Once Read is Enough: Finetuning-free Language Models with Cluster-guided Sparse Experts for Long-tail Domain Knowledge

Anonymous Author(s)
Affiliation
Address
email

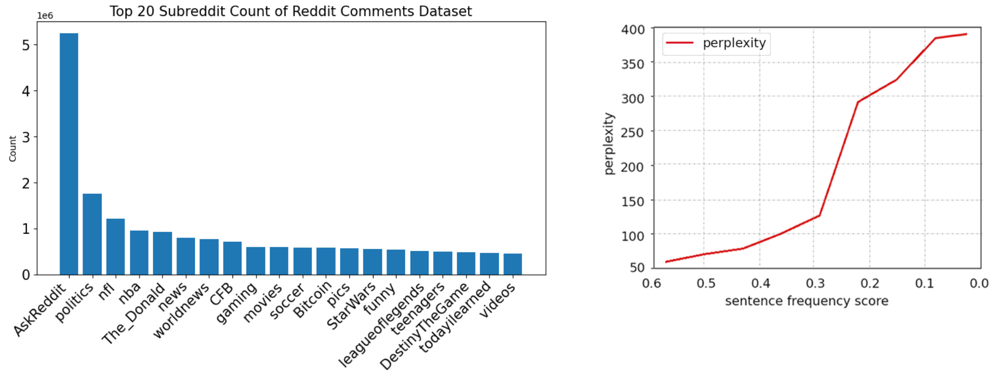
Abstract

Language models (LMs) only pretrained on a general and massive corpus usually cannot attain satisfying performance on domain-specific downstream tasks, and hence, finetuning pretrained LMs is a common and indispensable practice. However, domain finetuning can be costly and time-consuming, hindering LMs' deployment in real-world applications. In this work, we consider the incapability to memorize domain-specific knowledge embedded in the general corpus with rare occurrences and "long-tail" distributions as the leading cause for pretrained LMs' inferior downstream performance. Analysis of Neural Tangent Kernels (NTKs) reveals that those long-tail data are commonly overlooked in the model's gradient updates and, consequently, are not effectively memorized, leading to poor domain-specific downstream performance. Based on the intuition that data with similar semantic meaning are closer in the embedding space, we devise a Cluster-guided Sparse Expert (CSE) layer to actively learn long-tail domain knowledge typically neglected in previous pretrained LMs. During pretraining, a CSE layer efficiently cluster domain knowledge together and assign long-tail knowledge to designate extra experts. CSE is also a lightweight structure that only needs to be incorporated in several deep layers. With our training strategy, we found that during pretraining, data of long-tail knowledge gradually formulate isolated, "outlier" clusters in an LM's representation spaces, especially in deeper layers. Our experimental results show that only pretraining CSE-based LMs is enough to achieve superior performance than regularly pretrained-finetuned LMs on various downstream tasks, implying the prospects of finetuning-free language models.

1 Introduction

In natural language processing, it is a prevalent paradigm to pretrain language models (LMs) on a large-scale unlabeled corpus covering a plethora of knowledge, and those pretrained LMs have exhibited impressive performance in language tasks in the general domain [40]. When it comes to downstream tasks requiring specialized domain knowledge, e.g., legal search or medical question answering [24, 7], those models usually fail to expertise in such knowledge and cannot acquire desirable performance. As such, finetuning on domain-specific datasets is deemed essential to fulfill pretrained LMs' potential in various downstream tasks [22, 14, 41, 36]. However, finetuning an LM could require domain expertise from humans, for instance, the involvement of a doctor for healthcare tasks [31], which can be costly and laborious. The associated catastrophic forgetting issue [27] could further complicate the finetuning process.

In this work, we re-visit the pretraining-finetuning paradigm and raise the following question: *is finetuning indispensable to LMs*? Notably, the domain-specific knowledge necessary for various



(a) (b)

Figure 1: a) The top 20 subreddits with the highest amount of data in the Reddit Comments Dataset, where a typical long-tail distribution can be observed. b) Language Models struggle to memorize long-tail domain knowledge during pre-training. The less frequently a sentence appears in the training corpus, the higher its perplexity, indicating that it is not effectively memorized.

downstream tasks is usually embedded in the pretraining corpus of extensive information sources. Those pieces of domain-specific information may only appear a few times in the massive corpus, significantly less frequently than other ubiquitous and general knowledge, and there can be numerous pieces of such rare information, a distribution usually defined as “long-tail”. In Fig 1(a), we plot the frequency of the top-20 subreddits count on Reddit Comments Dataset [2], and a typical long-tail distribution can be observed. Previous works have verified that LMs are not good learners of long-tail knowledge in the pretraining dataset with Question-Answering as the downstream task[21]. Our experiments, as shown in Figure 1(b), further illustrate that pretrained LMs do not adequately retain domain-specific knowledge in long-tail sequences which is evidenced by a surge in perplexity corresponding to decreased frequency score. This could result in inferior performance on downstream tasks. Finetuning improves LMs' domain performance by providing a second lesson, which could be avoided if the first (pretraining) is appropriately delivered.

To unveil the hidden mechanisms under LM's incapability to learn long-tail domain-specific knowledge, we investigate the behaviors of a GPT on the Wikipedia dataset. We examine LMs' learning capabilities on long-tail data by analyzing the Neural Tangent Kernels (NTKs) of long-tail data and all data. Recent research [5, 39] has indicated that the updating of deep networks can be governed by the gradient direction corresponding to the principle eigenvector of an NTK matrix, which reflects the most common gradient-descending direction across the entire input space. Following those works, we consider an NTK's principle eigenvector (PE) gradient direction as a primary indicator of an LM's overall gradient-updating direction over a data space. Our analysis has revealed that the PE gradient direction of long-tail data, indicating the gradient-descending direction from long-tail knowledge, is generally diverged from that of overall data, which rules the overall updating of network parameters. The observation that long-tail data cannot substantially impact LMs' parametric updates under regular pretraining settings explains pretrained LMs' incompetence on domain-specific knowledge of rare occurrences, necessitating an effective solution.

To this end, we propose the Cluster-guided Sparse Expert (CSE) layer, an effective, efficient, and easy-to-implement approach to improve LMs' long-tail knowledge awareness. In a CSE layer, with intuition such that data with similar semantic meaning are closer in the embedding space, we perform efficient clustering on the embeddings to group data from different domains, and additional experts will be assigned to explicitly and appropriately memorize the information within those clusters. Models trained with CSE show pronounced cluster structure in the embedding space, where long-tail data forms small, outlier clusters. We empirically demonstrate that converting several deep layers into CSE ones can be enough to achieve satisfying results, such as the last two layers of GPT[29] or BERT[10], and the incurred computational costs are comparatively small and arguably acceptable. We have verified that pretrained CSE-based LMs have outperformed regularly pretrained-finetuned LMs on downstream tasks from various domains, which implies that domain finetuning may not be essential if long-tail knowledge can be sufficiently learned.

Our contributions are summarized as follows:

- We have presented that datasets show a long-tail distribution, with domain specific data in the long-tail, and revealed that long-tail data cannot substantially affect LMs' training, which is a leading cause of LMs' incompetence on learning rare, domain-specific knowledge.
- We have devised a Cluster-guided Sparse Expert (CSE) architecture to better pretrain LMs to memorize the long-tail domain knowledge. With such a training strategy, LMs can effectively capture long-tail domain data in the representation space as outlier clusters, thereby enhancing their ability to handle less frequent contexts efficiently.
- Promising performance on downstream tasks has verified the effectiveness of the proposed method, indicating that finetuning may not be indispensable to LMs.

2 Analysis of Long-Tail Domain Data

In this section, we first elucidate the challenges associated with learning from long-tail data through gradient analysis. We then explore the embedding space using the Cluster-guided Sparse Expert (CSE) layer, which effectively captures the structural nuances of long-tail data. Furthermore, we examine the dynamics of these clustering structures, offering insights into how the learning processes of long-tail clusters adapt and evolve across various training stages and model layers.

2.1 Challenges in Learning Long-Tail Domain Data

This subsection explores the significant challenges posed by long-tail domain data within language models (LMs). The primary issue stems from the divergence in gradient directions between long-tail data and the general gradient-updating trajectory of these models, which critically hampers effective learning.

2.1.1 Preliminaries and Definitions

Informed by seminal works [12, 19], we utilize Neural Tangent Kernels (NTKs) to scrutinize the gradient behavior of neural networks under a gradient descent training regime. The NTK, represented as $\boldsymbol{\Theta}(\mathcal{X}, \mathcal{X})$, is defined as the outer product of the gradients of network outputs relative to its parameters $\boldsymbol{\Theta}(\mathcal{X}, \mathcal{X}) = J_{\boldsymbol{\theta}}(\mathcal{X}) J_{\boldsymbol{\theta}}(\mathcal{X})^\top$, where $J_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} f(\mathcal{X}; \boldsymbol{\theta})$ denotes the Jacobian matrix of the function f at the data points $\mathcal{X}$.

To determine the predominant gradient-descending direction across the input space, which is influenced by the gradient direction associated with the principal eigenvector of the NTK matrix, we first perform an eigenvalue decomposition of the NTK matrix. Recognized as a positive semi-definite real symmetric matrix, the NTK decomposes into $\boldsymbol{\Theta} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$. Here, n represents the total number of training instances. The principal eigenvector $\mathbf{u}_{max}$ is identified as the vector corresponding to the maximum eigenvalue. Then the primary gradient direction for a given input set $\mathcal{X}$ is $\mathbf{g}_{\boldsymbol{\theta}}(\mathcal{X}) = \mathbf{u}_{max} J_{\boldsymbol{\theta}}(\mathcal{X})$. Building upon above preliminaries, we introduce the metric of Gradient Consistency (GC) to evaluate the alignment between gradient directions for specific data subsets and the overall dataset.

Definition 1 *(Gradient Consistency (GC)). Let $\mathcal{X}'$ be a specific subset of the training set $\mathcal{X}$. The gradient consistency of $\mathcal{X}'$ is evaluated by computing the cosine similarity between the most prevalent gradient direction of $\mathcal{X}'$ and that of the entire dataset $\mathcal{X}$:*

$$GC_\theta(X') = \frac{\mathbf{g}_{\boldsymbol{\theta}}(\mathcal{X}) \cdot \mathbf{g}_{\boldsymbol{\theta}}(\mathcal{X}')}{\|\mathbf{g}_{\boldsymbol{\theta}}(\mathcal{X})\| \|\mathbf{g}_{\boldsymbol{\theta}}(\mathcal{X}')\|}. \tag{1}$$

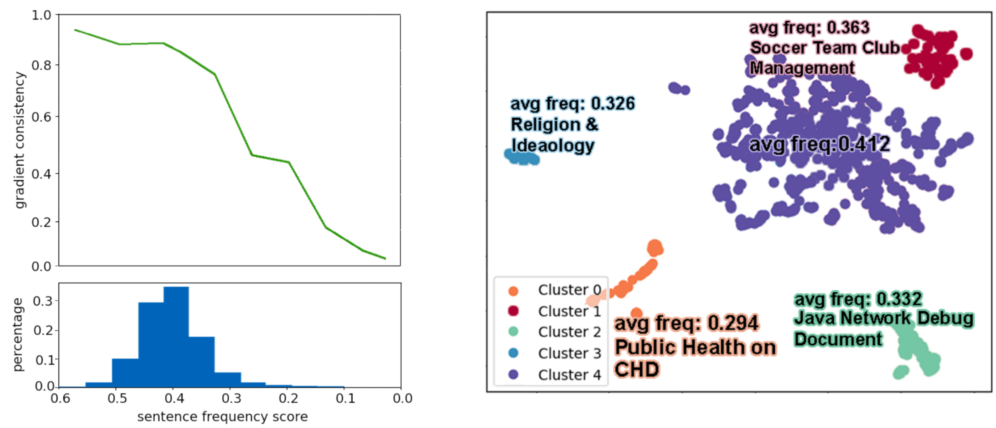
A higher GC value indicates that the model's optimization updates are well-aligned with the needs of the specific subset X', suggesting focused and effective learning of this data. Conversely, a lower value indicates suboptimal learning of these data, pointing to potential areas for improvement in model training strategies.

2.1.2 Gradient Consistency (GC) Analysis

We assess the sentences from Wikipedia on a standard GPT model using sentence frequency score to gauge how frequently each sentence appears in the corpus. This score is calculated by averaging the frequency of its constituent tokens. Figure 2(a) displays the relationship between GC and sentence frequency score. Additionally, the figure includes a histogram that details how many percentage of sentences across the whole dataset falling into each frequency bin.

There is a significant correlation between gradient consistency and the frequency with which sentences appear in the corpus. Notably, for sentences less frequently encountered in the dataset, the model demonstrates substantial ineffectiveness in learning. As demonstrated, the GC value sharply declines from 0.8 to 0.4 as the sentence frequency score decreases from 0.3 to 0.2. Furthermore, the GC value continues to diminish as the sentence frequency score decreases further, indicating that the model's gradient descent direction struggles to align with the requirements of these rare sentences.

Our analysis indicates that the optimization requirements for long-tail sentences are significantly overlooked under standard pretraining conditions, resulting in the unique characteristics of long-tail domain data not being effectively captured. This oversight substantially impairs the performance of LMs when learning domain-specific knowledge involving rare occurrences, underscoring the need for a more effective solution.



(a) (b)

Figure 2: a) The correlation between sentence frequency score and gradient consistency. A histogram is also included showing how many percentage of sentences across the whole dataset falling into each frequency bin. b) A sampled embedding space containing 4 long-tail clusters, taken from our CSE layers.

2.2 Embedding Space Analysis With Cluster-guided Sparse Expert (CSE) layer

Prior research[2] has shown that extensive domain-specific data reside within the long-tail distribution of a general pretraining corpus, as illustrated in Figure 1(a). These data, often semantically similar, are likely to cluster closely within the embedding space, facilitating potential aggregation for dedicated learning. However, our analysis in Section 2.1 underscores significant challenges in learning from long-tail data. Specifically, the model's gradient updates frequently fail to align with the optimization needs of these data, leading to their underrepresentation in the embedding space. Such misalignment obscures the inherent group structures that these domain data form based on their semantic similarities, thereby impeding dedicated learning efforts.

To address the issues outlined above and to facilitate a more effective examination of long-tail domain data in the embedding space, we propose the Cluster-guided Sparse Expert (CSE) layer. This layer groups proximate long-tail data points into clusters and directs them to specialized experts for dedicated learning. As demonstrated in Figure 3(a), the GC value of long-tail data initially increases at the beginning of the training stage but rapidly declines thereafter, indicating that the model's inability to capture the learning dynamics of long-tail data begins early in the training process. Our CSE layer capitalizes on the clustering structure at the point where the GC value peaks, subsequently taking effect to channel domain-specific clusters into dedicated learning pathways. Further details about this approach are provided in Section 3.

The clustering results from the CSE-based LM, shown in Figure 2(b), reveal four smaller clusters alongside a predominant one. Detailed analysis shows high domain coherence within the smaller clusters, each comprising sentences closely related to specific domains. The average sentence frequency score of these domain clusters falls into the long-tail of the sentence frequency distribution, as shown in Figure 2(a). In contrast, the predominant cluster, colored in purple, contains a diverse mix of more common data and exhibits a higher average sentence frequency compared to the smaller clusters. Further analysis of sentences with frequency scores below 0.2 shows their random distribution across clusters, suggesting these extremely infrequent sentences may serve as noise in the learning process.

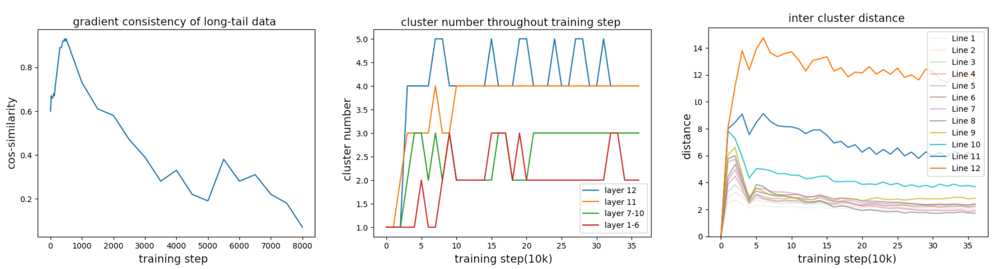
This analysis demonstrates that our proposed CSE-based architecture effectively groups long-tail data from the same domains for dedicated learning, fostering a domain-specific clustering structure within the embedding space. The long-tail domain clusters, distinct from clusters containing common data, show a higher degree of compactness and are clearly separated, highlighting the unique features embodied by these clusters.

2.3 Dynamic of Long-Tail Domain Clusters

In this subsection, we explore the learning dynamics of long-tail domain data by tracking how clusters evolve across different training stages and model layers. We utilize K-Means clustering [20] and employ the elbow method to determine the optimal number of clusters.

Long-tail clusters can be seen early in the training stage. As shown in Figure 3(b) and Figure 3(c), the number of clusters quickly peaks early in the training stage, accompanied by a peak in inter-cluster distances. This indicates that our CSE-based architecture effectively promotes the formation of a clustering structure early on.

The swift emergence of these clusters signifies substantial model adaptation to global features at the start of training, allowing for effective differentiation between clusters. As training progresses, inter-cluster distances gradually decrease, suggesting a stabilization in the learning dynamics and a potential shift in focus toward refining intra-cluster nuances.

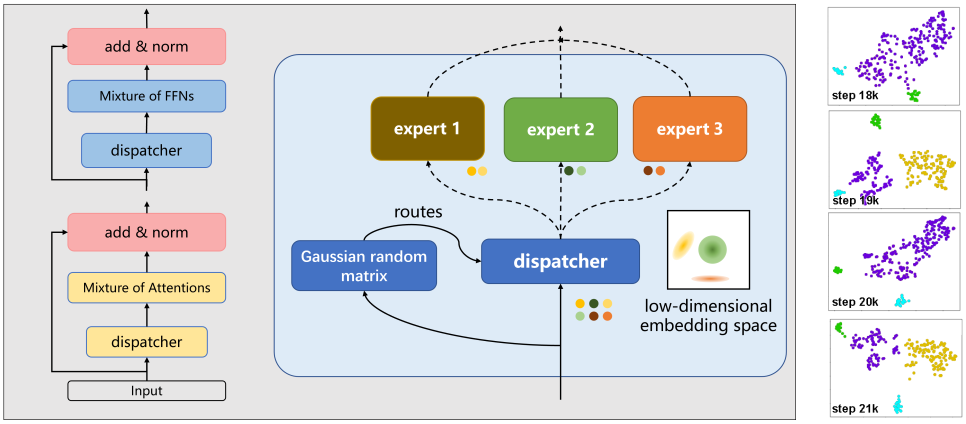


(a) (b) (c)

Figure 3: a) Evolution of the Gradient Consistency (GC) of long-tail data over the former 8000 training steps. GC scores beyond this range are omitted, as they consistently remain below 0.2. b) Evolution of number of clusters over training steps. c) Evolution of inter-cluster distances over training steps.

Long-tail clusters become more pronounced with increasing network depth. Figures 3(b) and 3(c) demonstrate that the number of clusters is consistently higher in the deeper layers compared to the lower layers, with inter-cluster distances escalating significantly in the last two layers and reaching their maximum in the final layer. This pattern indicates that clusters become increasingly distinct and better separated as they progress through the network's layers.

The enhanced separation of clusters in deeper layers can be attributed to the hierarchical feature extraction inherent in deep neural networks. As data moves through successive layers, the network abstracts and compiles more complex features, transitioning from general to more specific attributes. This hierarchical processing allows the final layers to capture and enhance subtle distinctions between different data groups, leading to more defined and isolated clusters. This process not only underscores



(a) (b)

Figure 4: a) Overview of the Cluster-guided Sparse Expert (CSE) layer. b) The cluster number fluctuation is mainly caused by the big common cluster. These four figures arranged sequentially from top to bottom, were sampled at every 10,000 steps throughout the process from the FFN of the 10-th layer in a GPT model.

the capability of deep layers to refine and emphasize key features but also illustrates the network's efficiency in encoding progressively finer-grained information as layer depth increases.

3 Clsuter-guided Sparse Expert (CSE)

To avoid the troublesome and costly domain finetuning, we design a novel strategy, named **C**lsuter-guided **S**parse **E**xpert (CSE), to help the model capture the long-tail domain knowledge during pretraining. Since long-tail domain data show poor gradient consistency with overall data, we employ a sparse expert architecture within the Transformer model to assign data to different parameters, thereby avoiding the gradient conflict in each parameter group. This strategy can be applied on either attention or FFN. To dispatch data, with a straightforward and generally accepted intuition such that data with similar semantic meaning are closer in the embedding space, we design a very simple, efficient but effective online clustering algorithm operating concurrently with the language model pretraining, separate embeddings into different clusters, and use the outcome of this algorithm to instruct the dispatching of embeddings. The proposed algorithm is outlined in Algorithm 1.

Dimension Reduction In high-dimensional vector clustering, computational efficiency poses a significant challenge due to the $O(d^2)$ complexity of computing vector distance where d denotes the dimensionality. So, we employ the same way of dimension reduction as is discussed in Section 2 before applying clustering on the embeddings, using a Gaussian random initialized matrix to project embeddings to a low-dimensional space[23]. This process, grounded in the Johnson-Lindenstrauss Lemma, effectively preserves the pairwise distances between embeddings while reducing their dimensionality, thereby enhancing the efficiency of our clustering algorithm.

Algorithm 1 Cluster-guided Sparse Expert

Require: w: Warm-up step count
Require: N: Initialization data count
Require: M: Gaussian random matrix $\in \mathbb{R}^{d \times d'}$ for reducing dimension
Require: S: Incoming embedding stream
Require: α: center update factor

1: Wait w steps till the warm-up end.
2: Sample N data and run a clustering algorithm. Initialize cluster structure with the outcome by recording the cluster center c_i and radius r_i for each cluster.
3: **for** v in S **do**
4: $v' = Mv$
5: $i = \arg\min_{j=1}^{C} ||v' - c_j||/r_j$
6: Dispatch v to parameter group i
7: $c_i = \alpha c_i + (1 - \alpha)v'$
8: **end for**

Initialization We commence by training a baseline dense model devoid of any expert structure. Our findings in Section 2 illuminate an initial rise in gradient consistency between long-tail domain data and the general dataset at the onset of training, subsequently followed by a downturn. Consequently, we adopt a warm-up stage, letting the model learn the common features of long-tail and non-long-tail data. In our experiments, this process typically accounts for no more than 1% of the overall training. We then sample N instances from the dataset and use its clustering result to initialize the cluster structure. We utilized DBSCAN [30] in our experiments, a clustering algorithm that does not explicitly require the number of clusters. For every identified cluster, we document its centroid and define its radius as the average distance of all constituent data points from this central point

After this warm-up period, we fix the number of clusters and copy the module into cluster number copies. The module selection is introduced in the next paragraph. In our experiments, we noticed that the variations in the number of clusters were primarily driven by the splitting and merging of larger clusters, as illustrated in Figure 4(b); the smaller, long-tail clusters, however, remained largely unchanged. Consequently, adopting the initial clustering configuration directly, without further adjustments during training, was found to have no detrimental effect on model performance or the distribution of data handling. This approach capitalizes on the stability of the long-tail clusters and the dynamics of the larger ones, ensuring efficient data processing without compromising accuracy.

Select Layer Our motivation for performing clustering is rooted in the premise that semantically similar data tends to be closer. However, it is important to note that models learn the semantics of data progressively through layers; as we delve deeper into the model layers, the semantic information becomes increasingly rich, which may in turn amplify distinctions between data points. To quantify this variation, we apply our strategy only on layers with larger inter-cluster distance. Since the last 2 layers show a significant increase in inter-cluster distance, we apply our strategy in the last 2 layers, which is also the empirical best practice observed in existing moe-related works.[11, 26]

Dispatch Embeddings For each coming embedding, we decide the index of the expert it is dispatched to with $i = \arg\min_{j=1}^{n} ||v' - c_j||/r_j$, where c_j denotes the center of cluster j, and r_j denotes the radius of cluster j. Note that the v' here is the sequence embedding rather than a token embedding and is defined as the mean of all token embeddings in the sequence[17], and the dispatching also happens on the sequence level.

Update cluster center The model's parameter space undergoes gradual updates throughout training, causing a slow drift in the embedding space as the parameters evolve. To tackle this, we incorporate a dynamic mechanism to update the cluster centers concurrently with the assignment of clusters. For a given cluster mc_i, let its center at time t be denoted as c_i^t. When a new embedding v arrives and is assigned to mc_i, we update c_i^t with: $c_i^{t+1} = \alpha \cdot c_i^t + (1 - \alpha) \cdot v'$, where, $\alpha \in [0, 1]$ is a center update factor that determines the influence of the new embedding v' on the existing center c_i^t. This adaptive updating scheme ensures that cluster centers remain representative of the current state of the embedding space, even as it evolves through the training process.

4 Related Works

Long-Tail Prior research addressing the issue of long-tail learning has predominantly been conducted within the domain of computer vision. The objective is to accurately recognize and classify rare or infrequently occurring classes in a given dataset together with frequently occurring classes [43]. There are several approaches to address the problem, including re-weighting [8], logit adjustment [4, 44], robust distributional matching [18, 35], and knowledge transfer [38, 34]. [37] declare that as the number of samples increases, the diminishing phenomenon suggests that there is a decreasing marginal benefit for a model to extract additional information from the data due to the presence of information overlap. Research in natural language processing has identified significant limitations in language models' capacity to learn long-tail knowledge [28, 3]. Furthermore, [45] suggests that attempting to address this issue during the finetuning stage is often too late.

Domain-Specific Finetuning Domain-specific finetuning, also known as domain-specific pretraining, is highly advantageous to assist language models in requiring specialized domain knowledge. In one approach, contextualized embeddings are adapted to text from the target domain using masked language modeling, as detailed by Han and Eisenstein [16]. The concept of multi-phase pretraining involves secondary-stage unsupervised pretraining, exemplified by broad-coverage domain-specific

BERT variants like BioBERT [25]. Research by Gururangan et al. [15] extends this by proposing domain-adaptive pretraining (DAPT) from a broader corpus and task-specific pretraining (TAPT) which uses unlabeled data increasingly aligned with the task distribution. These studies underscore the importance of domain-relevant data for pretraining in both high and low-resource scenarios [16, 15].

5 Experiments

This section presents the experimental results of our model and other methods. In the experiments, our model only undergoes a pretrained phase, reading domain-specific data once. Other methods are pretrained on the same dataset and then finetuned on domain-specific datasets. Subsequently, all models are used as embedding models with all parameters frozen to generate embeddings for downstream tasks.

Dataset and Evaluation We employ Wikipedia [13] as our pretraining dataset, which is also widely accepted in other works [25, 10]. We adopt some legal and medical domain-specific downstream tasks to show the effectiveness of our model. To ensure that the pretraining data do contain domain knowledge required by the downstream tasks, we mixed a relatively small amount (less than 8%) of legal-domain-specific data [1] and medical-domain-specific data [9] into the pretraining data to simulate a long-tail distribution. The datasets selected are listed in Table 3 in Appendix A. Concurrently, we report the test perplexity of each model after the pretraining phase, serving as evidence of model convergence. Task performances are reported by accuracy.

Baselines Since our strategy is not restricted to a specific model structure, we adopt both BERT [10] and GPT [29] as the base models and compare all the strategies on these base models respectively. We also compare with a Switch-MoE [11] version of them to show the effectiveness of our routing strategy. More Detailed implementation setting is listed in Appendix A.

5.1 Main Result

Table 1 and Table 2 shows the performance of all models/strategies under our experiment setting with a trainable linear classifier for downstream tasks. ***/med** means a model finetuned on medical-domain-specific data, and ***/legal** means a model finetuned on legal-domain-specific data. We tested Clsuter-guided Sparse Expert on Attention and FFN respectively, denoted as **MoA** and **MoF**.

Our method outperforms other models/strategies on almost all tasks, with an average improvement of around 3%, showing an ability to learn long-tail data from the pretraining dataset. Our method can be applied to either the Attention module or the FFN module, and both way will yield a better result compared with the finetuned baselines, showing a potential for eliminating the need for domain finetuning. While in certain scenarios, domain finetuning remains indispensable due to the privacy concerns associated with proprietary data, we argue that when pretraining datasets encompass domains similar to the proprietary one, our approach can still facilitate an enhanced domain finetuning performance. It is also notable that domain finetuning leads to overfitting and even catastrophic forgetting, resulting in a decrease in performance on tasks from non-related domains. More details are shown in Appendix A.

Table 1: Results of strategies applied on BERT

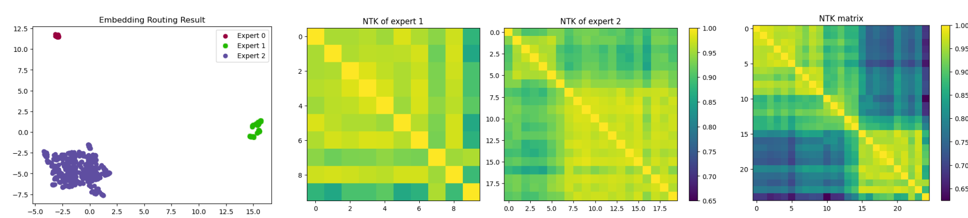
Models	Pretrain ppl	Overruling	Casehold	GAD	EUADR	SST2	average
BERT/med	37.00	<u>86.67</u>	50.51	67.09	84.23	<u>66.86</u>	71.07 ± 0.22
BERT/legal	37.00	86.67	<u>50.93</u>	66.83	84.79	65.14	70.87 ± 0.23
MoE/med	31.00	85.00	50.49	64.52	83.10	64.79	69.58 ± 0.20
MoE/lgeal	31.00	85.83	50.30	64.32	84.79	63.88	69.82 ± 0.19
Ours/MoA	28.25	86.62	**50.94**	**72.90**	<u>90.09</u>	66.60	<u>73.43 ± 0.18</u>
Ours/MoF	34.64	**89.10**	50.82	<u>71.65</u>	**91.23**	**67.98**	$\mathbf{74.16 \pm 0.20}$

Table 2: Results of strategies applied on GPT

Models	Pretrain ppl	Overruling	Casehold	GAD	EUADR	SST2	average
GPT/med	55.59	88.33	49.82	71.56	84.23	73.90	73.57 ± 0.17
GPT/legal	55.59	89.17	50.58	71.69	81.69	74.50	73.53 ± 0.23
MoE/med	40.69	91.25	50.11	72.77	83.38	72.03	73.91 ± 0.12
MoE/legal	40.69	91.60	49.68	72.66	83.38	71.97	73.86 ± 0.23
Ours/MoA	42.99	91.68	50.70	71.75	**85.91**	74.61	74.93 ± 0.08
Ours/MoF	43.38	**93.33**	**51.26**	**73.30**	85.63	**76.00**	$\mathbf{75.90 \pm 0.19}$

5.2 Analysis

Expert analysis We analyze our model's embedding space to determine if our method dispatches embeddings correctly. We sample data and perform a forward inference pass through the model, visualizing the dispatching path of our model. As is shown in Figure 5, our distribution strategy correctly and effectively dispatches data from different long-tail clusters to different experts. We further visualize the NTK in each expert of our model, and it can be observed that by dispatching long-tail data separately, the NTK in each expert becomes more consistent. Whereas in a baseline model, its NTK matrix shows a poor consistency of the batch data, since long-tail and non-long-tail data are not separated.



(a) (b) (c)

Figure 5: a) The embedding space and routing result of our model. b) The NTK in each expert in our model. c) The NTK in baseline. b) and c) are sampled from the FFN in the 10th layer.

Overhead Analysis For our method, the warm-up phase incurs no additional computation. At the end of the warm-up, The clustering algorithms are bounded by their worst-case time complexity $O(N^2 d')$, thus their impact on the total FLOPs compared to the whole pretraining is negligible when posed against the extensive computations involved in the model's forward-backward passes. Our dispatching strategy introduces $O(Cd'^2)$ for comparing distance with each cluster, which is also negligible. For baseline methods, since they all undergo a finetuning stage, they introduce an additional 5% computation compared to the pretraining stage under our experimental settings.

6 Conclusion

In this paper, we seek to elucidate why language models require domain finetuning despite the presence of domain knowledge in their pretraining data. Our investigation uncovers that Sentences with lower frequency scores show diminished gradient consistency, resulting in increased test perplexity. This misalignment, particularly pronounced in low-frequency sentences, culminates in elevated test perplexity, suggesting a deficiency in effectively leveraging domain-specific information. To address this challenge, we introduce Cluster-guided Sparse Experts (CSE), grouping diverse long-tail domain data and dispatching them to different experts to enhance gradient consistency within each expert, thereby enabling the model to incorporate long-tail domain knowledge during pretraining. Experiments suggest that our approach has the potential to supplant the need for a dedicated domain finetuning stage. Through this approach, long-tail domain instances promote the formation of small, outlier clusters in the representation space, exhibiting a characteristic signature across varying stages of training and architectural depths.

References

[1] Caselaw access project. `https://case.law/`,, 2024.

[2] Reddit comments dataset. `https://clickhouse.com/docs/en/getting-started/example-datasets/reddit-comments`,, 2024.

[3] Mallen A., Asai A., Zhong V, Das R., Khashabi D., and Hajishirzi H. When not to trust language models: Investigating effect. *in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[4] Krishna Menon Aditya, Jayasumana Sadeep, Singh Rawat Ankit, Jain Himanshu, Veit Andreas, and Kumar. Sanjiv. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[5] Benjamin Bowman and Guido Montúfar. Spectral bias outside the training set for deep networks in the kernel regime. *ArXiv*, abs/2206.02927, 2022. URL `https://api.semanticscholar.org/CorpusID:249431476`.

[6] Àlex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I Furlong. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC Bioinformatics*, 16(1), February 2015. doi: 10.1186/s12859-015-0472-9. URL `https://doi.org/10.1186/s12859-015-0472-9`.

[7] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: "preparing the muppets for court"'. *ArXiv*, abs/2010.02559, 2020. URL `https://api.semanticscholar.org/CorpusID:222141043`.

[8] Huang Chen, Li Yining, Change Loy Chen, and Tang Xiaoou. Learning deep representation for imbalanced classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[9] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097. URL `https://aclanthology.org/N18-2097`.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] W. Fedus, B. Zoph, , and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.

[12] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M. Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/405075699f065e43581f27d67bb68478-Abstract.html`.

[13] Wikimedia Foundation. Wikimedia downloads. URL `https://dumps.wikimedia.org`.

[14] Zhen Guo and Yining Hua. Continuous training and fine-tuning for domain-specific language models in medical question answering. *ArXiv*, abs/2311.00204, 2023. URL `https://api.semanticscholar.org/CorpusID:264832958`.

[15] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. *ArXiv*, abs/2004.10964, 2020. URL `https://api.semanticscholar.org/CorpusID:216080466`.

[16] Xiaochuang Han and Jacob Eisenstein. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://api.semanticscholar.org/CorpusID:202541481.

[17] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. Whiteningbert: An easy unsupervised sentence embedding approach. *arXiv preprint arXiv:2104.01767*, 2021.

[18] Zheng Huangjie, Chen Xu, Yao Jiangchao, Yang Hongxia, Li Chunyuan, Zhang Ya, Zhang Hao, Tsang Ivor, Zhou Jingren, and Zhou. Mingyuan. Contrastive attraction and contrastive repulsion for representation learning. *Transactions on Machine Learning Research*, 2023.

[19] Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

[20] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.*, 31:651–666, 2008. URL https://api.semanticscholar.org/CorpusID:11152703.

[21] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pp. 15696–15707. PMLR, 2023.

[22] Zixuan Ke, Yijia Shao, Haowei Lin, Hu Xu, Lei Shu, and Bing Liu. Adapting a language model while preserving its general knowledge. *arXiv preprint arXiv:2301.08986*, 2023.

[23] Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. *arXiv preprint arXiv:1411.2404*, 2014.

[24] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234 – 1240, 2019. URL https://api.semanticscholar.org/CorpusID:59291975.

[25] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[26] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *In International Conference on Learning Representations*, 2021.

[27] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *ArXiv*, abs/2308.08747, 2023. URL https://api.semanticscholar.org/CorpusID:261031244.

[28] Kandpal N., Deng H., Roberts A., Wallace E., and Raffel C. Large language models struggle to learn long-tail knowledge. *In International Conference on Machine Learning*, 2023.

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[30] Erich Schubert, Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited. *ACM Transactions on Database Systems (TODS)*, 42:1 – 21, 2017. URL https://api.semanticscholar.org/CorpusID:5156876.

[31] K. Singhal, Shekoofeh Azizi, Tao Tu, Said Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather J. Cole-Lewis, Stephen J. Pfohl, P A Payne, Martin G. Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, P. A. Mansfield, Blaise Agüera y Arcas, Dale R. Webster, Greg S. Corrado, Yossi Matias, Katherine Hui-Ling Chou, Juraj Gottweis, Nenad Tomaev, Yun Liu, Alvin Rajkomar, Joëlle K. Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge. *Nature*, 620:172 – 180, 2022. URL `https://api.semanticscholar.org/CorpusID:255124952`.

[32] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D13-1170`.

[33] Erik M. van Mulligen, Annie Fourrier-Reglat, David Gurwitz, Mariam Molokhia, Ainhoa Nieto, Gianluca Trifiro, Jan A. Kors, and Laura I. Furlong. The eu-adr corpus: Annotated drugs, diseases, targets, and their relationships. *Journal of Biomedical Informatics*, 45(5): 879–884, 2012. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2012.04.004. URL `https://www.sciencedirect.com/science/article/pii/S1532046412000573`. Text Mining and Natural Language Processing in Pharmacogenomics.

[34] Chen Xu, Chen Siheng, Yao Jiangchao, Zheng Huangjie, Zhang Ya, and W. Tsang. Ivor. Learning on attribute-missing graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[35] Chen Xu, Pan Yuangang, Tsang Ivor, and Zhang. Ya. Learning node representations against perturbations. *Pattern Recognition, 145:109976*, 2024.

[36] Haoran Yang, Yumeng Zhang, Jiaqi Xu, Hongyuan Lu, Pheng Ann Heng, and Wai Lam. Unveiling the generalization power of fine-tuned large language models. *ArXiv*, abs/2403.09162, 2024. URL `https://api.semanticscholar.org/CorpusID:268385476`.

[37] Cui Yin, Jia Menglin, Lin Tsung-Yi, Song Yang, and Belongie. Serge. Class-balanced loss based on effective number of samples. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[38] Wang Yu-Xiong, Ramanan Deva, and Hebert. Martial. Learning to model the tail. *Advances in Neural Information Processing Systems*, 2017.

[39] Shi Yubin, Chen Yixuan, Dong Mingzhi, Yang Xiaochen, Li Dongsheng, Wang Yujiang, P. Dick Robert, Lv Qin, Zhao Yingying, Yang Fan, Lu Tun, Gu Ning, and Shang Li. Train faster, perform better: Modular adaptive training in over-parameterized models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[40] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. A survey of large language models. *ArXiv*, abs/2303.18223, 2023. URL `https://api.semanticscholar.org/CorpusID:257900969`.

[41] Jiawei Zheng, Hanghai Hong, Xiaoli Wang, Jingsong Su, Yonggui Liang, and Shikai Wu. Fine-tuning large language models for domain-specific machine translation. *ArXiv*, abs/2402.15061, 2024. URL `https://api.semanticscholar.org/CorpusID:267897581`.

[42] Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset. *arXiv e-prints*, art. arXiv:2104.08671, April 2021. doi: 10.48550/arXiv.2104.08671.

[43] Zhou Zhihan, Yao Jiangchao, Wang Yan-Feng, Han Bo, and Zhang Ya. Contrastive learning with boosted memorization. *In International Conference on Machine Learning*, 2022.

[44] Zhou Zhihan, Yao Jiangchao, Hong Feng, Zhang Ya, Han Bo, and Wang. Yanfeng. Combating representation learning disparity with geometric harmonization. *In Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[45] Zeyuan Allen Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023.

Appendix

A Experiments

Table 3 shows the datasets used in our experiments. Table 4 shows the hyperparameters used in our implementations. We use a machine with 8 NVIDIA GeForce RTX 3090 GPUs with 24GB GPU memory as our experiment platform. Pretraining costs about 24 hours.

Table 3: Datasets used for experiments

Pretraining dataset	Description
Wikipedia ([13])	Wikipedia dataset containing cleaned articles of all languages. The datasets are built from the Wikipedia dump with one split per language. Each example contains the content of one full Wikipedia article with cleaning to strip markdown and unwanted sections.
legal([1])	In collaboration with Ravel Law, Harvard Law Library digitized over 40 million U.S. court decisions consisting of 6.7 million cases from the last 360 years into a dataset that is widely accessible to use.
PubMed([9])	PubMed comprises more than 36 million citations for biomedical literature from MEDLINE, life science journals, and online books.
Finetuning task	**Description**
Overruling ([42])	A law dataset corresponds to the task of determining when a sentence is overruling a prior decision.
Casehold([42])	Case Holdings On Legal Decisions, comprising over 53,000+ multiple choice questions to identify the relevant holding of a cited case.
GAD([6])	A relation extraction dataset, to decide if a gene is related to a specific disease.
EUADR([33])	Another relation extraction dataset, to decide if a gene is related to a specific disease.
SST2([32])	The Stanford Sentiment Treebank consists of sentences from movie reviews and human annotations of their sentiment.

Table 4: Hyperparameters of Models

Hyperparameters	BERT-based	GPT-based
FFN modules	4	6
Attention modules	4	6
attention heads	8	12
our-strategy-based layers	2	2
transformer layers	12	12
Hidden dimension size	768	768
Droupt	0.1	0.1
Attention dropout	0.1	0.1
Sequence length	128	256
Batch size	100	64
Max steps	36k	300k
Learning rate decay	Cosine	Cosine
random seed used	14, 24	22, 80

By monitoring the validation loss of the pretraining dataset(Figure 6), we show the Catastrophic Forgetting problem of the BERT model and its MOE method in the domain-specific finetuning phase. Despite our attempts at various combinations of generic data and domain-specific data during domain finetuning, the best outcome among these still resulted in a decline in model performance on domains unrelated to its fine-tuning, indicating a limitation in the generalizability of the adapted model. As domain-specific finetuning proceeds, the validation loss of pretraining dataset has a significant rise and stays well above the convergence position of pretraining.

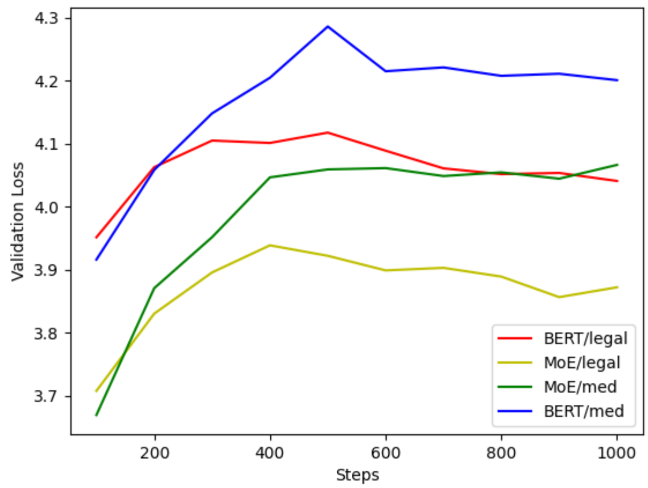


Figure 6: The validation loss of the pretraining dataset during the domain-specific finetuning phase.

B Limitations Discussions

Although we use the method of mixing small-scale domain-specific datasets into pretraining data to simulate the long-tail distribution in those huge corpora, we cannot fully simulate the extremely rich pretraining data used on LLMs due to the limited training resources.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim clearly in Abstract and Introduction 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We write them in Appendix B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We achieved this part in the Analysis 2, and the relevant assumptions are indicated in the Reference.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The Table 4 in the Appendix A shows the random seeds used in our experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release our source code and detailed instructions for reproducing our results upon acceptance of this paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We show our experimental setting/details both in our Experiments 5 and in the Appendix A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We show the standard deviation of accuracy in Table 1 and Table 2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We show these in Appendix A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have checked it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the creators and introduce assets in the Appendix A.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.