Continual-learning for Modelling Low-Resource Languages from Large Language Models

Santosh Srinath K, Mudit Somani, Varun Reddy Padala,
Prajna Devi Upadhyay, and Abhijit Das
Machine Intelligence Group, Department of CS&IS,
Birla Institute of Technology and Sciences, Pilani, India.
abhijit.das@hyderabad.bits-pilani.ac.in

Abstract

Modelling a language model for a multi-lingual scenario includes several potential challenges, among which catastrophic forgetting is the major challenge. For example, small language models (SLM) built for low-resource languages by adapting large language models (LLMs) pose the challenge of catastrophic forgetting. This work proposes to employ a continual learning strategy using parts-of-speech (POS)-based code-switching along with a replay adapter strategy to mitigate the identified gap of catastrophic forgetting while training SLM from LLM. Experiments conducted on vision language tasks such as visual question answering and language modelling task exhibits the success of the proposed architecture.

1 Introduction

LLMs (Chang et al., 2024) have evolved as a potential aspect of research in the area of artificial intelligence due to their ability to process and interpret both visual and/or textual data for diverse sets of tasks. LLMs often focus on a single modality, but Visual Language Models (VLMs) (Wang et al., 2023; Chen et al., 2023; Li et al., 2019; Chen and Wang, 2022) upscale the context by enabling both visual and textual context at the same time. This capability is particularly important for tasks like image captioning (Stefanini et al., 2022; Hossain et al., 2019), visual question answering (VQA) (Lu et al., 2023; Qian et al., 2024), and multi-modal content generation (Wu et al., 2023; Koh et al., 2024), where understanding both the image context and text is essential. Additionally, VLMs hold immense potential in areas like assistive technology (de Freitas et al., 2022; Xie et al., 2024), automated content creation (Luo et al., 2021), and medical diagnostics (Delbrouck et al., 2022; Boecking et al., 2022).

Most LLMs and VLMs are trained in high-resource languages like English, limiting their use in a multilingual world. Multilingual LLMs and VLMs are essential to ensure inclusivity, enabling models to understand and generate content in various languages and cultural contexts (Ko and Gu, 2022). Despite the importance of multilingual LLMs and VLMs, a significant challenge remains in modelling these for low-resource languages (LRLs). LRLs are referred to as those languages that have relatively less data available for training the model. There are multiple languages across the globe, and it is estimated that there are around 7000 languages in the world (Qin et al., 2024). Unfortunately, most of the technologies are built in English and only 17% of the world's population can speak and understand English as per UNESCO (UNESCO, 2010). A major challenge in natural language processing (NLP) is training language models to perform tasks on LRLs. This creates a gap, as speakers of low-resource languages are often excluded from the benefits of VLM technologies. Multilingual language modelling (MLLM) (Zhu et al., 2023; Geigle et al., 2023) is selected as one of the avenues to model LRLs (Zhou et al., 2021; Li et al., 2023; Zhou et al., 2021).

In MLLM settings, SLMs (Schick and Schütze, 2021) can effectively learn from smaller datasets, via training on LLMs. In other words, LLMs can be fine-tuned to model language models for SLMs without needing the vast amounts of data and resources that large models typically require. Among the existing techniques xMDETR (Wang and et al., 2023) is a recent work where a code switch (CS) is employed for MLLM for visual SLM. Code switch is a technique that is heavily employed in multilingual language modelling to augment datasets for LRL (Wang and et al., 2023; Huang et al., 2020). By definition, code switch is a process of replacing some words in English text in the LRL data following a bilingual dictionary while building the model. M3P (Huang et al., 2020) is a seminal work on code switch used in multilingual multi-modal

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6258–6275

training. In this work, a pre-trained transformer with altered multilingual, mono-lingual, and multi-modal streams was used.

There are also simple methods for cross-lingual language modelling, including training an entirely new model from scratch for each new language or training on all languages simultaneously, which can be inefficient and impractical. When faced with a vast array of languages, we use MLLM techniques to maximise learning from data, especially when data access is limited or short-lived. Continual learning leads to catastrophic forgetting (McCloskey and Cohen, 1989), which brings in the challenge of maintaining previously acquired knowledge (stability) while learning new information (plasticity) (Woł czyk et al., 2021). There are multiple ways of avoiding catastrophic forgetting (CF), such as experience replay (Smith et al., 2024), model expansion (M'hamdi and May, 2024; M'hamdi et al., 2023), regularisation and distillation-based methods (Chen et al., 2021; Feng et al., 2022). To the best of our knowledge, the code switch and catastrophic forgetting have never been addressed in the literature jointly to develop a unified effective model for the multi-lingual scenarios that can get updated with a new set of languages. Moreover, in code switch, syntactical differences are not considered while switching can lead to reduced translation quality, especially for structurally different languages.

On the other hand, one of the model expansion techniques is to integrate adapters to the LLMs. Adapter-based transfer mechanisms (Pfeiffer et al., 2020b) have proven effective for lightweight domain adaptation and modular training in multilingual settings. While MAD-X delivers strong cross-linguistic performance, existing methods often modify shared adapters or task-specific heads, which risks eroding previously acquired language knowledge.

To address limitations in multilingual retention, we introduce replay adapter a novel architectural component that decouples replay optimization from individual language adapters. This approach maintains strong cross-lingual generalization in continual learning settings with minimal increase (~ 2 %) in the overall parameter footprint, thereby enabling scalable and memory-efficient adaptation. Furthermore, to mitigate the dual challenges posed by code switch and multilingual language modeling, we propose a POS-guided code switch strategy integrated with an experience replay mechanism. This combined framework effectively curbs cross-lingual catastrophic forgetting in both vision-language models (VLMs) and large language models (LLMs). The specific contributions of the paper are as follows:

- We propose an adapter-based method that enables experience replay for mitigating catastrophic forgetting in a parameter-efficient way.
- We introduce a novel POS-based code switch strategy designed to preserve syntactic structures during experience replay in continual multilingual learning.

2 Proposed Methodology

2.1 Overview

In the proposed multilingual continual learning, the model learns sequentially across languages in multiple phases. In each phase, the model undergoes fine-tuning on a specific language. The model is trained sequentially with different languages in multiple phases. Let N be the total number of phases, equal to the total number of languages for training. Let l_n is the language introduced in phase p_n, where $n \in \{1, 2, \ldots, N\}$. In each phase p_n, we also replay the previous phase's languages $l_{1:n-1}$ where $n > 1$. We evaluate the performance of the model based on a metric M depending on the task. $M_{n,k}$ is the value of a metric for each language l_k computed at each phase p_n. However, as training progresses, the model often loses the previously acquired language knowledge, a phenomenon known as catastrophic forgetting. In contrast to previous we proposed to use task-specific replay adapters along with multi-adapter. This adapter-based learning paradigm is especially beneficial for low-resource languages, where training data is scarce (Pfeiffer et al., 2020b).

The first phase begins with task-specific training on English. At each phase of continual training, the multilingual text encoder (XLM-RoBERTa Base) is trained using the language adapter of the corresponding language stream and a shared replay adapter. From the second phase onward, i.e. while training the next language, in addition to training the language, a dedicated replay phase is performed on a fixed frequency basis using a POS-based code switch stream, during which only the replay adapter parameters are updated. In other words, the proposed model adopts a modular text encoder

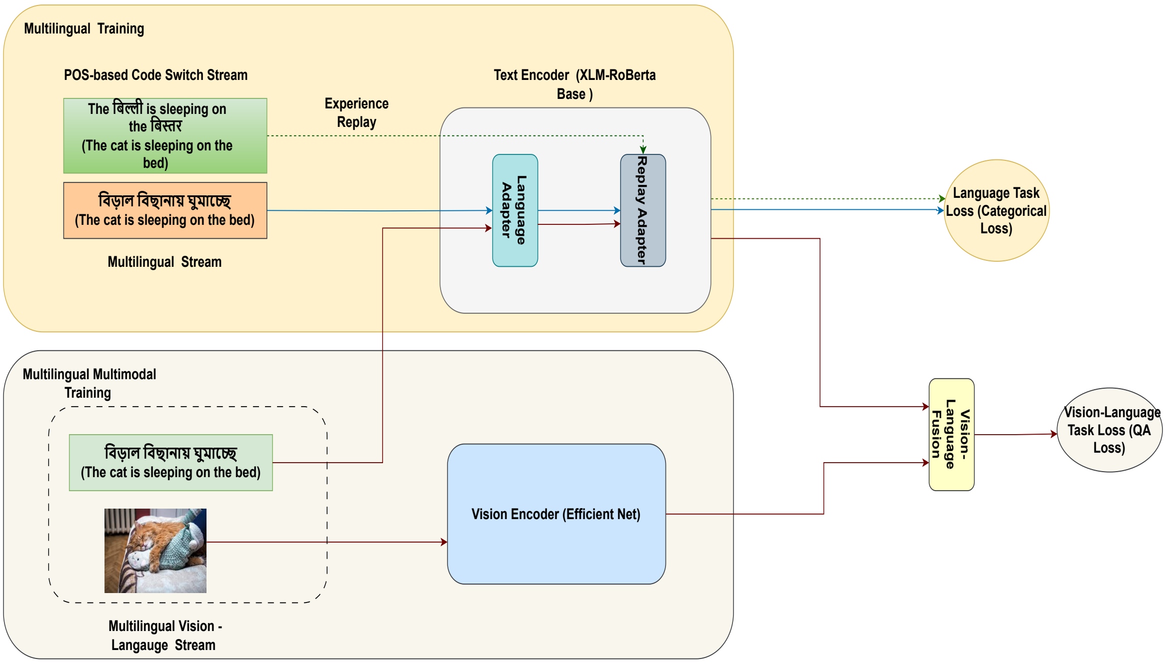

Figure 1: For multilingual text tasks, the model is trained with multilingual stream (blue) using both the language adapter corresponding to each language and a shared replay adapter. During experience replay (green), only the replay adapter is updated using a POS-based code switch stream. During multimodal training (red), the image encoder functions as a visual feature extractor, while the language adapter and replay adapter jointly process textual input. The replay adapter is further updated through a POS-based code switch stream

structure, integrating language-specific adapters for each target language and a unified replay adapter to enable experience replay (See Figure 1). Throughout multilingual continual learning, the encoder backbone remains frozen, while both the language-specific adapters and the shared replay adapter are jointly optimised. During replay phases, only the parameters of the replay adapter are updated, preserving the integrity of previously learned representations while mitigating catastrophic forgetting.

2.2 Proposed POS-Based Code Switch Replay Adapters

During training, we freeze the base model and only update the adapter weights. For each language l_i, we employ distinct adapter A_{l_i} from adapter set A, where $A = \{A_{l_1}, A_{l_2}, ...A_{l_N}\}$. To enable continual learning across languages, we introduce an additional shared task adapter for replay A_{replay}. However, this design offers a crucial advantage using a separate adapter for replay, enabling more effective retention of prior knowledge, especially compared to relying solely on the classification head, which has limited capacity to preserve language-specific information over time.

Code-switching is a phenomenon in linguistics wherein a multi-lingual speaker alternates between two or more languages in the context of a single conversation. It is also a technique which is used to teach a person a new language by switching words between a language they are already familiar with and the target language; the learner is able to grasp the target language more effectively. In a similar manner, code-switching is also used in training multi-lingual models.

POS-based code switching mimics this natural linguistic behavior. In addition to grammatical rules, POS-based code switch replacement also preserves the original meaning and context of the sentence (Arora et al., 2023). For instance, a code switch version of the English sentence *The cat is sleeping on the bed* with Hindi would be *The billi is sleeping on the bistar*, where the nouns *cat* and *bed* are replaced with *billi* and *bistar* respectively. The code mixed sentence still sounds natural despite the replacement. Therefore, taking into account the syntactic cues can be particularly helpful for training multi-language models on low-resource languages.

To foster cross-lingual generalization, we integrate a POS-guided code switch strategy during replay in continual training. As mentioned in the

Algorithm 1, for each input batch, a configurable fraction of tokens, primarily selected via POS tags, are substituted with equivalents sampled from a target language. Guided by the code switch ratio ρ, the subroutine prioritizes token replacements from the specified target part-of-speech (POS) category, where ρ denotes the proportion of tokens to be substituted within each text input during code-switching. If the available POS-tagged tokens are insufficient to meet the quota (α_i), the algorithm supplements them with randomly selected tokens outside the category. This adaptive selection ensures consistent token-level transformation across sentences, irrespective of POS sparsity.

Algorithm 1 POS-Based Code-Switching Subroutine

Input: Batch of sentences $\mathcal{B}$, CS ratio ρ, l_{replay}

Output: Code-switched batch $\mathcal{B}_{cs}$

Initialize code-switched batch $\mathcal{B}_{cs} \leftarrow \emptyset$

for each sentence $s_i \in \mathcal{B}$ **do**

  Compute $\alpha_i = \lceil \rho \cdot |s_i| \rceil$

  Extract POS-tagged set $\mathcal{P}_c^{(i)}$ of category c

  Define random sets:

$$\mathcal{R}_1^{(i)} \subseteq s_i \setminus \mathcal{P}_c^{(i)}, \quad \mathcal{R}_2^{(i)} \subseteq \mathcal{P}_c^{(i)}$$

  Determine switching targets:

$$\mathcal{S}_{cs}^{(i)} = \begin{cases} \mathcal{P}_c^{(i)} & \text{if } |\mathcal{P}_c^{(i)}| = \alpha_i \\ \mathcal{P}_c^{(i)} \cup \mathcal{R}_1^{(i)} & \text{if } |\mathcal{P}_c^{(i)}| < \alpha_i \\ \mathcal{R}_2^{(i)} & \text{if } |\mathcal{P}_c^{(i)}| > \alpha_i \end{cases}$$

  Apply code-switching using l_{replay} equivalent tokens : $\forall w \in \mathcal{S}_{cs}^{(i)},\ w \mapsto \text{CS}(w)$ { $\text{CS}(w)$ contains the word after code-switching }

  Append updated s_i to $\mathcal{B}_{cs}$

end for

Return $\mathcal{B}_{cs}$

2.2.1 Training procedure

The training (Algorithm 2) begins with English as the anchor language. For each language l_t, we initialize a language-specific adapter A_{l_t} while keeping a shared task adapter A_{replay} persistent across all stages. The base encoder remains frozen throughout to ensure parameter efficiency and control for catastrophic interference. During training, batches are streamed from the current dataset $\mathcal{D}_{l_t}$, and model updates are applied at the adapter level only. At each batch step, both the language adapter A_{l_t} and the replay adapter A_{replay} are updated. However, when the batch index satisfies $n \bmod f = 0$ and $t > 1$, where n denotes the batch counter and f is the replay frequency, a replay event is triggered in which only the replay adapter A_{replay} is updated. Replay involves sampling a batch from the English dataset $\mathcal{D}_{l_1}$, transforming it via a POS-based code switch using l_p equivalent tokens ($l_p \sim \{l_2, \ldots, l_t\}$), and updating only the replay adapter with this code-switched input. This targeted strategy ensures retention of prior knowledge and fosters syntactic generalization across diverse typologies without inflating parameter overhead.

Algorithm 2 Continual Training with Returned Code-Switched Replay and Selective Adapter Updates

Input: Language datasets $\mathcal{D}_{l_1}, \ldots, \mathcal{D}_{l_N}$ where $l_1 =$ English, CS ratio ρ, replay frequency f

Initialize: Frozen encoder θ, shared task adapter A_{replay}, language adapters $\mathcal{A} = \{A_{l_1}, \ldots, A_{l_N}\}$

for each language $l_t \in \{l_1, \ldots, l_N\}$ **do**

  Load or initialize language adapter A_{l_t}

  Set batch counter $n \leftarrow 0$

  for each batch $\mathcal{B}_t \in \mathcal{D}_{l_t}$ **do**

    Increment counter: $n \leftarrow n + 1$

    if $t > 1$ and $n \bmod f = 0$ **then**

      Sample replay batch $\mathcal{B}_r \sim \mathcal{D}_{l_1}$

      Sample target language for code-switching $l_{\text{replay}} \sim \{l_2, \ldots, l_t\}$

      Call Code-Switching Subroutine: $\mathcal{B}_{cs} \leftarrow$ CS-Subroutine$(\mathcal{B}_r, \rho, l_{\text{replay}})$

      Update only replay adapter A_{replay} using $\mathcal{B}_{cs}$

    else

      Update language adapter A_{l_t} using $\mathcal{B}_t$

      Update replay adapter A_{replay} using $\mathcal{B}_t$

    end if

  end for

end for

3 Experimental Setup

In this section, we will explain the dataset employed for experiments, experimental setting, results and analysis.

3.1 Dataset

We evaluate proposed method with datasets on diverse tasks of different languages and modalities to understand its effectiveness in mitigation of catastrophic forgetting. For MLVLM experiments, we leverage xGQA (Pfeiffer et al., 2021) to assess cross-lingual multimodal alignment under syntactic perturbations. Its structured multilingual vision question answering format, spanning seven typologically diverse languages, facilitates controlled evaluation of POS-based replay and code-switching strategies, thereby reflecting knowledge retention dynamics and modality-specific interference in continual learning

For MLLM experiments, we use the MTOP (Li et al., 2020) and PAXQA (Li and Callison-Burch, 2023) datasets. MTOP is designed for multilingual intent classification that covers 117 intents. Its parallel annotations is useful for probing syntactic retention, semantic drift in multilingual continual learning setups. PAXQA, a multilingual QA dataset spanning languages such as English, Arabic, Russian, and Chinese, is leveraged to draw comparative insights against multimodal question answering benchmarks. The dataset is generated using parallel corpora like GALE (Linguistic Data Consortium, 2008), NC (Barrault et al., 2020), and UN (Ziemski et al., 2016). Its diverse sources and lexically-constrained translation is useful for continual learning studies. For the experiments, languages are selected from a common source corpus to maintain consistency across training data.

For evaluation on the MTOP and xGQA datasets, missing language variants were supplemented via automatic translation using the NLLB (Team et al., 2022).

3.2 Code-switch Lexicon Preparation

We prepare bilingual lexicon using MUSE (Conneau et al., 2020; Lample et al., 2017) bilingual embeddings, which are aligned with the vocabulary setup in XMDETR (Wang and et al., 2023). To ensure token-level consistency, multiword expressions are excluded, and named entities are transliterated to preserve phonetic fidelity across scripts.

3.3 Metrics

The Table 1 summarizes the primary evaluation metric M used for each dataset to monitor model performance at every epoch. To assess catastrophic forgetting across all datasets, we rely on the average accuracy (Chaudhry et al., 2018).

Dataset	Modality	Task	Metric (M)
MTOP	Text	Intent cls	Accuracy
PAXQA	Text	QA	Exact Match
XGQA	Text+Image	VQA	Accuracy

Table 1: This table outlines the datasets used for evaluation. MTOP involves intent classification (Intent cls), GQA focuses on vision question answering (VQA), and PAXQA covers the question answering (QA) task. The exact match metric (Lewis et al., 2020) is a strict metric, useful for measuring precision in extractive QA tasks.

$$AA = \frac{1}{N}\sum_{k=1}^{N} M_{N,k}$$

where $M_{n,k}$ is the value of a metric for language ℓ_k evaluated at phase p_n and N is the total number of languages in the continual training sequence.

3.4 Implementation details

For all the experiments, we use XLM-RoBERTa Base (Conneau et al., 2020) as the text encoder with 12 layers as a trained multilingual transformer-based encoder model. Instead of fine-tuning the full multilingual text encoder, we employ MAD-X adapter modules (Pfeiffer et al., 2020b) within each Transformer layer and optimize only the adapter parameters, thereby preserving the pretrained encoder's linguistic generalization capabilities. If specific pretrained language adapters are not available, we train the adapter from scratch using the task specific training data. For the replay adapter, we use the same type of adapter as the language adapter (Pfeiffer et al., 2020a). For finetuning the adapters, we employ the Adam optimizer setting the learning rate to 3e-5 with a batch size of 16 for all experiments on different datasets. Across all experiments, we fix the code-switching ratio (ρ) at 0.5 to maintain a balanced distribution between original and switched tokens. Experience replay is integrated into continual training at regular intervals, specifically every 10th batch (f = 10). In each phase of language learning, for the MTOP and PAXQA dataset, we ran a maximum of 10 epochs. Based on our experiments, we have not seen significant improvement in performance beyond 10 epochs for each task.

To train the XGQA dataset, we finetune using both the language adapters and replay adapter integrated to multilingual text encoder (XLM-RoBERTa Base) of the XMDETR architecture

	Lower Bound		POS-based CS						Upper Bound
Language Sequence	No Replay	Random CS	NOUN	VERB	ADJ	ADV	PROPN	INTJ	
MTOP dataset									
$EN \rightarrow FR \rightarrow ES$	84.57	89.55	89.62	89.52	**89.69**	89.4	89.55	89.65	92.38
$EN \rightarrow ES \rightarrow FR$	75.33	89.87	90.05	90.06	90.03	**90.07**	90.03	90.01	92.38
$EN \rightarrow TH \rightarrow KO$	84.19	89.86	89.98	90.01	**90.08**	89.92	90.03	90.04	91.63
$EN \rightarrow DE \rightarrow FR$	86.7	90.52	90.4	90.59	90.6	90.59	90.60	**90.63**	92.34
$EN \rightarrow HI \rightarrow BN$	91.67	91.49	91.54	91.59	**91.80**	91.57	91.60	91.61	92.18
Average	84.49	90.25	90.31	90.35	**90.44**	90.31	90.36	90.38	92.18
XGQA dataset									
$EN \rightarrow PT \rightarrow ES$	29.11	33.31	31.27	34.82	33.23	32.96	**35.05**	33.73	36.83
$EN \rightarrow HI \rightarrow BN$	27.76	27.63	26.20	27.65	26.89	**28.26**	27.19	27.70	37.61
$EN \rightarrow KO \rightarrow ID$	28.23	29.35	28.13	**30.00**	29.56	27.84	29.14	29.21	35.42
Average	28.36	30.09	28.53	**30.82**	29.89	29.68	30.45	30.21	36.62
PAXQA dataset									
$EN \rightarrow AR \rightarrow ZH$	61.68	64.33	**67.49**	64.33	65.13	62.19	64.58	64.46	70.53
$EN \rightarrow RU \rightarrow ZH$	78.20	77.25	77.66	78.12	76.35	77.17	**78.23**	76.38	79.86
$EN \rightarrow ZH \rightarrow RU$	77.45	78.41	78.79	**79.36**	77.98	76.64	77.28	76.99	79.86
$EN \rightarrow RU \rightarrow AR$	74.48	72.39	72.65	71.87	72.65	**75.26**	72.91	72.91	78.12
Average	73.09	73.09	**74.14**	73.42	73.02	72.81	73.25	72.68	77.09

Table 2: This table presents the results of continual training on MTOP, PAXQA and XGQA datasets for various language sequences using average accuracy (Section 3.3). Evaluation of POS-based code switch is focused on POS categories like noun (NOUN), verb (VERB), adjective (ADJ), adverb (ADV), proper noun (PROPN), interjection (INTJ). Details of lower bound and upper bound are mentioned in the Section 3.5. The **bold** values indicate the highest average accuracy (mean across 3 seeds) for the corresponding language sequence. A higher average accuracy signifies better cross-lingual knowledge retention. Statistical analysis of the results is in appendix Table 20

based on proposed method, while keeping the remaining components of the XMDETR architecture frozen. We train the model for 60K steps with a batch size of 16 and a learning rate of 1e-4.

3.5 Baselines

The lower-bound experiments are divided into two types: (1) *no replay* - where prior language data is excluded from experience replay, and (2) *random CS* - where random words are substituted with previous languages during replay. For upper-bound experiments, we employ joint training across all languages rather than sequentially (continual learning).

3.6 Results

We conduct experiments on the MTOP, PAXQA, and XGQA datasets across thirteen languages: English (EN), Chinese (ZH), German (DE), French (FR), Russian (RU), Arabic (AR), Spanish (ES), Portuguese (PT), Korean (KO), Indonesian (ID), Thai (TH), Hindi (HI), and Bengali (BN). Detailed results are presented in Table 2. In our experimental setup, we target open-class categories from the Universal POS tagset (Jurafsky and Martin, 2025) due to their consistent replay effectiveness and minimal cross-lingual interference, as supported by prior findings (de Vries et al., 2022).

Results on MTOP Across diverse language sequences, POS-based code switch driven by adjectives consistently outperform other parts of speech, even though nouns and verbs tend to encode more semantic information compared to other POS categories. Although interjections generally encode limited information, their inclusion in code-switching proves to be notably impactful in enhancing the effectiveness of experience replay. We can also observe that POS-based code switch outperforms random code switch for all language sequences. Another important observation is that the language sequences that are not syntactically similar ($EN \rightarrow HI \rightarrow BN$ and $EN \rightarrow TH \rightarrow KO$) to English (base language for code switch) still benefit POS-based code switch. The Hindi and Bengali are similar languages with high lexical sharing that helps to retain knowledge even without any form of replay.

Results on XGQA POS-based code switch consistently outperforms random code switch across all evaluated language sequences. Notably, the $EN \rightarrow PT \rightarrow ES$ sequence yields the most significant gains from POS-based code switch, likely

Language Sequence	POS-based CS	Random	Sample Size
$EN \rightarrow HI \rightarrow BN$	**91.55**	91.44	10%
	91.70	91.59	30%
	91.53	91.48	50%
	91.61	**91.64**	70%
	91.8	91.49	100%
$EN \rightarrow TH \rightarrow KO$	**89.68**	89.65	10%
	89.80	89.75	30%
	89.90	**90.20**	50%
	89.74	89.66	70%
	90.08	89.86	100%

Table 3: Average accuracy scores on the MTOP dataset using different memory replay sample sizes. **Bold** indicates the top score per language sequence.

	Random CS	POS-based CS
Sequence: $EN \rightarrow FR \rightarrow ES$		
Without Replay Adapter	62.31	62.44
With Replay Adapter	89.55	**89.69**
Sequence: $EN \rightarrow ES \rightarrow FR$		
Without Replay Adapter	81.56	81.57
With Replay Adapter	89.87	**90.07**
Sequence: $EN \rightarrow TH \rightarrow KO$		
Without Replay Adapter	88.60	89.23
With Replay Adapter	89.86	**90.80**
Sequence: $EN \rightarrow DE \rightarrow FR$		
Without Replay Adapter	88.40	88.10
With Replay Adapter	90.52	**90.63**
Sequence: $EN \rightarrow HI \rightarrow BN$		
Without Replay Adapter	91.33	91.33
With Replay Adapter	91.49	**91.80**

Table 4: Average accuracy scores on the MTOP dataset, comparing random and POS-based code switch with and without a replay adapter. **Bold** indicates the highest score per sequence.

due to the high linguistic proximity between Portuguese and Spanish, with performance approaching the upper bound . In contrast, sequences such as $EN \rightarrow HI \rightarrow BN$ and $EN \rightarrow KO \rightarrow ID$ exhibit more modest gains, likely due to the challenges posed by low-resource language status and the increased complexity of multimodal tasks inherent to XGQA.

Results on PAXQA The PAXQA dataset comprises high-resource languages. Across all evaluated sequences, POS-based code switch consistently outperforms random substitution. We also observe that order of languages in language sequences plays an important role in knowledge retention. Although the languages are the same in the sequences $EN \rightarrow RU \rightarrow ZH$ and $EN \rightarrow ZH \rightarrow RU$, the latter sequence has better knowledge retention due to smoother syntactic transition of the languages in continual learning and also probably due to code switch compatibility with English. Though the sequence $EN \rightarrow AR \rightarrow ZH$ yields a lower average accuracy relative to other paths possibly due to Arabic-induced syntactic interference, POS-based code switch still outperform random code switch.

3.7 Comparison with Learning without Forgetting (LWF) and Elastic Weight Consolidation (EWC)

LWF (Li and Hoiem, 2016) mitigates catastrophic forgetting by distilling knowledge from previous tasks without requiring access to old data. It preserves prior task performance by aligning logits between the current and frozen models during continual updates. EWC (Kirkpatrick et al., 2017) is a regularization method that penalizes changes to parameters that are important for previously learned tasks by using the Fisher Information Matrix to estimate their importance. To make it comparable with the proposed approach, we use adapters for training each language task. As shown in Table 5, the proposed method exhibits consistent robustness across languages, outperforming prior approaches particularly in low-resource scenarios.

3.8 Ablation Studies

Impact of Memory Size in Experience Replay The Table 2 represents the results of continual learning by replaying code switch with English as the base language. In this section, we explore the impact of the memory sample size in experience replay on both random and POS-based code switch strategies. In the Table 3, we observe that as the memory sample size increases in the experience replay, POS-based code switch performs better than the random code switch.

3.9 Effectiveness of Replay Adapter

The replay adapter facilitates the preservation of cross-lingual knowledge during experience replay. Without it, the burden of knowledge retention shifts entirely to the classification head. The Table 4 reports the effectiveness of the replay adapter in mitigation of catastrophic forgetting.

3.10 Detailed analysis

POS Tag Frequency and average accuracy Correlation In this section, we discuss the relation between POS frequency distribution and the average accuracy mentioned in Table 2. The correlation analysis follows a two-step approach. We aggregate the POS frequency distributions across languages by computing their average, followed by

Language Sequence	LWF	EWC	Proposed Method
MTOP dataset			
$EN \rightarrow FR \rightarrow ES$	88.94	**90.88**	89.69
$EN \rightarrow ES \rightarrow FR$	88.73	**91.46**	90.07
$EN \rightarrow TH \rightarrow KO$	87.89	89.85	**90.08**
$EN \rightarrow DE \rightarrow FR$	89.33	**91.49**	90.63
$EN \rightarrow HI \rightarrow BN$	78.67	90.32	**91.8**
PAXQA dataset			
$EN \rightarrow AR \rightarrow ZH$	51.71	54.11	**67.49**
$EN \rightarrow RU \rightarrow ZH$	60.88	63.44	**78.23**
$EN \rightarrow ZH \rightarrow RU$	60.5	60.55	**79.36**
$EN \rightarrow RU \rightarrow AR$	61.26	62.37	**75.26**

Table 5: Comparison of proposed method with LWF and EWC. **Bold** indicates the highest average accuracy per sequence.

Langs	NOUN	VERB	ADJ	ADV	PROPN	INTJ
MTOP dataset						
l_2	-0.76	-0.57	**-0.77**	-0.35	-0.57	-0.26
l_3	-0.54	-0.37	**-0.69**	-0.35	-0.61	-0.54
(l_2, l_3)	-0.64	-0.48	**-0.89**	-0.41	-0.59	-0.64
(l_1, l_2)	-0.76	-0.58	**-0.76**	-0.35	-0.57	-0.26
(l_1, l_2, l_3)	-0.64	-0.48	**-0.89**	-0.41	-0.60	-0.67
PAXQA dataset						
l_2	0.02	0.78	-0.16	**0.93**	0.89	-0.86
l_3	0.37	-0.07	**0.57**	0.14	0.36	-0.36
(l_1, l_2)	0.31	0.83	0.35	**0.95**	0.93	-0.92
(l_2, l_3)	0.77	0.65	0.78	0.80	0.93	**-0.99**
(l_1, l_2, l_3)	0.75	0.71	**0.99**	0.79	0.87	**-0.99**

Table 6: This table shows the Pearson correlation between POS frequency in combined training data and average accuracy across language sequences and POS categories. The *Langs* column lists languages used for aggregated POS distribution, and **bold** values mark the strongest negative correlations.

calculating the Pearson correlation with the corresponding average accuracy of POS category. For example, we calculate the average frequency of verbs of all languages in the sequence across all language sequences and then correlate with average accuracy values of all language sequences under verb category as mentioned in the Table 2. As shown in the Table 6, there exists a strong negative correlation between the aggregated POS frequency distribution across all languages (l_1, l_2, l_3) in the sequence and its corresponding average accuracy in the MTOP dataset. The frequency distribution of adjectives demonstrates the highest correlation with the average accuracy, a pattern that remains consistent with the overall average accuracy across all POS categories. In PAXQA dataset, the optimal POS categories differ across language sequences, likely because POS frequency shows little correlation with the choice of optimal category.

Attention Weights Analysis To further analyze the behavior of POS-based code-switch in comparison with random code switch, we quantified the model's internal processing using the attention entropy and attention mass. Attention entropy is a metric that measures the distribution of the model's focus (Clark et al., 2019). Higher entropy values indicate that the model is relying on the broader sentence context to resolve meaning, rather than fixating on specific lexical tokens (keywords). Attention mass is interpreted through the lens of subwords (Rust et al., 2021), this metric analyzes the computational load. A higher attention mass signifies increased token fragmentation, demonstrating that the code-switched anchors impose a greater processing requirement on the encoder.

We conducted experiments and performed statistical analysis on the MTOP dataset using a model trained on the $EN \rightarrow HI \rightarrow BN$ sequence, and measured attention metrics for Hindi. First, the higher attention entropy observed in POS-based code-switching suggests that the model moves beyond simple keyword shortcuts (McCoy et al., 2019) and instead distributes its focus across the full sentence context to verify meaning (Clark et al., 2019). Second, regarding attention mass, our results show that POS-based code-switching produces higher values, indicating that it introduces fragmented, high-density tokens which require greater processing effort from the model (Rust et al., 2021). Overall, our results show that POS-guided code switch provides a stronger training signal than random code-switching.

Metric	Attention Entropy	Attention Mass
Random code switch	2.09	12.43
POS-based code switch	2.18	13.31

Table 7: Comparison of attention metrics by code switch type

Language Specific Forgetting Analysis During training on Bengali in the sequence $EN \rightarrow HI \rightarrow BN$, we incorporate replay from Hindi and track Hindi accuracy after each epoch under various replay strategies, as shown in Figure 2. Notably, POS-based code switch helps retain cross-lingual knowledge more effectively than random code switch. Additionally, there is not much forgetting of Hindi knowledge. This may be attributed to the linguistic proximity between Hindi and Bengali that share substantial lexical overlap.

In contrast, Figure 3 shows that although Thai knowledge degrades over time during training on Korean, code switch based replay still outperforms replay using the target language alone. This can be attributed to the linguistic divergence of the Korean and Thai languages.

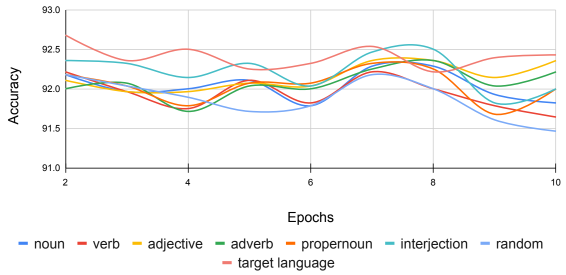

Figure 2: Cross-lingual retention curve for Hindi during Bengali training in the $EN \rightarrow HI \rightarrow BN$ sequence, evaluated after each epoch to assess transfer stability

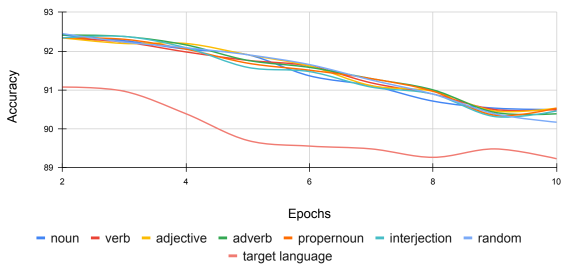

Figure 3: Cross-lingual retention curve for Thai during Korean training in the $EN \rightarrow TH \rightarrow KO$ sequence, evaluated after each epoch to assess transfer stability

Forgetting Analysis using Layer Probing To evaluate linguistic retention throughout continual learning, we extract language-specific embeddings from replay adapters at each encoder layer. These layer-wise representations serve as input to lightweight classifiers trained for task-specific predictions (Alain and Bengio, 2016). Comparing classifier accuracies across layers and phases allows us to track the trajectory of knowledge retention over time. We performed experiments on MTOP dataset for the language sequences $EN \rightarrow HI \rightarrow BN$ (Seq1), $EN \rightarrow FR \rightarrow ES$ (Seq2), $EN \rightarrow TH \rightarrow KO$ (Seq3), $EN \rightarrow ES \rightarrow FR$ (Seq4), $EN \rightarrow DE \rightarrow FR$ (Seq5). The Table 8 reveals that forgetting predominantly occurs at lower encoder layers, while the upper layers exhibit signs of positive backward transfer (Díaz-Rodríguez et al., 2018), indicating reinforcement of earlier linguistic knowledge through replay adapter.

Comparison of the proposed method with adapter fusion method We conducted experiments on the MTOP dataset using the AdapterFusion framework. The continual training followed a two phase design: in the first phase, each language adapter was trained for 7 epochs. In the second phase, the adapter layers were frozen and only the common fusion layer was optimized for 3 epochs by replaying previous languages with random and POS based code switching, where adjectives identified as the optimal POS category based on Table 2 were replaced with target language words. The Table 9 shows that the proposed method outperforms adapter based method.

Layer	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Avg
Layer 1	0	0	0	0	0	0
Layer 2	-1.79	-0.5	-0.87	-1.27	-5.26	-1.93
Layer 3	-0.14	+0.29	+4.13	-1.37	-2.76	+0.03
Layer 4	+0.57	-0.53	-0.04	-0.33	+0.2	-0.02
Layer 5	-1.36	+0.63	-2.50	-1.24	-0.62	-1.01
Layer 6	-3.45	-1.13	+0.04	+3.24	**+0.59**	-0.14
Layer 7	-0.82	-0.29	+0.03	+1.43	-2.19	-0.36
Layer 8	-1.51	-0.47	-0.25	-2.87	-1.89	-1.39
Layer 9	+0.61	-1.07	+1.49	-1.60	-0.09	-0.13
Layer 10	+0.50	**+3.60**	**+9.84**	-8.14	-2.62	+0.63
Layer 11	**+1.65**	+6.80	+0.91	**+20.35**	-4.59	**+5.02**
Layer 12	+1.11	-0.44	+3.98	+14.28	-0.51	+3.68

Table 8: Layer-wise forgetting is quantified in this table via accuracy shifts in the second language (l_2) across phases, capturing retention (positive) and erosion (negative). **Bold** values denote peak retention per sequence, while underlined values indicate pronounced forgetting.

	Adapter Fusion		
Language Sequence	Random	POS	Proposed Method
EN→FR→ES	88.00	87.93	**89.69**
EN→ES→FR	87.98	87.99	**90.07**
EN→TH→KO	87.33	87.25	**90.08**
EN→DE→FR	88.10	86.58	**90.63**
EN→HI→BN	86.51	86.58	**91.80**

Table 9: Performance comparison between adapter fusion method and proposed method

4 Conclusion

This paper proposes a novel continual learning framework for adapting large language models (LLMs) to low-resource languages using adapter-based modular architectures and POS-based code switch exploiting the syntactic cues of the languages. To mitigate catastrophic forgetting in multilingual settings, we introduce a shared replay adapter that retains cross-lingual knowledge during sequential training. The proposed method augments replay data using a POS-based code switch, which replaces tokens in target language inputs with syntactically aligned counterparts from prior languages. Experiments across multilingual tasks including intent classification, question answering, and vision-language understanding demonstrate that POS-guided replay significantly improves retention and generalization, outperforming random and no-replay baselines. The approach scales effectively for continual multilingual model adaptation.

5 Limitations

Linguistic justification for why a specific POS category outperforms other POS categories. Although our approach is parameter-efficient, the exploration of other PEFT methods, including Prefix Tuning and QLoRA, may yield further improvements. To support evaluation of low resource languages, we translated samples using the NLLB model. While this improves multilingual coverage, translation quality may vary across languages.

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Gaurav Arora, Srujana Merugu, and Vivek Sembium. 2023. CoMix: Guide transformers to code-mix using POS structure and phonetics. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7985–8002, Toronto, Canada. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, and 1 others. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*. Association for Computational Linguistics.

Benedikt Boecking, Naoto Usuyama, Shruthi Bannur, Daniel C Castro, Anton Schwaighofer, Stephanie Hyland, Maria Wetscherek, Tristan Naumann, Aditya Nori, Javier Alvarez-Valle, and 1 others. 2022. Making the most of text semantics to improve biomedical vision–language processing. In *European conference on computer vision*, pages 1–21. Springer.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *arXiv preprint arXiv:1801.10112*.

Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. 2023. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56.

Pei-Hung Chen, Wei Wei, Cho-Jui Hsieh, and Bo Dai. 2021. Overcoming catastrophic forgetting by bayesian generative regularization. In *International Conference on Machine Learning*, pages 1760–1770. PMLR.

Xi Chen and Xiao Wang. 2022. Pali: Scaling language-image learning in 100+ languages. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Maurício Pasetto de Freitas, Vinícius Aquino Piai, Ricardo Heffel Farias, Anita MR Fernandes, Anubis Graciela de Moraes Rossetto, and Valderi Reis Quietinho Leithardt. 2022. Artificial intelligence of things applied to assistive technology: a systematic literature review. *Sensors*, 22(21):8531.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from POS tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Jean-benoit Delbrouck, Khaled Saab, Maya Varma, Sabri Eyuboglu, Pierre Chambon, Jared Dunnmon, Juan Zambrano, Akshay Chaudhari, and Curtis Langlotz. 2022. ViLMedic: a framework for research at the intersection of vision and language in medical AI. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 23–34, Dublin, Ireland. Association for Computational Linguistics.

Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. 2018. Don't forget, there is more than forgetting: new metrics for continual learning. *arXiv preprint arXiv:1810.13166*.

Tao Feng, Mang Wang, and Hangjie Yuan. 2022. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9427–9436.

Jay Gala, Pranjal A. Chitale, A. K. Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M. Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Gregor Geigle, Abhay Jain, Radu Timofte, and Goran Glavaš. 2023. mblip: Efficient bootstrapping of multilingual vision-llms. *Preprint*, arXiv:2307.06930.

MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36.

Haoyang Huang, Lin Su, Di Qi, Nan Duan, Edward Cui, Taroon Bharti, Lei Zhang, Lijuan Wang, Jianfeng Gao, Bei Liu, Jianlong Fu, Dongdong Zhang, Xin Liu, and Ming Zhou. 2020. M3p: Learning universal representations via multitask multilingual multimodal pre-training.

Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript, January 12, 2025 draft.

Katharina Kann, Arya Ebrahimi, Karina Stenzel, and Alexis Palmer. 2022. Machine translation between high-resource languages in a language documentation setting. In *Proceedings of the Workshop on Field Matters: Methodological Issues in NLP for Less-Resourced Languages*, pages 19–28, Seattle, Washington. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Danilo Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwińska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Byungsoo Ko and Geonmo Gu. 2022. Large-scale bilingual language-image contrastive learning. *arXiv preprint arXiv:2203.14463*.

Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. 2024. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 36.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Bryan Li and Chris Callison-Burch. 2023. PAXQA: Generating cross-lingual question answering examples at training scale. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 439–454, Singapore. Association for Computational Linguistics.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557.

Zejun Li, Zhihao Fan, Jingjing Chen, Qi Zhang, Xuanjing Huang, and Zhongyu Wei. 2023. Unifying cross-lingual and cross-modal modeling towards weakly supervised multilingual vision-language pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5939–5958, Toronto, Canada. Association for Computational Linguistics.

Zhizhong Li and Derek Hoiem. 2016. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, pages 614–629. Springer.

Linguistic Data Consortium. 2008. GALE Arabic-English Parallel Text – Part 1. `https://catalog.ldc.upenn.edu/LDC2008T09`. LDC Catalog ID: LDC2008T09.

Siyu Lu, Mingzhe Liu, Lirong Yin, Zhengtong Yin, Xuan Liu, and Wenfeng Zheng. 2023. The multimodal fusion in visual question answering: a review of attention mechanisms. *PeerJ Computer Science*, 9:e1400.

Grace Luo, Trevor Darrell, and Anna Rohrbach. 2021. Newsclippings: Automatic generation of out-of-context multimodal media. *arXiv preprint arXiv:2104.05893*.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448. Association for Computational Linguistics.

Meryem M'hamdi and Jonathan May. 2024. Leitner-guided memory replay for cross-lingual continual learning.

Meryem M'hamdi, Xiang Ren, and Jonathan May. 2023. Cross-lingual continual learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3908–3943, Toronto, Canada. Association for Computational Linguistics.

Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin O Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. 2021. xgqa: Cross-lingual visual question answering. *arXiv preprint arXiv:2109.06082*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. *CoRR*, abs/2007.07779.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. 2024. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4542–4550.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers. *Preprint*, arXiv:2404.04925.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, and Sebastian Ruder. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3118–3135, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. *Preprint*, arXiv:2009.07118.

James Seale Smith, Lazar Valkov, Shaunak Halbe, Vyshnavi Gutta, Rogerio Feris, Zsolt Kira, and Leonid Karlinsky. 2024. Adaptive memory replay for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3605–3615.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2022. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 48 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

UNESCO. 2010. UNESCO World Report of Languages. `https://en.wal.unesco.org/about-unesco-wal/unesco-world-report-languages`.

Ying Wang and et al. 2023. Adapting grounded visual question answering models to low resource languages. In *CVPRW*, pages 2595–2604.

Ying Wang, Jonas Pfeiffer, Nicolas Carion, Yann LeCun, and Aishwarya Kamath. 2023. Adapting grounded visual question answering models to low resource languages. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2596–2605.

Maciej Woł czyk, MichałZając, Razvan Pascanu, Ł ukasz Kuciński, and Piotr Mił oś. 2021. Continual world: A robotic benchmark for continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 28496–28510. Curran Associates, Inc.

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

Jingyi Xie, Rui Yu, He Zhang, Sooyeon Lee, Syed Masum Billah, and John M Carroll. 2024. Emerging practices for large multimodal model (lmm) assistance for people with visual impairments: Implications for design. *arXiv preprint arXiv:2407.08882*.

Mingyang Zhou, Luowei Zhou, Shuohang Wang, Yu Cheng, Linjie Li, Zhou Yu, and Jingjing Liu. 2021. Uc2: Universal cross-lingual cross-modal vision-and-language pre-training. *Preprint*, arXiv:2104.00332.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *Preprint*, arXiv:2304.04675.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*.

A Additional Experimental Setup Details

A.1 Software, Computational Resources and Training Time

We use HuggingFace Transformers [1], PyTorch[2], spaCy[3], and AdapterHub[4] for model implementation and preprocessing. All experiments were conducted on an NVIDIA RTX A6000 GPU with 48 GB of VRAM. For continual training with a language sequence length of 3, MTOP completed in approximately 3 hours, PAXQA required 3.5 hours, while XQGA incurred a significantly higher runtime of 47 hours, attributed to the computational demands of the XMDETR architecture. We observe that the runtime difference between random and POS-based code-switching configurations remains within 10 minutes per experiment.

A.2 Hyperparameter Sensitivity

We perform hyperparameter sensitivity analysis by varying batch size, code switch ratio, and training epochs on the MTOP test dataset using our proposed method. Table 13 reports the average accuracy across configurations for the language sequence $EN \rightarrow HI \rightarrow BN$ using our proposed method.

A.3 Dataset Properties

The properties of the MTOP, PAXQ, and XGQA datasets are summarized in Tables 10, 11, and 12, respectively. While MTOP and PAXQ are primarily used for training and validation, XGQA serves as a multilingual evaluation benchmark. To enable training on XGQA using the XMDETR architecture, we leverage the GQA dataset which is in English and apply code-switching techniques to incorporate non-English languages, following the multilingual adaptation strategy outlined in XMDETR (Wang et al., 2023).

Language	Train (Approx.)	Test (Approx.)
English	11,830	4,386
German	9,575	3,549
Spanish	8,085	2,998
French	8,620	3,193
Hindi	7,525	2,789
Thai	7,460	2,765

Table 10: MTOP dataset languages and data splits

Language	Train (Approx.)	Test (Approx.)
English	660,000	1,788
Arabic	141,000	1,788
Chinese	334, 000	1,788
Russian	185,000	1,788

Table 11: PAXQA dataset languages and data splits

B Additional Experiments

B.1 Code Switch Base Language Analysis

In this section, we investigate the choice of the base language in code-switching for experience replay. Departing from the conventional use of English as the default base, we explore alternative high-resource languages such as German and French to assess their impact on cross-lingual knowledge retention in continual learning on MTOP dataset. Table 14 shows that code switch based on the German language outperforms code switch based on the French language. However, code switch based on English (Table 2) performs better than German and French.

B.2 Replacing Adapters with low-rank adaptation (LoRA) Modules in the Proposed Architecture

In this section, we evaluate our proposed method by substituting adapters with LoRA modules. We keep all hyperparameters the same as in the adapter setup, including setting the LoRA rank to 16. As shown in Table 15, the proposed method that uses adapter-based configuration consistently outperforms LoRA across all language sequences.

B.3 Evaluating the Proposed Method under Language Sequence and Length Variation

The Table 2 presents results for language sequences of fixed length 3. To further assess the robustness of our method, we evaluate continual learning performance across varying sequence lengths on MTOP datset. The Table 16 demonstrates that POS-based curriculum strategies consistently outperform random curriculum baselines. Notably, as the language sequence length increases, we observe a gradual decline in average accuracy, indicating the growing challenge of knowledge retention and transfer in longer task sequences.

[1]github.com/huggingface/transformers version 3.4.0 pre-trained on 104 languages, including all languages covered in our evaluation.

[2]https://pytorch.org

[3]https://spacy.io

[4]https://adapterhub.ml

Language	Validation/Test (Approx.)
English	11,000
German	11,000
Portuguese	11,000
Russian	11,000
Indonesian	11,000
Bengali	11,000
Korean	11,000
Chinese	11,000

Table 12: XGQA dataset languages and data splits

Hyperparameter	Value	Average Accuracy
Batch Size	16	91.80
	32	89.08
	64	86.94
Epochs	3	85.61
	5	88.05
	7	89.92
	10	91.80
CS Ratio	0.1	91.62
	0.3	91.72
	0.5	91.80
	0.7	91.57

Table 13: Hyperparameters and Average Accuracy

B.4 Comparison with Other Machine Translation Models

We translated from English version of data to Bengali for MTOP dataset, leveraging English as a high-resource language (Kann et al., 2022), using IndicTrans2-en-indic-1B model (Gala et al., 2023) and compared results against NLLB for the sequence $EN \rightarrow HI \rightarrow BN$. We observe similar results across both NLLB and IndicTrans2 in Table 17, confirming the robustness of our findings.

B.5 Experiments on Decoder-only model

To evaluate the proposed method, we implemented it using LLAMA2 (Touvron et al., 2023). For architectural parity with XLM-Roberta, we applied deep pruning to retain only 12 layers. We prune LLAMA2 using a representative subset of layers: early (layer 0-3) middle (layers 8, 14, 20, 26) and final (layers 28–31). This selection captures hierarchical representations while minimizing computational overhead. This configuration balances representational diversity while reducing computational overhead. We reformulated the MTOP intent classification task as text generation using the template shown in Table 18. We conducted hyperparameter tuning over POS categories and selected the optimal category for code-switching. Table 19 demonstrates two distinct effects: POS-based switching improves generation quality over random switching, and the replay adapter enhances continual learning by reducing forgetting.

Language Sequence	Base Language	Random CS	NOUN	VERB	ADJ	ADV	PROPN	INTJ
$EN \rightarrow FR \rightarrow ES$	DE	88.79	89.23	89.12	88.92	89.46	89.45	89.1
	FR	66.74	65.74	64.14	66.22	66.38	66.37	65.9
$EN \rightarrow ES \rightarrow FR$	DE	89.05	89.38	89.37	89.52	89.18	89.85	88.99
	FR	65.91	67.95	67.45	67.45	68.25	68.61	68.59
$EN \rightarrow TH \rightarrow KO$	DE	89.80	90.24	89.77	90.06	90.19	89.89	90.14
	FR	88.30	88.02	87.53	88.11	88.46	88.52	89.26
$EN \rightarrow DE \rightarrow FR$	DE	90.82	90.02	90.4	90.58	90.65	90.37	90.70
	FR	54.62	58.29	58.93	55.40	56.35	55.94	55.68
$EN \rightarrow HI \rightarrow BN$	DE	91.7	91.59	91.5	91.75	91.17	91.4	91.85
	FR	90.88	91.08	90.04	90.34	90.81	90.94	90.93

Table 14: Analysis of base languages for code switch

Language Sequence	Random CS	NOUN	VERB	ADJ	ADV	PROPN	INTJ	Proposed Method
$EN \rightarrow FR \rightarrow ES$	57.00	57.07	57.03	56.3	57.82	56.97	57.06	**89.69**
$EN \rightarrow ES \rightarrow FR$	58.87	59.98	60.3	60.85	59.88	60.32	60.92	**90.07**
$EN \rightarrow TH \rightarrow KO$	59.61	59.63	59.56	59.68	59.45	59.33	59.51	**90.08**
$EN \rightarrow DE \rightarrow FR$	66.36	65.17	64.09	65.46	64.82	65.52	65.4	**90.63**
$EN \rightarrow HI \rightarrow BN$	83.6	83.75	83.51	83.82	83.78	83.4	83.74	**91.8**

Table 15: Comparison of LoRA and adapter setup

Language Sequence	Random CS	NOUN	VERB	ADJ	ADV	PROPN	INTJ
$EN \rightarrow FR \rightarrow ES \rightarrow DE$	89.01	89.02	**89.03**	88.98	88.95	89.12	**89.03**
$EN \rightarrow FR \rightarrow ES \rightarrow DE \rightarrow HI$	84.65	85.21	**85.59**	85.13	84.49	85.00	85.2
$EN \rightarrow FR \rightarrow ES \rightarrow DE \rightarrow HI \rightarrow BN$	79.12	79.01	79.35	79.41	79.32	79.15	**79.93**
$EN \rightarrow TH \rightarrow KO \rightarrow BN$	**88.49**	88.34	88.26	88.21	88.29	88.26	88.33
$EN \rightarrow TH \rightarrow KO \rightarrow BN \rightarrow HI$	88.42	**88.43**	88.09	88.19	88.42	88.22	88.23
$EN \rightarrow TH \rightarrow KO \rightarrow BN \rightarrow HI \rightarrow DE$	88.44	88.38	87.89	**88.55**	88.28	88.32	88.39

Table 16: Evaluation of proposed method on different language sequence lengths

Metric	NLLB	IndicTrans2
No Replay	**91.67**	90.00
Random	**91.49**	90.70
NOUN	**91.54**	91.46
VERB	**91.59**	91.45
ADJ	**91.80**	91.54
ADV	**91.57**	91.07
PROPN	**91.60**	91.05
INTJ	**91.61**	91.34
Joint Training	**92.18**	92.03

Table 17: Vertical comparison of NLLB and IndicTrans2 across replay strategies and POS categories.

Component	Content
A. Prompt Template	
Template	`[INST] «SYS»\n{system_message}\n«/SYS»\n\n{text}\n[/INST] [IN:{label}]`
B. Instantiated Example	
`{system_message}`	You are a task-oriented classification model. Output the top-level intent tag (e.g., [IN:TRAVEL_BOOKING]).
`{text}`	book a flight from hyderabad to delhi for tomorrow
`{label}`	`TRAVEL_BOOKING`

Final Prompt (Training)	`[INST] «SYS»`
	`You are a task-oriented classification model. Output the top-level intent tag (e.g., [IN:TRAVEL_BOOKING]).`
	`«/SYS»\n\nbook a flight from hyderabad to delhi for tomorrow\n[/INST] [IN:TRAVEL_BOOKING]`

Final Prompt (Inference)	...‘book a flight from hyderabad to delhi for tomorrow\n[/INST] [IN:‘

Table 18: The prompt structure for intent classification. Section A shows the general template with placeholders. Section B provides a concrete example, followed by the final formatted prompts used for model training and inference.

Language Sequence	No Replay	Random CS	POS-based CS
$EN \rightarrow FR \rightarrow ES$	**74.11**	73.98	73.8
$EN \rightarrow ES \rightarrow FR$	74.57	**74.72**	74.57
$EN \rightarrow TH \rightarrow KO$	58.16	**60.16**	**60.16**
$EN \rightarrow DE \rightarrow FR$	72.88	73.19	**73.22**
$EN \rightarrow HI \rightarrow BN$	54.59	54.88	**55.13**

Table 19: Performance evaluation of the proposed method using LLAMA2

	Lower Bound		POS-based CS						Upper Bound
Language Sequence	No Replay	Random CS	NOUN	VERB	ADJ	ADV	PROPN	INTJ	
MTOP dataset									
$EN \rightarrow FR \rightarrow ES$	84.57 (0.051)	89.55 (0.062)	89.62 (0.078)	89.52 (0.090)	**89.69** (0.031)	89.4 (0.067)	89.55 (0.092)	89.65 (0.087)	92.38 (0.077)
$EN \rightarrow ES \rightarrow FR$	75.33 (0.099)	89.87 (0.034)	90.05 (0.056)	90.06 (0.089)	90.03 (0.056)	**90.07** (0.043)	90.03 (0.098)	90.01 (0.032)	92.38 (0.066)
$EN \rightarrow TH \rightarrow KO$	84.19 (0.033)	89.86 (0.056)	89.98 (0.021)	90.01 (0.089)	**90.08** (0.020)	89.92 (0.032)	90.03 (0.045)	90.04 (0.067)	91.63 (0.069)
$EN \rightarrow DE \rightarrow FR$	86.7 (0.022)	90.52 (0.056)	90.4 (0.089)	90.59 (0.025)	90.6 (0.080)	90.59 (0.037)	90.60 (0.029)	**90.63** (0.023)	92.34 (0.064)
$EN \rightarrow HI \rightarrow BN$	91.67 (0.021)	91.49 (0.089)	91.54 (0.098)	91.59 (0.037)	**91.80** (0.073)	91.57 (0.021)	91.60 (0.080)	91.61 (0.037)	92.18 (0.049)
XGQA dataset									
$EN \rightarrow PT \rightarrow ES$	29.11 (1.211)	33.31 (0.720)	31.27 (1.246)	34.82 (1.784)	33.23 (1.236)	32.96 (1.144)	**35.05** (0.901)	33.73 (1.093)	36.83 (1.234)
$EN \rightarrow HI \rightarrow BN$	27.76 (0.123)	27.63 (0.312)	26.20 (0.173)	27.65 (0.190)	26.89 (0.411)	**28.26** (0.109)	27.19 (0.236)	27.70 (0.147)	37.61 (0.101)
$EN \rightarrow KO \rightarrow ID$	28.23 (0.143)	29.35 (0.090)	28.13 (0.341)	**30.00** (0.211)	29.56 (0.111)	27.84 (0.321)	29.14 (0.213)	29.21 (0.312)	35.42 (0.512)
PAXQA dataset									
$EN \rightarrow AR \rightarrow ZH$	61.68 (1.040)	64.33 (0.234)	**67.49** (0.234)	64.33 (0.341)	65.13 (0.126)	62.19 (0.321)	64.58 (0.131)	64.46 (0.141)	70.53 (0.171)
$EN \rightarrow RU \rightarrow ZH$	78.20 (0.323)	77.25 (0.341)	77.66 (0.122)	78.12 (0.246)	76.35 (0.612)	77.17 (0.239)	**78.23** (0.232)	76.38 (0.443)	79.86 (0.341)
$EN \rightarrow ZH \rightarrow RU$	77.45 (0.249)	78.41 (0.441)	78.79 (0.431)	**79.36** (0.211)	77.98 (1.141)	76.64 (1.141)	77.28 (1.141)	76.99 (1.141)	79.86 (1.141)
$EN \rightarrow RU \rightarrow AR$	74.48 (0.244)	72.39 (0.243)	72.65 (0.351)	71.87 (0.243)	72.65 (0.335)	**75.26** (0.243)	72.91 (0.123)	72.91 (0.247)	78.12 (0.243)
Average	73.09	73.09	**74.14**	73.42	73.02	72.81	73.25	72.68	77.09

Table 20: This table presents the results of continual training on MTOP, PAXQA and XGQA datasets for various language sequences using average accuracy (Section 3.3) across 3 seeds.