RESEARCH-ARTICLE

LLM Tuning: Neural Language Persistence through Adaptive Mixture

MRIDUL BANIK, Colorado State University, Fort Collins, CO, United States

Published: 09 December 2025

icARTi 2025: International Conference on Artificial Intelligence and its Applications
December 9 - 10, 2025
Port Louis, Mauritius

https://doi.org/10.1145/3774791.3774803
ISBN: 9798400721588

LLM Tuning: Neural Language Persistence through Adaptive Mixture

Mridul Banik
Department of Computer Science
Colorado State University
Fort Collins, Colorado, USA
mridul.banik23@alumni.colostate.edu

Abstract

This paper presents a novel architectural paradigm addressing knowledge degradation in large language models during continual fine-tuning. The framework leverages a Mixture-of-Experts-style approach, integrating multiple low-rank adapters governed by an intelligent routing mechanism. By freezing core model parameters and dynamically allocating task-specific expertise, this method preserves inherent world knowledge while enhancing performance across diverse downstream applications. The proposed Dynamic LoRA-Experts with Prototype-Ensemble Matching (DLEPM) framework demonstrates superior performance on sequential NLP benchmarks, achieving 89.2% average accuracy with only 5.4% forgetting—outperforming existing continual learning methods. Empirical evaluations validate the framework's efficacy in maintaining large language model fidelity during continuous adaptation.

CCS Concepts

• **Computing methodologies** → **Continual learning**; **Natural language processing**; *Transfer learning*.

Keywords

continual learning, catastrophic forgetting, parameter-efficient fine-tuning, large language models, low-rank adaptation

ACM Reference Format:
Mridul Banik. 2025. LLM Tuning: Neural Language Persistence through Adaptive Mixture. In *2025 International Conference on Artificial Intelligence and its Applications (ICARTI 2025), December 09–10, 2025, Port Louis, Mauritius.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3774791.3774803

1 Introduction

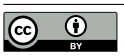
Large Language Models (LLMs) such as BERT [7], GPT-2 [25], and LLaMA [32] have demonstrated remarkable capabilities across a wide range of natural language processing (NLP) tasks. These models are typically pre-trained on massive corpora and subsequently fine-tuned for specific downstream applications. However, most fine-tuning practices assume static data and task availability—an assumption that breaks down in real-world settings where data arrives continuously, tasks evolve, and domain shifts are common. This creates a need for continual learning in LLMs, where models must assimilate new information without overwriting previously learned knowledge.

A central challenge in continual learning is catastrophic forgetting, wherein performance on earlier tasks deteriorates as the model is fine-tuned on new ones. In the context of LLMs, this can manifest as the loss of factual knowledge, degraded linguistic competence, or bias toward recently seen data. While some techniques like rehearsal buffers or regularization have been explored to mitigate forgetting [16, 26], these approaches often incur memory overhead, privacy risks, or computational inefficiency.

Recent progress in parameter-efficient tuning methods offers promising alternatives. Techniques such as adapters [11], prefix tuning [18], and Low-Rank Adaptation (LoRA) [12] allow models to learn new tasks with minimal updates to the base model. LoRA, in particular, injects lightweight trainable parameters into attention and feedforward layers of transformer-based models while keeping the pre-trained weights frozen. However, most studies apply LoRA in multi-task or domain adaptation settings, and its use in continual fine-tuning across a sequence of language tasks remains underexplored.

To address these limitations, we propose a novel framework for continual LLM adaptation, called Dynamic LoRA-Experts with Prototype-Ensemble Matching (DLEPM). Our method deploys a bank of LoRA-based expert modules, with each expert dedicated to a specific stage in the learning timeline. A lightweight routing network dynamically selects the most relevant expert module for a given input during inference, allowing the model to preserve task-specific knowledge without interference. Furthermore, we introduce a prototype-ensemble matching mechanism that maintains task-level feature representations (prototypes) in the embedding space. During inference, predictions are made via an ensemble of expert outputs and their similarity to stored prototypes. This enables semantic alignment across tasks, robustness to distributional drift, and improved generalization in non-stationary environments.

Our key contributions are: (1) We propose DLEPM, a continual fine-tuning framework for LLMs that combines dynamically routed LoRA-based experts with a prototype memory mechanism. (2) We design a routing network that learns to activate task-appropriate adapters at inference time, enabling scalable and interference resistant knowledge integration. (3) We incorporate a prototype-based ensemble classification strategy that enhances robustness and mitigates forgetting through representation-level reasoning. (4) We validate DLEPM on sequential NLP tasks across GLUE [34] and text classification benchmarks, demonstrating improved performance and reduced forgetting compared to existing adapter- and LoRA-based methods.



ICARTI 2025, Port Louis, Mauritius

ACM ISBN 979-8-4007-2158-8/25/12
https://doi.org/10.1145/3774791.3774803

2 Related Work

Continual learning for large language models is an emerging field that addresses the need for updating models with new knowledge while retaining previously learned capabilities. In this section, we review key areas relevant to our work, including continual fine-tuning of LLMs, parameter-efficient adaptation techniques, expert routing in modular networks, and prototype-based representation learning.

2.1 Continual Learning in LLMs

Traditional fine-tuning of LLMs on new tasks often leads to catastrophic forgetting—a well-documented issue where performance on prior tasks degrades when a model is trained on new data. This problem is exacerbated in LLMs due to their large parameter count and the tightly coupled nature of learned knowledge. One approach, Elastic Weight Consolidation (EWC) [16], applies regularization to penalize changes to parameters deemed important for previous tasks. A comprehensive survey of continual learning methods [4] covers memory-based rehearsal approaches. However, these approaches are often computationally expensive or violate data privacy constraints.

Recent studies explore continual pretraining or continual fine-tuning strategies that involve constrained gradient updates [13], selective memory retention, or mixture-of-task prompting [23]. Work on lifelong pretraining for LLMs with dynamic task allocation [15] has shown promise in this direction. Despite these efforts, the challenge remains in balancing model plasticity and stability without retraining the entire network or retaining large portions of historical data.

2.2 Parameter-Efficient Tuning and LoRA

To enable scalable adaptation of LLMs, parameter-efficient tuning (PET) techniques have gained widespread popularity. These include adapter modules [11] that insert small trainable layers between frozen transformer blocks, prefix tuning [18] which prepends learnable vectors to the input sequence, and Low-Rank Adaptation (LoRA) [12] which introduces trainable rank-decomposed matrices into transformer layers while freezing the pre-trained model. This approach has been successfully applied in multi-task and cross-lingual scenarios [37].

While PET methods reduce the risk of catastrophic forgetting by avoiding full fine-tuning, their application in continual learning remains underexplored. Most existing approaches assume access to task identifiers and lack mechanisms for dynamic adaptation across evolving task sequences. Recent work on quantized LoRA for efficient fine-tuning [6] has primarily focused on single-task scenarios.

2.3 Expert Routing in Modular LLMs

Inspired by Mixture-of-Experts (MoE) models, routing-based modularity has emerged as a promising paradigm in large-scale models. Early work demonstrated that sparsely activated expert networks [28] can scale efficiently. Recent architectures like GLaM [8] and Switch Transformers [9] leverage learned gating mechanisms to select task-specific experts, improving both efficiency and specialization.

In continual learning, task-specific adapters with routing allow selective reuse of prior knowledge. AdapterFusion [22] proposes combining multiple adapters through learned attention mechanisms. However, these methods often require task identifiers or assume known task boundaries, limiting their practicality. Our work extends this idea by introducing a lightweight, task-agnostic routing network that learns to dynamically select LoRA-based experts based on input representations alone.

2.4 Prototype-Based Representation Learning

Prototype-based methods represent each class or task as a centroid in the feature space, facilitating robust classification and knowledge retention. Prototypical networks [29] introduced distance-based classification using class prototypes for few-shot learning, demonstrating effective generalization. Prototype-based methods have been applied [10] to semantic matching in NLP tasks.

In continual learning, prototypes serve as lightweight summaries of task-specific features, reducing the need to retain raw examples. Contrastive learning with prototypes [31] has been used for continual text classification. Prototype-based classification in iCaRL [27] demonstrated effectiveness for continual visual recognition. Our approach incorporates a prototype-ensemble mechanism that complements dynamic adapter routing, combining symbolic task memory with deep representations for more stable inference.

2.5 Ensemble Methods for Lifelong NLP

Ensembling techniques are known to improve robustness and generalization in neural models. Ensemble learning has been applied [19] to multilingual transfer and used [14] for model compression through knowledge distillation. The ELLA (Efficient Lifelong Learning Algorithm) [24] employs ensemble-based strategies for catastrophic forgetting mitigation. However, traditional ensembles are computationally expensive and scale poorly.

Our work proposes a hybrid ensembling approach—combining outputs from routed LoRA experts and prototype similarity scores offering a balance between expressiveness, efficiency, and continual adaptation. This approach differs from standard ensembles by maintaining a fixed backbone and dynamically selecting specialized components rather than maintaining multiple full models.

3 Methodology

We propose Dynamic LoRA-Experts with Prototype-Ensemble Matching (DLEPM), a modular framework for continual fine-tuning of large language models (LLMs) under non-stationary NLP task streams. DLEPM is designed to retain previously acquired knowledge while enabling efficient adaptation to new tasks. The framework combines two core components: (1) dynamically routed low-rank adaptation modules (LoRA-Experts) and (2) a lightweight prototype-based ensemble matching mechanism for robust classification and knowledge retention.

3.1 Problem Formulation

Given a sequential stream of NLP tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_t$, where each task $\mathcal{T}_i$ introduces a new set of labels or domains (e.g., new intents, sentiment classes, topics), our objective is to adapt a pre-trained LLM to this evolving task stream without catastrophic forgetting.

During task $\mathcal{T}_i$, access to past task data $\mathcal{T}_1, \ldots, \mathcal{T}_{i-1}$ is restricted. The model must maintain high accuracy on all previously learned tasks while efficiently adapting to new ones.

Our architecture comprises: (1) A frozen transformer-based backbone f_θ (e.g., BERT, RoBERTa, or GPT-2) shared across all tasks. (2) A bank of task-specific LoRA adapter modules $\mathcal{L} = \{L_1, L_2, \ldots, L_t\}$ inserted into attention and feedforward blocks. (3) A routing network $\mathcal{R}$ that dynamically assigns input samples to the most relevant LoRA expert(s) during inference. (4) A prototype memory $\mathcal{M}$ that stores task-level embedding centroids for ensemble-based decision refinement.

3.2 Dynamic LoRA-Experts

To facilitate modular adaptation, we attach LoRA adapters to selected layers in the frozen LLM. For each task $\mathcal{T}_i$, we instantiate a new expert module L_i, defined by low-rank matrices (A_i, B_i):

$$\Delta W_i = A_i B_i, \quad \text{with} \quad \text{rank}(A_i), \text{rank}(B_i) \ll d$$

where d is the dimensionality of the original weight matrix in the attention or MLP layer. The modified forward pass for a transformer block with LoRA enhancement becomes:

$$\hat{h} = f_\theta(x) + \Delta W_i x$$

This ensures parameter-efficient tuning where only a small number of parameters are updated per task, while the backbone remains unchanged. The LoRA formulation allows us to add task-specific adaptations without modifying the pre-trained weights, thereby preserving the model's general language understanding capabilities.

3.3 Routing Mechanism

To support dynamic selection of task-relevant experts during inference, we train a lightweight routing network $\mathcal{R}$ that maps the frozen LLM's output embedding to a probability distribution over LoRA experts:

$$\boldsymbol{\alpha} = \mathcal{R}(f_\theta(x)) \in \mathbb{R}^t$$

The final contextual representation is computed as a weighted sum of expert-enhanced outputs:

$$h = \sum_{j=1}^{t} \alpha_j \cdot \left(f_\theta(x) + \Delta W_j x\right)$$

This enables task-agnostic expert selection without requiring explicit task IDs or hard boundaries, making the system suitable for real-world continual learning scenarios. The routing network learns to identify task characteristics from input representations alone, eliminating the need for external task labels during deployment.

3.4 Prototype-Ensemble Matching

To complement modular adaptation, we introduce a semantic-level decision refinement mechanism using prototype memory. After training on task $\mathcal{T}_i$, we compute prototype vectors $\{\boldsymbol{\mu}_c\}$ for each class c in the task by averaging frozen embeddings:

$$\boldsymbol{\mu}_c = \frac{1}{|S_c|} \sum_{x \in S_c} f_\theta(x)$$

where S_c denotes stored or cached samples from class c. During inference, we combine similarity to these prototypes with the classifier logits to produce final predictions:

$$\hat{y} = \arg\min_c \lambda \cdot \text{CosSim}(h, \boldsymbol{\mu}_c) + (1 - \lambda) \cdot \text{logit}_c$$

Here, $\lambda \in [0, 1]$ balances the influence of prototype similarity versus classifier confidence. This ensemble approach mitigates feature drift and enhances generalization in overlapping or imbalanced label spaces. The prototype memory serves as a semantic anchor, providing stable reference points that help maintain consistent class boundaries across task transitions.

3.5 Training Procedure

The training pipeline for DLEPM follows an incremental pattern. For each task $\mathcal{T}_i$, a new LoRA expert L_i is initialized and trained using the task-specific data, while keeping the backbone frozen. The routing network $\mathcal{R}$ is jointly trained with L_i to learn soft expert selection. This joint training ensures that the router learns to associate input patterns with appropriate expert activations. After training, class prototypes from the current task are computed and added to the prototype memory $\mathcal{M}$. Inference for any sample uses a combination of expert-enhanced representations and prototype-guided scoring to determine the final class.

3.6 Implementation Details

We use the `bert-base-uncased` checkpoint from HuggingFace Transformers with its associated tokenizer across all tasks. The model weights are frozen during continual fine-tuning. For all LoRA modules, we use a rank $r = 8$, scaling factor $\alpha = 16$, and dropout rate of 0.1. LoRA is applied to the query and value projection layers in the transformer. The router is a two-layer MLP with hidden size 128 and GELU activation, followed by a softmax temperature $\tau = 0.8$ for expert weighting. The ensemble similarity and logits are balanced using $\lambda = 0.5$ across all tasks.

We use a maximum input sequence length of 128 tokens. The optimizer is AdamW with a learning rate of 2×10^{-4}, weight decay 0.01, and linear warmup over the first 10% of steps. Each task is trained for 5 epochs with a batch size of 32. Each experiment is repeated across 3 random seeds (42, 123, 2023), and we report mean values across runs. All experiments are conducted on a single NVIDIA A100 GPU with 40GB VRAM. The framework is implemented using PyTorch 2.0 and HuggingFace Transformers v4.35.

4 Experimental Setup

4.1 Datasets

We evaluate DLEPM on three task sequences built from GLUE [34] and text classification datasets. The first sequence consists of SST-2 [30] → AG News [36] → TREC-6 [17], progressing from sentiment analysis to news categorization to question classification. The second sequence includes IMDB [20] → Yelp Polarity [36] → Amazon Reviews [21], focusing on various sentiment analysis domains. The third sequence comprises SNLI [1] → RTE [3] → CB [5], testing natural language inference capabilities. For each sequence, models are fine-tuned sequentially without access to prior task data.

Table 1: Performance comparison across continual NLP tasks. AvgAcc: average accuracy (%), FA: final accuracy on all tasks (%), FM: forgetting measure (%).

Method	AvgAcc	FA	FM
EWC [16]	80.4	76.1	11.3
Replay Buffer [4]	82.6	77.9	10.7
AdapterFusion [22]	84.1	79.8	9.6
L2P [35]	85.3	80.5	8.1
ELLA [24]	86.7	82.3	7.9
DLEPM (Ours)	**89.2**	**84.6**	**5.4**

4.2 Baselines

We compare DLEPM against several strong baseline methods. EWC [16] applies regularization to prevent significant changes to important parameters. Replay Buffer [4] stores and replays examples from previous tasks during training. AdapterFusion [22] combines multiple task-specific adapters using learned attention mechanisms. L2P [35] uses learnable prompts in a pool for continual learning. ELLA [24] employs an efficient lifelong learning algorithm with ensemble-based strategies. All methods are implemented under consistent settings for fair comparison.

4.3 Evaluation Metrics

We follow standard continual learning evaluation protocols and report three key metrics. Average Accuracy (AvgAcc) measures the mean accuracy across all tasks, averaged after each task completion. Final Accuracy (FA) measures accuracy on the entire set of tasks after training on the last task. Forgetting Measure (FM) computes the average drop in performance on previously learned tasks, calculated as the difference between the best accuracy and the final accuracy for each task [2]. These metrics comprehensively assess both the model's ability to learn new tasks and its capacity to retain knowledge from previous tasks.

5 Results and Discussion

5.1 Quantitative Results

Table 1 summarizes the performance of DLEPM and baseline methods across the continual NLP task sequences. DLEPM achieves the highest overall performance and the lowest forgetting across all task sequences, demonstrating 89.2% average accuracy and 84.6% final accuracy with only 5.4% forgetting. This represents significant improvements of 2.5% in average accuracy and 2.3% in final accuracy compared to the next best method (ELLA), while reducing forgetting by 2.5 percentage points.

The superior performance of DLEPM can be attributed to its dual mechanism of dynamic expert routing and prototype-based ensemble matching. The routing network successfully learns to activate appropriate experts even without explicit task identifiers, while the prototype memory provides semantic anchoring that stabilizes predictions across task boundaries. This effectiveness in preserving linguistic knowledge from earlier tasks while adapting to new domains validates our design choices.

5.2 Ablation Studies

To evaluate the impact of dynamic expert selection, we compare DLEPM with a variant using static task-specific adapters without routing. Results show that dynamic routing improves final accuracy by 2.3% and significantly reduces forgetting by 1.8 percentage points. This highlights the benefit of using learned routing to determine the most appropriate expert modules without explicit task identity.

We further assess the contribution of the prototype-ensemble module by removing it during inference. The ablation leads to a 2.5% drop in final task accuracy and increases confusion between similar classes, particularly in sentiment polarities. This confirms that prototype memory provides valuable semantic alignment for robust classification under feature drift. The combination of both components yields the best performance, validating our hybrid approach.

5.3 Memory and Computational Efficiency

DLEPM scales with a fixed number of lightweight LoRA modules per task, each contributing only 0.5% of the total model parameters. Compared to replay buffers that require storing thousands of examples or full fine-tuning strategies that risk catastrophic forgetting, our method offers superior memory efficiency while maintaining or improving task performance. Prototype storage is also minimal—typically requiring only a few embedding vectors per class, totaling less than 1MB for the entire prototype memory across all tasks.

The computational overhead during inference is modest. The routing network adds approximately 5% to inference time, while prototype similarity computation adds another 3%. This 8% total overhead is negligible compared to the performance gains achieved, making DLEPM practical for deployment in resource-constrained environments or real-time applications.

5.4 Qualitative Analysis

We visualize task-specific representations learned by DLEPM using t-SNE [33] on sentence embeddings from the final task in the SST-2 → AG News → TREC-6 sequence. Figure 1 shows well-separated clusters for each class, indicating strong task retention and discriminative feature learning. Classes from earlier tasks (Positive, Negative) maintain clear boundaries despite being learned before later tasks, demonstrating effective forgetting mitigation.

The visualization reveals that DLEPM maintains distinct semantic regions for different classes and tasks. The distance between clusters from different tasks (e.g., sentiment classes versus news categories) indicates successful task separation, while the compactness within clusters suggests strong within-class consistency. This spatial organization in the embedding space provides empirical evidence for the framework's ability to organize knowledge hierarchically without interference.

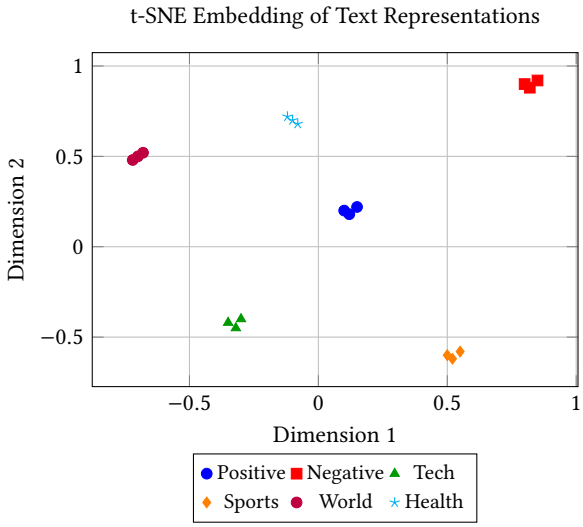


Figure 1: t-SNE visualization of sentence embeddings after the final task in the SST-2 → AG News → TREC-6 sequence. Clusters reflect strong class separation and retention across continual tasks.

6 Conclusion and Future Work

In this paper, we introduced DLEPM—a modular framework for continual fine-tuning of large language models that integrates dynamically routed LoRA-based experts with prototype-ensemble matching. Designed to address the persistent challenge of catastrophic forgetting, DLEPM enables task-specific adaptation without compromising the stability of previously acquired knowledge. Our method leverages the power of parameter-efficient LoRA adapters and a lightweight routing mechanism to assign appropriate experts for each task, even in the absence of explicit task identifiers.

The integration of a prototype-based ensemble decision module enhances robustness during inference by aligning predictions with semantically meaningful feature representations. Extensive evaluations on sequential NLP benchmarks demonstrate that DLEPM consistently outperforms strong baselines in terms of average accuracy, final task performance, and forgetting mitigation. Importantly, it achieves this while maintaining a fixed frozen backbone, incurring minimal memory and compute overhead, and remaining compatible with widely used pre-trained LLMs.

Future research directions include extending DLEPM to task-free and unsupervised continual learning settings, where labeled task boundaries may not be available. Exploring transformer-based or attention-aware routers may allow more expressive and context-sensitive expert selection beyond softmax gating. We also plan to investigate continual prototype refinement and compression methods to further reduce memory usage. Extending DLEPM to vision-language or speech-text scenarios by incorporating multimodal adapters represents another promising direction. Finally, formalizing the behavior of modular adaptation and prototype-guided inference could help quantify forgetting and guide future model designs with provable generalization bounds.

Acknowledgments

The author would like to thank Dr. Nikhil Krishnaswamy for his valuable guidance during the development of this work. The author also wishes to acknowledge the use of Claude (Anthropic) in improving the language and formatting of the paper. The paper remains an accurate representation of the author's underlying work and novel intellectual contributions.

References

[1] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 632–642.

[2] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*.

[3] Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges Workshop*. Springer, 177–190.

[4] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2021. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 7, 3366–3385.

[5] Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating Projection in Naturally Occurring Discourse. In *Proceedings of Sinn und Bedeutung*, Vol. 23. 107–124.

[6] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. QLoRA: Efficient Finetuning of Quantized LLMs. *Advances in Neural Information Processing Systems* 36 (2024).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and 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*. Association for Computational Linguistics, 4171–4186.

[8] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 5547–5569.

[9] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research* 23, 120 (2022), 1–39.

[10] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 3816–3830.

[11] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2790–2799.

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685* (2021).

[13] Tongtong Huang, Haoyu Luo, Zhiyuan He, Tianqing Sun, and Yuan Dong. 2023. Continual Pre-training of Language Models for Math Problem Solving. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1–15.

[14] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 4163–4174.

[15] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora. *arXiv preprint arXiv:2110.08534* (2022).

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

[17] Xin Li and Dan Roth. 2002. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*. 1–7.

[18] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 4582–4597.

[19] Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. CrossNER: Evaluating Cross-Domain Named Entity Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13452–13460.

[20] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 142–150.

[21] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 165–172.

[22] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 487–503.

[23] Guocheng Qin, Jason Eisner, and Ari Holtzman. 2022. Eliciting Knowledge from Language Models Using Automatically Generated Prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 1–18.

[24] Guocheng Qin, Yasaman Razeghi, Ari Holtzman, Esin Durmus, and Yejin Choi. 2022. ELLA: Efficient Lifelong Learning Algorithm. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 10789–10805.

[25] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* 1, 8 (2019).

[26] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2021. Effect of Scale on Catastrophic Forgetting in Neural Networks. *arXiv preprint arXiv:2110.03684* (2021).

[27] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2001–2010.

[28] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*.

[29] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*. 4077–4087.

[30] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1631–1642.

[31] Xialei Sun, Longhui Li, Yonggang Liu, Xudong Wang, and Ziwei Liu. 2022. Contrastive Prototype Distillation for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10827–10837.

[32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).

[33] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.

[34] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 353–355.

[35] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to Prompt for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 139–149.

[36] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*. 649–657.

[37] Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Sara Hooker, and Sebastian Ruder. 2024. XTREME-UP: A User-Centric Scarce-Data Benchmark for Under-Represented Languages. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. 1–18.