99.93	0.568	60.14	0.658	138.39	0.460	100.76	0.358
Fine-Tuning	Stage 1	99.93	0.568	-	-	-	-	-	-
	Stage 2	2.74	0.178	39.25	0.519	-	-	-	-
	Stage 3	37.26	0.280	21.34	0.158	121.29	0.550	-	-
	Stage 4	26.69	0.199	23.11	0.519	23.40	0.266	86.12	0.342
Replay (1%)	Stage 1	99.93	0.568	-	-	-	-	-	-
	Stage 2	0.98	0.101	48.48	0.640	-	-	-	-
	Stage 3	35.18	0.337	8.30	0.138	124.89	0.546	-	-
	Stage 4	9.39	0.192	17.67	0.498	1.06	0.255	83.38	0.347
Replay (10%)	Stage 1	99.93	0.568	-	-	-	-	-	-
	Stage 2	0.63	0.171	47.38	0.641	-	-	-	-
	Stage 3	68.26	0.470	12.83	0.372	134.07	0.575	-	-
	Stage 4	3.42	0.241	18.33	0.540	2.81	0.228	87.31	0.342
EWC	Stage 1	99.93	0.568	-	-	-	-	-	-
	Stage 2	29.97	0.442	59.31	0.672	-	-	-	-
	Stage 3	19.91	0.375	29.25	0.611	148.26	0.578	-	-
	Stage 4	45.25	0.327	23.39	0.511	47.53	0.524	98.91	0.320
EProj	Stage 1	58.59	0.349	-	-	-	-	-	-
	Stage 2	58.59	0.349	54.85	0.637	-	-	-	-
	Stage 3	58.59	0.349	54.85	0.637	123.04	0.470	-	-
	Stage 4	58.59	0.349	54.85	0.637	123.04	0.470	92.16	0.298
PathWeave	Stage 1	99.93	0.568	-	-	-	-	-	-
	Stage 2	71.79	0.420	55.07	0.648	-	-	-	-
	Stage 3	63.75	0.380	38.73	0.628	148.61	0.577	-	-
	Stage 4	55.23	0.370	37.23	0.603	85.67	0.521	87.04	0.361
MERA (1%)	Stage 1	99.93	0.568	-	-	-	-	-	-
	Stage 2	64.79	0.470	58.77	0.684	-	-	-	-
	Stage 3	66.76	0.377	43.00	0.595	147.99	0.547	-	-
	Stage 4	70.40	0.387	51.02	0.651	140.20	0.538	93.33	0.362
MERA (10%)	Stage 1	99.93	0.568	-	-	-	-	-	-
	Stage 2	87.10	0.505	58.99	0.695	-	-	-	-
	Stage 3	79.24	0.437	58.65	0.650	145.56	0.552	-	-
	Stage 4	81.05	0.425	60.28	0.653	146.72	0.569	97.62	0.367

Table 12: Raw data of reverse order training. Results that are better than the last stage are colored in green, indicating a Positive Backward Transfer.