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

Anonymous ACL submission

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 necessary to reuse the existing ones and further extend them to more modalities through Modality-incremental Continual Learning (MCL). However, this often comes with a performance degradation in the previously learned modalities. In this work, we revisit the MCL and investigate a more severe issue it faces in contrast to traditional continual learning, that its degradation comes not only from catastrophic forgetting but also from the misalignment between the modality-agnostic and modality-specific components. To address this problem, we propose an elegantly simple MCL paradigm called "MErge then ReAlign" (MERA). Our method avoids introducing heavy training overhead or modifying the model architecture, hence is easy to deploy and highly reusable in the MLLM community. Extensive experiments demonstrate that, despite the simplicity of MERA, it shows impressive performance, holding up to a 99.84% Backward Relative Gain when extending to four modalities, achieving a nearly lossless MCL performance.

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), 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. Such architecture and training strategy have been successfully applied to a wide range of modalities, i.e., image(Liu et al., 2024b,a), 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 simultaneously trained on datasets of all predefined modalities. However, it is challenging to extend an existing MLLM to new modalities as it requires another round of joint training on the datasets of previous modalities and the new modalities.

Accordingly, Continual Learning (CL) is proposed to learn from a stream of data. When incrementally learning on the new data, a phenomenon called catastrophic forgetting(McCloskey and Cohen, 1989; Goodfellow et al., 2014) often occurs, 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., 2024) have been proposed to alleviate catastrophic forgetting. Moreover, Modality-incremental Continual Learning (MCL)(Yu et al., 2024a) focuses on the particular scenario of incrementally extending MLLMs to new modalities. However, it is worth noting that: the performance degradation encountered in MCL comes not only from forgetting but also from the misalignment between modality-agnostic and 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. Inspired by the great success of model merging technique in multi-task learning(Yadav et al., 2024a; Yu et al., 2024c; Yang et al., 2024b,a), we introduce model merging to

our MCL framework for the purpose of mitigating forgetting. In this work, we focus on the simplest model merging method, i.e., weight averaging, and revise it into an MCL form, aiming to provide a basic framework. 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 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 revisit the Modality-incremental Continual Learning (MCL) and investigate a more severe issue it faces in contrast to traditional continual learning: its performance degradation comes not only from forgetting 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 demonstrate that our MERA significantly outperforms other representative continual learning methods, and **even achieves a 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 Multimodal Large Language Models (MLLMs) have extended Large Language Models (LLMs) to perceive multimodal inputs such as image, video, audio, point cloud, etc. Early attempts like Flamingo(Alayrac et al., 2022) integrate vision encoders with LLMs via cross-attention to perform image captioning and visual question answering (VQA) tasks. Subsequent methods like InstructBLIP(Dai et al., 2023) leverage instruction tuning to build a general-purpose multimodal system. These attempts lead to the rapid development of MLLMs.

Among these 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 while leveraging visual instruction tuning. 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). 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.

2.2 Continual Learning

Extending existing MLLMs to new modalities can be viewed as a Continual Learning (CL) problem. CL approaches allow models to continually acquire new knowledge with minor forgetting of previously learned knowledge. Existing CL methods mainly fall into the following three 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 the importance matrix requires extra memory with the same scale of the trainable parameters at training time. **Architecture-based methods**(Yu et al., 2024a,b; Zadouri et al., 2024) add task-specific parameters to the base model for each new task. The drawbacks of this category are that it requires modifications of the model architecture, harming its reusability, and the model scale often grows linearly as tasks increase, introducing extra memory overhead. **Replay-based methods**(Scialom et al., 2022; Wang et al., 2024) leverage a small subset of previous data and replay them while learning on new data. A drawback of this category is that it requires access to partial data from the previous tasks or distributions. However, this drawback is relatively minor in real applications as the replay data are often accessible.

Table 1 summarizes the characteristics of different CL categories. Among all the CL methods, replay-based methods are the simplest and most widely used for LLMs since they are extra-train-memory-free and arch-modification-free. In this work, we propose a simple MCL paradigm that has the same advantages as replay-based methods.

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

Table 1: Characteristics of different CL categories and our proposed MERA. Extra-train-memory-free: whether introduces extra GPU memory overhead at training time. Arch-modification-free: whether requires modifying the architecture of the model or adding auxiliary components. Replay-data-free: whether requires 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.

2.3 Model Merging

Model merging is an emerging technique often used for multi-task learning(Yadav et al., 2024a; Yu et al., 2024c; Yang et al., 2024b,a). It aims to combine the strengths of multiple isomorphic models into one unified model. The simplest model merging method is weight averaging(Utans, 1996), where weights from each model are averaged to form the merged model. Model merging is based on the assumption that the models being merged lie in the same flat loss basin(Neyshabur et al., 2020), hence their interpolations also have low losses. Aside from its applications in multi-task learning, model merging can be naturally leveraged to reduce forgetting(Marczak et al., 2024).

3 Analysis of Degradations in MCL

Modality-incremental Continual Learning (MCL) is a special scenario of CL, where models incrementally learn on the data from new modalities. The first attempt of MCL is (Yu et al., 2024a). However, we point out that the MCL faces a more severe problem. In contrast to traditional CL scenarios, where the degradation comes solely from the forgetting of old knowledge, the degradation encountered in MCL comes from two aspects: forgetting and misalignment.

Forgetting: the modality-agnostic components (the LLM backbone for MLLM) forget the knowledge of old modalities.

Misalignment: the modality-agnostic components are misaligned with the modality-specific ones. When adapting to a new modality, the LLM backbone and the newly added modality encoder and connector are updated while the old ones are kept frozen. Therefore, the LLM backbone inevitably drifts away from the original multimodal feature space of old modality encoders (even the feature space of the new modality encoder, if its connector is not synchronously updated with LLM backbone). This process results in misalignment, further worsening the degradation. A potential exception to misalignment is when replay-based methods are applied since the old modality-specific components are also updated synchronously. However, empirical results in Section 5.5 suggest that replay-based methods still suffer from a certain degree of misalignment and it can be compensated by our proposed realigning stage.

Due to the existence of misalignment, MCL problem requires special treatments compared to traditional CL problems.

4 Method

First, 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 $\{D_1, D_2, \ldots, D_m\}$, the model sequentially learns on each D_i with the permission to access a $r\%$ size subset of previous data as replay data[1] $R_i \leftarrow$ sample $r\%$ data from $\{D_1, D_2, \ldots, D_i\}$. We denote the model after the i-th incremental training stage as $\theta_i = \{\theta_i^{agn}, \theta_i^{spec}\}$, where $\theta_i^{agn}, \theta_i^{spec}$ denote modality-agnostic and modality-specific components, respectively. For MLLM, the modality-agnostic component is the LLM backbone, and modality-specific components include encoders and connectors of each modality.

To mitigate performance degradations in the previously learned modalities, we propose a two-stage MCL paradigm called "MErge then ReAlign" (MERA). In each stage of MCL, MERA executes

[1]Different from the definition of replay data in the context of replay-based methods, where R_i is sampled from $\{D_1, D_2, \ldots, D_{i-1}\}$. In this paper, if not explained, R_i for replay-based methods is sampled from $\{D_1, D_2, \ldots, D_{i-1}\}$ while R_i for our realigning stage is sampled from $\{D_1, D_2, \ldots, D_i\}$.

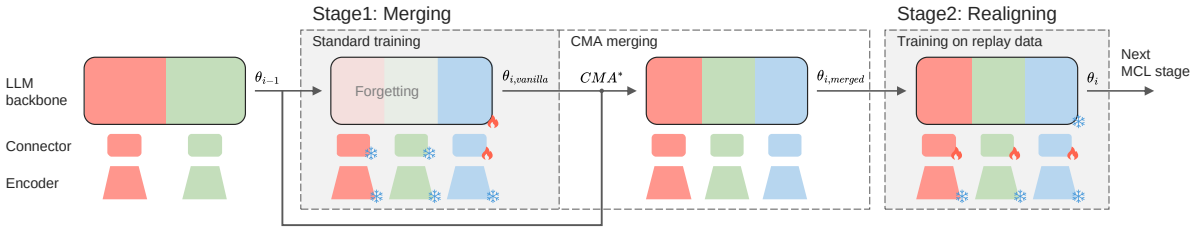

Figure 1: Pipeline of the proposed MERA. The procedures in gray boxes involve training. ❄ and 🔥 represent the frozen and trainable modules, respectively.

the following two stages: merging and realigning, to address the forgetting and misalignment respectively. The overall pipeline of MERA is illustrated in Figure 1.

4.1 Stage 1: Merging

Motivation. Model merging has achieved great success in the field of multi-task learning for its superior capability of combining the strengths of multiple models into one. 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.

However, model merging methods generally can not be directly applied to continual learning. In this work, we only focus on the simplest model merging method, i.e., weight averaging, and revise it into a CL form to provide a basic framework. To adapt weight averaging to CL, we associate its merging coefficients with the progress of CL stages, forming a Cumulative Moving Average (CMA) merging. At the i-th training stage, the CMA merged model is calculated by:

$$CMA(\theta_{i-1}, \theta_i; i) = \frac{i-1}{i}\theta_{i-1} + \frac{1}{i}\theta_i$$

To further adapt to the MCL framework, we only merge the modality-agnostic components (the LLM backbone) and ensemble the modality-specific components (encoders and connectors of each modality):

$$CMA^*(\theta_{i-1}, \theta_i; i) = \{CMA(\theta_{i-1}^{agn}, \theta_i^{agn}; i), \theta_{i-1}^{spec}, \theta_i^{spec}\}$$

To perform MCL, when starting from θ_{i-1} to incrementally learn a new modality M_i, firstly we directly train the model on D_i to get $\theta_{i,vanilla}$. Note that the training process here is the same as the standard way of training an MLLM on a single modality, except that the initial weights of the model are inherited from θ_{i-1}. Then secondly, we apply CMA merging to $\theta_{i,vanilla}$ and θ_{i-1} to derive the merged model $\theta_{i,merged} = CMA^*(\theta_{i-1}, \theta_{i,vanilla}; i)$ with integrated knowledge of both new modality and old modalities.

Discussion. Different from other CL methods, the LLM backbone is asynchronously updated with the new connector in the merging stage. To be specific, in the standard MLLM training step, they are synchronously updated indeed, however, in the CMA merging step, only the LLM backbone is updated, resulting in an additional misalignment between them[2].

4.2 Stage 2: Realigning

To address the misalignment issue, we simply leverage a small replay dataset $R_i \leftarrow$ sample $r\%$ data from $\{D_1, D_2, \ldots, D_i\}$ to further fine-tune all the connectors of $\theta_{i,merged}$ to obtain the final θ_i. This process realigns the encoders of each modality with the LLM backbone. The training objective for realigning is unchanged from the original MLLM training objective, i.e., the auto-regressive loss.

The realigning stage is similar to replay-based CL methods(Scialom et al., 2022; Wang et al., 2024) in form as they both leverage a replay dataset, however, they are essentially different. First, replay methods train on the joint dataset of D_i and R_i, while our realigning stage trains solely on R_i. Second, the purpose of replay is to review the previous knowledge, while the motivation of realigning stage is to align modality spaces. Third, replay methods update all the parameters to consolidate old knowledge, in contrast, keeping the LLM backbone frozen is crucial in the realigning stage, otherwise the LLM would overfit on replay data.

[2]This is the reason we sample R_i from $\{D_1, D_2, \ldots, D_i\}$ rather than $\{D_1, D_2, \ldots, D_{i-1}\}$ in our realigning stage.

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 listed 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 A.

Evaluation Metrics. First, we leverage Relative Gain(Scialom et al., 2022; Wang et al., 2024) as a normalized metric across different tasks. We train 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 degradations 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 in Appendix B. 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 regular 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 B.

Baselines. In our experiments, we compare our MERA with non-CL fine-tuning and the representative methods of each CL category: **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, to review knowledge of earlier tasks while learning the new one. **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. **PathWeave**(Yu et al., 2024a): an architecture-based CL method, also the first 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[3]. Implementation details of each baseline method are in Appendix D.

[3]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.

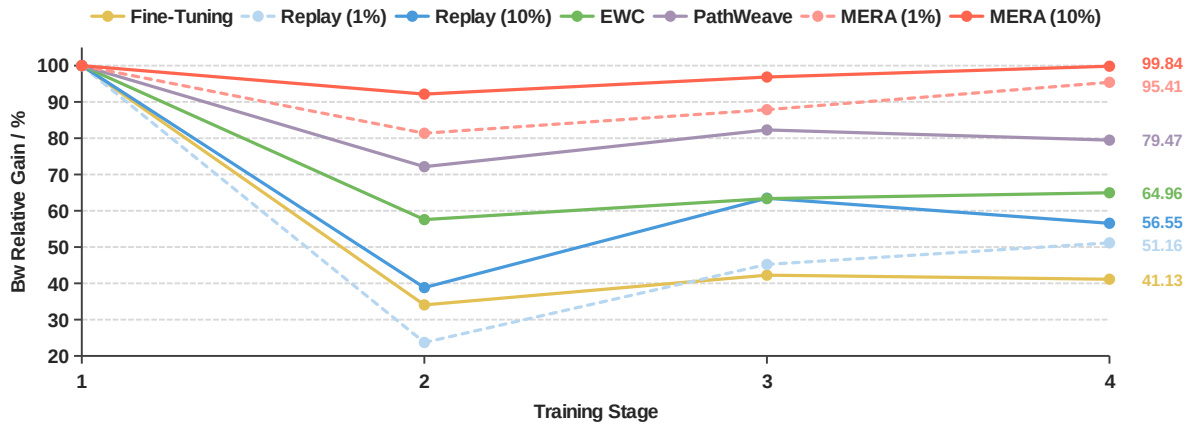

Figure 2: Progressive Backward Relative Gain in modality-incremental continual learning. For each stage i, we plot the average score of corresponding Backward Relative Gain with two different training orders. We set Backward Relative Gain to 100% for the 1-st stage, denoting the initial performance without degradation.

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
PathWeave	86.85	12.17	80.09	13.31
MERA (1%)	<u>97.90</u>	<u>6.02</u>	<u>84.42</u>	<u>12.93</u>
MERA (10%)	**101.00**	**3.90**	**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>.

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. The progressive Backward Relative Gains averaged from different training orders are plotted in Figure 2. 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 the four modalities, MERA (10%) holds up to a 99.84% Backward Relative Gain, indicating that **MERA can achieve a nearly lossless MCL performance**, with at least 20.37% absolute improvements of Backward Relative Gain compared with other baselines. Notably, when only leveraging 1% replay data, MERA (1%) can still achieve at least 15.94% absolute improvements over other baselines.

Further, we calculated the mean and standard deviation of Backward Relative Gains in all training stages for each method, in different training orders. Table 2 shows that our MERA (10%) achieves the highest mean and lowest standard deviation in both training orders, indicating its superior performance and 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 14.15% 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 11.05% and 4.33% absolute improvements in sequential and reverse orders respectively.

From Figure 2 and its raw data shown in Appendix E, we also 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 CL methods and non-CL fine-tuning, the Backward Relative Gain comes to a low level when incrementally learning the first modality (corresponding to the 2-nd training stage), but starts to stabilize or even increase when incrementally learning more modalities.

5.3 Efficiency Comparisons

We further compare the efficiency of different baselines and our MERA, as shown in Table 3. It is observed that our MERA can achieve optimal results

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 3: 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
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. The mean and standard deviation of Backward Relative Gains in all the training stages are reported on different training orders. The best results are in **bold**.

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 or non-CL methods can further improve their Backward Relative Gain and stability. The realigning stage uses 10% replay data. The increased mean indicates better performance, while the decreased standard deviation indicates better stability. Improvements are colored in green.

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.

5.4 Ablation Study

We conduct ablation studies to investigate the effectiveness of each component in MERA. Results are shown in Table 4. Firstly, from Table 4 and Table 2, it is observed that the merging stage alone can already beat many other baselines, achieving the best and second-best performances among baselines in sequential and reverse orders respectively. Secondly, the realigning stage alone can also achieve performance improvements by addressing the misalignment issue. Combining both the 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.

5.5 Misalignment Is 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 *is misalignment a common phenomenon in MCL*? To examine this, we perform

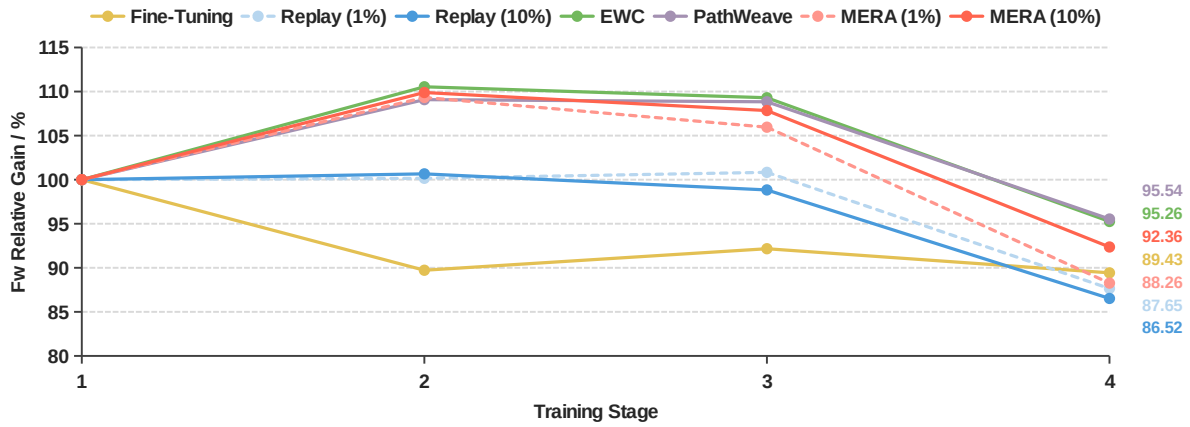

Figure 3: Progressive Forward Relative Gain in modality-incremental continual learning. For each stage i, we plot the average score of corresponding Forward Relative Gain with two different training orders. We set Forward Relative Gain to 100% for the 1-st stage, denoting the initial lossless plasticity.

the realigning stage at the end of every training stage for different CL or non-CL methods to observe whether there are performance improvements. 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 compensated by our proposed realigning stage.

5.6 Plasticity Analysis

Aside from alleviating performance degradation, the capability to adapt to new knowledge, i.e., plasticity, is also an important aspect. We use Forward Relative Gain as the metric to evaluate it. The progressive Forward Relative Gains averaged from different training orders are plotted in Figure 3. It is observed that the most elastic CL methods are EWC and PathWeave, while our MERA (10%) demonstrates comparable plasticity.

Notably, from Figure 3, we observe a strong phenomenon of Positive Forward Transfer(Ke et al., 2021) that the knowledge or skills acquired from earlier tasks improve the learning efficiency of new tasks. The Positive Forward Transfer emerges before the 4-th training stage, on EWC, PathWeave, and MERA. This phenomenon is also reported by other MCL literature(Yu et al., 2024a). In contrast to Positive Forward Transfer, there is a gradual plasticity loss(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 plasticity loss comes to a dominant position.

6 Future Work

This work is one of the early attempts of MCL, mainly focusing on designing effective methods and addressing the misalignment issue. However, our extensive experiments suggest that there are many other interesting aspects of MCL that warrant attention. In Section 5.2 and Section 5.6, we observe the phenomenons of Positive Backward Transfer and Positive Forward Transfer, respectively. This could be related to the complex cross-modal interaction in multimodal learning, urging for further research on the mechanisms of modality interaction and methods that boost a positive modality interaction, in the context of MCL.

7 Conclusion

In this paper, we mainly focus on designing effective methods for Modality-incremental Continual Learning (MCL). First, we revisit MCL and investigate a more severe issue it faces in contrast to traditional continual learning that its performance degradation comes not only from forgetting but also from misalignment. To address both the forgetting and misalignment, we propose MERA, a simple yet effective MCL paradigm. Extensive experiments demonstrate that MERA significantly outperforms the baselines, and even achieves a nearly lossless MCL performance. Further, we observe a sign of complex cross-modal interaction in MCL, providing a direction for future work.

Limitations

Our 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, our experiments are limited to two types of tasks for each modality due to the lack of general-purpose and diverse instruction datasets and their corresponding benchmarks for some modalities, i.e., audio and point cloud. Third, our method 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 any-to-any MLLMs since their architecture topologies are similar. Fourth, our work only explores the simplest model merging method in the context of MCL, aiming to provide a universal framework, leaving the adaptation of other model merging methods to MCL for future work.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*.

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*.

Wenliang Dai, Junnan Li, D Li, AMH Tiong, J Zhao, W Wang, B Li, P Fung, and S Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv 2023. *arXiv preprint arXiv:2305.06500*.

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*.

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*.

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*.

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*.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? *Advances in Neural Information Processing Systems*.

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*.

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*.

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*.

Joachim Utans. 1996. Weight averaging for neural networks and local resampling schemes. In *Proc. AAAI-96 Workshop on Integrating Multiple Learned Models. AAAI Press*.

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*.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024. 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.

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*.

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*.

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*.

Appendix

A Dataset Details

Table 6 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 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.

B 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.

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

We set the hyperparameters mainly following previous worksLiu et al., 2024b,a, as listed in Table 7. 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 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 8. All the reported experimental results are from single runs.

D Implementation of Baselines

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

- **Replay** leverage a replay dataset $R_i \leftarrow$ sample $r\%$ data from $\{D_1, D_2, \ldots, D_{i-1}\}$ when training on a new modality M_i. It can be seen as training on a joint dataset of R_i and D_i.

Hyperparameters	Std-Stage 1	Std-Stage 2	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 7: Hyperparameters for each training stage. Std-stage 1 and Std-stage 2 refer to the stage 1 and 2 of the standard MLLM training process.

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 8: 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 in 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.

E Complete Raw Data

Table 9 and Table 10 show the raw data of Figure 2 and Figure 3.

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
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 9: 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
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 10: Raw data of reverse order training. Results that are better than the last stage are colored in green, indicating a Positive Backward Transfer.