Multi-Stage Balanced Distillation: Addressing Long-Tail Challenges in Sequence-Level Knowledge Distillation

Anonymous ACL submission

Abstract

Large language models (LLMs) have significantly advanced various natural language processing tasks, but deploying them remains computationally expensive. Knowledge distillation (KD) is a promising solution, enabling the transfer of capabilities from larger teacher LLMs to more compact student models. Particularly, sequence-level KD, which distills rationale-based reasoning processes instead of merely final outcomes, shows great potential in enhancing students' reasoning capabilities. However, current methods struggle with sequence-level KD under long-tailed data distributions, adversely affecting generalization on sparsely represented domains. We introduce the Multi-Stage Balanced Distillation (BalDistill) framework, which iteratively balances training data within a fixed computational budget. By dynamically selecting representative head domain examples and synthesizing tail domain examples, BalDistill achieves state-of-the-art performance across diverse long-tailed datasets, enhancing both the efficiency and efficacy of the distilled models.

1 Introduction

Large language models (LLMs) like GPT-4 and LLaMA have revolutionized tasks ranging from text generation to language translation through their deep understanding and generation of human-like text (OpenAI, 2023; Touvron et al., 2023; Chiang et al., 2023; Jiang et al., 2023). Despite their success, the deployment of these models is hindered by their substantial size and computational demands, especially in environments with limited resources. Knowledge distillation (KD) offers a viable solution by transferring knowledge from expensive teacher models to smaller, efficient student models. Specifically, *sequence-level KD* focuses on distilling rationale-based reasoning processes rather than final outcomes. It leverages the teacher's reasoning processes, encapsulated in chain-of-thought (CoT) rationales, to enhance the student models' generative capabilities (Kim and Rush, 2016; Ho et al., 2022; Shridhar et al., 2022; Hsieh et al., 2023).

However, there are a few challenges to fully leverage the power of sequence-level KD, as follows. **(C1)** Sequence-level KD encounters significant challenges when training with long-tailed data distributions, which are prevalent in real-world scenarios — data often follows a power-law distribution with a few dominant classes (head) and many rare classes (tail) (Liu et al., 2019). Such distributions feature a few dominant classes and many underrepresented ones, leading to models that generalize poorly on sparsely represented domains. **(C2)** Traditional KD methods in the text area to solve long-tail challenges, often reliant on direct access to model weights or loss adjustment primarily suited for straightforward classification tasks (Zhou et al., 2023; Schick and Schütze, 2021; Dai et al., 2023; Zhang et al., 2022; Tepper et al., 2020), falter under the complexities of sequence-level KD, especially when the teacher model is a black box and the task is generative, which is our target. **(C3)** Addressing this imbalance is critical, yet resource-intensive, as it typically requires generating a large volume of synthetic data to balance the dataset Tepper et al. (2020). Moreover, naively up-sampling the long-tailed dataset may dramatically increase the number of calls to the teacher models. Budget constraints play a crucial role in KD for black-box LLMs, as querying the teacher for rationales can be costly and time-consuming (Chen et al., 2023; Zhou and Ai, 2024).

Our proposed solution, the Multi-Stage Balanced Distillation (BalDistill), tackles all the challenges above by strategically generating balanced training sets within budget constraints and iteratively fine-tuning the student model with actively selected and synthetic data for multiple stages. BalDistill progressively refines the training data by selecting key examples from well-represented

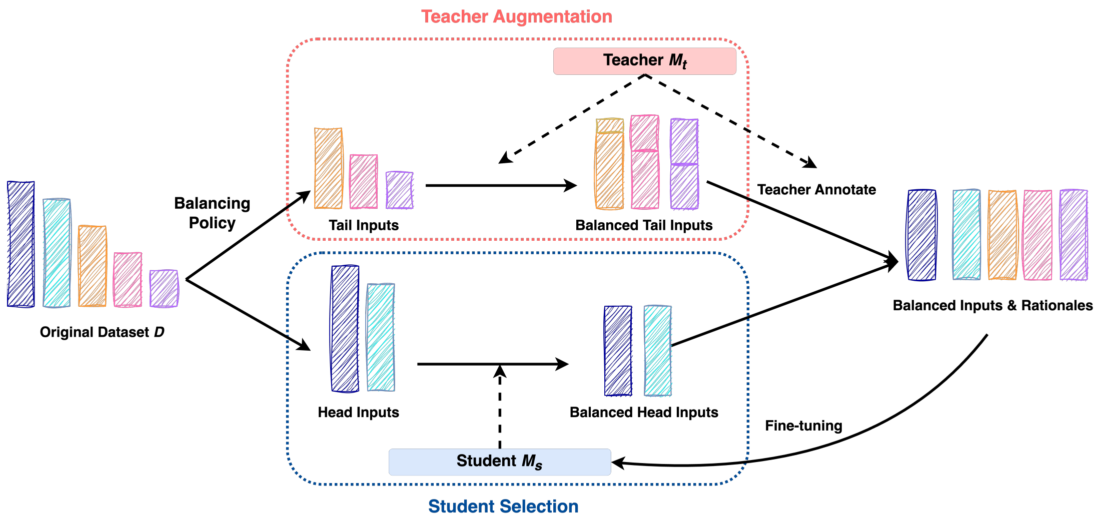


Figure 1: **Overview of the proposed iterative BalDistill framework.** The framework is composed of multiple stages. For each stage, we apply the balancing policy to decide the data distribution in the training batch. For head domains with sufficient data, we actively extract the examples by IFD metrics using the student model. For the tail domains, we call the teacher model to generate the synthetic examples and the corresponding rationales. The teacher model finally annotates the balanced training batch and fine-tunes the student model.

domains and generating necessary synthetic data for underrepresented ones, ensuring comprehensive domain coverage and model robustness. By dynamically selecting representative head domain examples and synthesizing tail domain examples, BalDistill achieves state-of-the-art (SoTA) performance on various long-tailed datasets, enhancing both the efficiency and efficacy of the method.

Our contributions are summarized as follows:

- **Innovative Problem Framing:** We address the under-explored challenge of applying sequence-level KD to long-tailed distributions, where the teacher model is a black-box LLM.

- **Strategic Framework:** BalDistill innovatively combines active example selection with synthetic data generation for multiple stages to maintain training balance within predefined budget limits.

- **SoTA Performance:** Our framework demonstrably improves the student models' effectiveness and robustness across diverse domains, setting new benchmarks in performance. We empirically demonstrate that our distilled student models achieve state-of-the-art performance across a range of benchmark datasets.

2 Related Work

Knowledge Distillation uses the outputs of a larger LLM (Teacher), such as ChatGPT (OpenAI, 2023), to train a smaller model (Student), such as LLaMa-7B (Touvron et al., 2023). For details of knowledge distillation (KD) of large language models, we refer to the survey for more details (Xu et al., 2024). In this work, we focus on KD with black-box teacher models. There are two lines of work with respect to knowledge distillation. The first is to ask teacher models to generate the final answers and to fine-tune on the final answers (Zhou et al., 2023; Schick and Schütze, 2021). Another line of work asks teacher models to generate rationales at the reasoning process and fine-tunes student models on the rationales in the sequence level to improve their reasoning ability (Ho et al., 2022; Shridhar et al., 2022; Hsieh et al., 2023), which proves to be more effective. In this work, we mainly discuss using a teacher model to generate rationales and improve the student's reasoning ability on a long-tailed dataset. Despite the progress of KD in the LLM era, existing works fail to establish a pipeline to gain knowledge from long-tailed datasets with the sequence-level KD, as few rationale examples are provided for tail knowledge.

Long-Tail Learning focuses on long-tail distributed data and has been an emerging topic of interest in the NLP community (Liu et al., 2019; Wang et al., 2017; Godbole and Jia, 2022; Dai et al., 2023; Zhang et al., 2022). Approaches to solving the long-tail problem include rebalancing, information augmentation, and module improvement (Zhang et al., 2021; He et al., 2021; Cui et al., 2021). Despite the importance of long-tail learning, studies have shown that LLMs struggle to learn long-tail knowledge (Kandpal et al., 2023; Sun et al., 2023). In this work, we propose to improve LLMs' ability to learn long-tail knowledge via multi-stage distillation over balanced datasets.

Active Learning aims to reduce labeling effort by selecting only the most useful examples. Traditional active learning can be categorized into uncertainty-based methods (Prabhu et al., 2019; Margatina et al., 2021) and diversity-based methods (Ru et al., 2020; Ash et al., 2019). In the LLM era, active learning has been used to reduce human annotation costs by (1) strategically selecting the most informative examples for human feedback or annotation (Margatina et al., 2023; Osband et al., 2022; Wang et al., 2020) and (2) integrating language models as annotators within an active learning framework without human supervision (Xiao et al., 2023; Rouzegar and Makrehchi, 2024; Zhang et al., 2023). In this work, we propose to solve the long-tail problem in the student LLM by actively distilling knowledge from a black-box teacher LLM to meet the budget requirement.

3 Methodology

3.1 Problem Statement

We define our research problem as follows: Given the teacher LLM ($\mathcal{M}_t$), the student LLM ($\mathcal{M}_s$), a long-tailed dataset $\mathcal{D}$ (with domain number $[d_1, d_2, \dots, d_l]$ for l domains in total) and a fixed budget B to query the teacher, we seek to propose an efficient framework to fine-tune an effective and robust student model, $\mathcal{M}_s$, over $\mathcal{D}$.

3.2 Overall Approach

To mitigate the performance bias in KD caused by long-tailed datasets within budget constraints, we employ a strategy that combines synthetic data augmentation with active selection. This approach ensures effective fine-tuning across both well-represented ('head') and underrepresented ('tail') domains. As depicted in Figure 1, we propose a *multi-stage* framework to create the training data *iteratively*. We operate in a pool-based setting where a large dataset, denoted as $\mathcal{D}$, is available but lacks annotations from a teacher model.

At each stage of our BalDistill process, we first implement a balancing policy, which we have designed, to determine the appropriate data distribution for each domain within the training batch. This policy is based on the principles of data equality and training effectiveness across domains, aiming to optimize learning outcomes despite data scarcity in certain areas. The total number of stages is predefined based on the consideration of efficiency and the optimal performance.

For domains well-represented in our dataset $\mathcal{D}$ (referred to as 'head domains'), we employ active selection techniques (Touvron et al., 2023; Yuan et al., 2020) using the fine-tuned student model $\mathcal{M}_s$ to identify and extract the most informative examples from the pool. Conversely, for domains lacking sufficient data ('tail domains'), we utilize the teacher model $\mathcal{M}_t$ to generate both synthetic samples and corresponding annotations, enriching the training material available.

After selecting and/or generating these samples, we query the teacher model to provide detailed rationales for examples in the training batch. These annotated examples are then used to fine-tune the student model $\mathcal{M}_s$ in preparation for the next stage. Detailed descriptions of these components, along with the algorithms outlining this procedure, are presented in Algorithm 1 in Appendix B.

3.3 Balancing Policy

Considering K total stages in our framework, we first evenly divide our budget B into K parts, which means that for each stage, we create a small training batch with $\frac{B}{K}$ examples extracted from $\mathcal{D}$ with teacher-annotated rationales. Within a small training batch, we propose two strategies to allocate the budget over different domains.

Naive Balancing Since our goal is to mitigate the bias towards head domains, our first balancing policy is to use naive balancing, which selects the same number of inputs for each domain in the training batch. Formally, the number of samples for each domain in the small training batch is $\frac{B}{Kl}$, where l is the number of domains in the dataset.

Adaptive Balancing One of our staged learning framework's key features is utilizing the fine-tuned student model to actively select representative inputs from well-represented domains, known as head domains. However, employing a naive balancing policy typically results in the disproportionate allocation of the training budget to data from underrepresented domains, or tail domains. This training batch may lead the fine-tuned student model to struggle to select truly effective examples from the head domains, particularly in the initial stages. Such selections are crucial for the model to learn effectively from these domains. To address this, we implement an adaptive balancing policy. This policy starts by constructing the training batch with a distribution akin to random selection, thus primarily focusing on head data in the early stages to 'warm up' the model. As the process advances, the policy gradually shifts towards a more balanced distribution by the final stage, ensuring comprehensive learning across both head and tail domains.

Formally, the number of examples for each domain is the weighted average between the numbers for random selection and the numbers for naive balancing. For stage i, domain d, we select $(\frac{n_d}{N} \cdot \frac{B}{K}) \cdot \frac{K-i}{K} + \frac{B}{Kl} \cdot \frac{i}{K}$ examples for domain d to build the training batch for adaptive balancing, where N and n_d are the total number and the domain size in the original data $\mathcal{D}$.

Then, domains are naturally categorized based on whether the number of required samples per domain exceeds the available samples in the pool. Domains requiring more samples than available are designated as 'head domains' for that particular stage, while those with fewer required examples than available are categorized as 'tail domains.'

For tail domains, where there are insufficient samples in the dataset $\mathcal{D}$, we rely on the teacher model to generate both the samples and their corresponding rationales, detailed in Section 3.4. In contrast, for head domains, which have a sufficient number of samples available to meet the demands of the training batch, we utilize the fine-tuned student model to actively select the most representative samples, as discussed in Section 3.5.

It is important to note that the classification of domains as head or tail can vary across different stages of the training process, depending on the evolving needs and data availability.

3.4 Teacher Data Augmentation

Motivated by the effectiveness of synthetic dataset generated by black-box LLMs (OpenAI, 2023; Radford et al., 2019; Zhou et al., 2024), we utilize the teacher LLMs to generate synthetic samples and corresponding annotations to upsample data for tail domains. To save the annotation budget, we require the teacher model to compose the sample and the corresponding rationales at the same time.

Suppose that we need m synthetic examples for domain a to satisfy the training batch requirement. Given an instruction following prompt P_c, composed of three demonstrations from domain a, and teacher model $\mathcal{M}_t$, we employ stochastic temperature sampling with a fixed temperature and repeat the process m times with generated samples $\hat{x}_{a1}, \cdots \hat{x}_{am}$ and rationales $\hat{y}_{a1}, \cdots \hat{y}_{am}$:

$$\hat{x}_{ai}, \hat{y}_{ai} = M_t(P_c, a) \quad \text{for } i \in \{1, \cdots, m\}$$

Then we add the generated samples and rationales to the training batch and combine with the extracted samples from $\mathcal{D}$. We present two examples of synthetic inputs and rationales from the teacher model in Table 9 in Appendix B. The case study suggests the effectiveness of the teacher model in generating tail examples.

3.5 Student Active Selection

For head domains, our strategy involves actively selecting instances from the original dataset to meet the numeric requirements of the balancing policy. We aim to mitigate information loss from data downsampling through this active data acquisition. The objective is to identify the most challenging or uncertain instances for the student model, thereby optimizing its learning trajectory.

To quantify instance uncertainty, we adapt the Instruction Following Difficulty (IFD) metric originally proposed by Li et al. (2024a,b). The IFD scores are used to measure a training instance's uncertainty level as perceived by the student model. IFD is calculated as the ratio of the perplexity of generating a response y with an input x to the perplexity of generating y without x: $\text{IFD}(x, y) = \frac{\text{PPL}(y|x)}{\text{PPL}(y)}$, where PPL represents perplexity, a metric widely used to evaluate language model performance (Jelinek et al., 1977). Studies have shown that IFD scores offer greater efficiency in data selection compared to methods like K-means diversity or sole reliance on perplexity (Li et al., 2024a; Settles, 2009; Yuan et al., 2020).

A higher IFD score indicates an increased difficulty for the model in generating the response, highlighting the instance's value for training (Li et al., 2024a). Unlike the approach in Li et al. (2024a), which utilizes ground-truth or advanced LLM-generated responses y, our setting imposes budge constraints that prevent such usage. Instead, we calculate IFD using rationals $\hat{y}_s$ generated by the previously fine-tuned student model, allowing us to assess the model's self-uncertainty and conserve the annotation budget from the teacher model.

At last, we rank the inputs by their IFD scores, selecting those with the highest values to include in the bath, as specified by the balancing policy.

3.6 Reasoning Generation and Fine-tuning

Building on methodologies from prior research that focus on distilling reasoning abilities from black-box LLMs (Ho et al., 2022; Hsieh et al., 2023), we employ a zero-shot CoT approach, where the teacher model is prompted to generate a reasoning explanation $\hat{y}_t$ for the samples in our constructed training batch. This zero-shot setting is crucial for demonstrating the model's ability to reason based on its pre-existing knowledge alone (Brown et al., 2020). In our experimental setup, which utilizes labeled datasets lacking rationale annotations, the final ground truth answer is included in the prompt. This inclusion ensures that the generated explanations are aligned with the correct outcomes, enhancing the accuracy and relevance of the CoT reasoning. It is important to note that for synthetic samples generated from tail domains in 3.4, we do not perform additional annotations in this part to maintain adherence to budget constraints.

After gathering the required samples and their associated rationales in the training batch, we integrate this batch with the annotated data accumulated from previous stages. This approach ensures that our student model is exposed to a diverse and comprehensive dataset, which helps mitigate the risk of overfitting — a common challenge in machine learning models as identified in prior studies (Dor et al., 2020). To facilitate this, we reinitialize and fine-tune the student model on the compiled rationale sequences from scratch at each stage.

The fine-tuning is performed using autoregressive language modeling with a cross-entropy loss, aligning with the original pre-training objectives of the student model (Touvron et al., 2023).

4 Experiment

Through our extensive empirical analysis, we aim to address the following research questions:

- **RQ1**: How effective is our KD framework compared to previous KD baseline methods?
- **RQ2**: How important is each component (balancing policy and active learning) to the framework?
- **RQ3**: How well does our method perform with different student models and budget restrictions?

Dataset To verify the effectiveness of our framework on various reasoning tasks, we evaluate our method on five long-tailed datasets, following previous work (Yu et al., 2023; Dai et al., 2023; Huang et al., 2021). These include text classification: R52 and Reuters (Hayes and Weinstein, 1990), question answering: AbstractiveQA and Multiple-choiceQA (Dai et al., 2023) and arithmetic: MATH (Hendrycks et al., 2021). For text classification datasets, we treat the label of inputs as the domain; for other datasets, the domain information of inputs is annotated as metadata from the data provider. The detailed construction process and domain information for these datasets can be found in Appendix A. We also show two example distributions of the datasets in Figure 5 in Appendix A. For each dataset, we prepare two budget settings for the experiment. In Table 1, we present the budget number, the test number, the domain number, and the evaluation metric of all five datasets.

Dataset	# Budget	# Test	# Domain	Metric	Task
R52	2,600/5,200	2,570	52	F1	TC
Reuters	4,500/9,000	3,745	90	F1	TC
Abstractive QA	5,000/10,000	10,000	5	F1	QA
Multi-choice QA	5,000/8,000	10,520	10	Accuracy	QA
Math	2,100/3,500	5,000	7	Accuracy	Arithmetic

Table 1: Dataset statistics. TC and QA represent the text classification and question answering, respectively.

Evaluation metrics Since we are dealing with long-tailed imbalanced data, for each dataset, we choose to use both the micro- and macro-averages to evaluate the method robustness (Henning et al., 2022). For the classification datasets (R52 and Reuters), we report micro-F1 and macro-F1, where micro-F1 is a global average F1 score and macro-F1 is computed by taking the unweighted mean of all the per-class F1 scores (Harbecke et al., 2022). For other datasets, we also report the micro-/macro-F1 for AbstractiveQA datasets and micro-/macro-accuracy for Math and Multi-choiceQA datasets. Note that the F1 score for the AbstractQA is the word-level F1 score between the token list of ground truth answer and the generated answer, different from the F1 for the classification task.

Model setup For the teacher model, we use GPT-4 (OpenAI, 2023) to generate the CoT rationales for each dataset. We choose between Llama2-7B and Llama3-8B as our student models (Touvron et al., 2023). We include the detailed configurations and implementations of the model in Appendix B.

Baseline methods We experiment with two variants of our proposed method with different balancing policies, as discussed in Section 3: In our first framework **BalDistill (N)**, we use naive balancing policy, and for second framework **BalDistill**

(A), we leverage adaptive balancing. We compare our framework with multiple baseline methods: (1) **Zero-shot CoT**. We directly prompt the student model to infer on the test data (Kojima et al., 2022). (2) **Random Finetune**. We randomly collect samples from the training data until the budget constraint is met and finetune student models on the final ground-truth labels (Radford et al., 2019). (3) **Random Finetune-CoT**. We randomly collect and use CoT rationales from the teacher model for student fine-tuning (Ho et al., 2022; Yao et al., 2022; He et al., 2023). (4) **Duplicate Finetune-CoT**. We construct the training data with a naive balancing policy, and for tail domains, we duplicate the inputs to satisfy the policy requirement.

5 Results

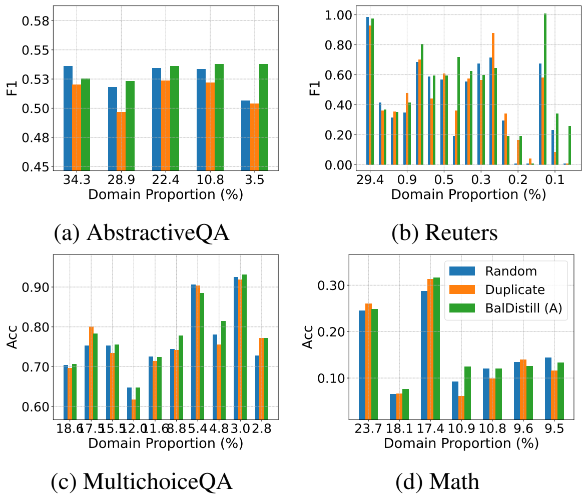
5.1 Comparison with Baseline Methods

BalDistill framework outperforms Random Finetune and Duplicate Finetune methods. We use Llama3 as the student model, GPT-4 as the teacher model, and choose the smaller budget for each dataset in Table 1 as our experiment settings for this subsection. We present the overall macro- and micro-average results of the proposed frameworks and the baseline methods in Table 2. From Table 2, we first observe that on the long-tailed dataset, the methods fine-tuned on teacher-generated rationales (CoT) can significantly outperform the ground-truth fine-tuning method (Random Finetune), which emphasizes the necessity of teacher-generated reasoning steps in the KD.

Among all sequence-level KD methods, our proposed BalDistill (N) and BalDistill (A) achieve the best average performance across various datasets on macro-averages, which obtain an average relative improvement of 2.24% and 6.81%, respectively, compared to the Random Finetune CoT baseline. The performance boost in BalDistill (N) implies the effectiveness of replacing the naive balancing policy with adaptive balancing.

Moreover, we note that the Duplicate Finetune CoT baseline fails to compete with the Random Finetune CoT method in most cases, which indicates that simply duplicating the input from the tail domains to ensure balanced data cannot address the underlying imbalanced data complexity.

To perform a detailed analysis of our framework, we visualize the F1 or accuracy score for each domain of the BalDistill (N) method and two baseline methods (Random Finetune CoT and Duplicate Finetune CoT) in Figure 2, with the x-axis representing the proportion of each domain in the dataset in descending order. From Figure 2, our proposed method can achieve comparable results in the head domains (left side of the figure) but substantially outperform the baseline methods in the tail domains (right side of the figure). This observation verifies our expectation in Section 3, where the balancing policy increases performance in the tail domain, and the active learning part improves the data efficiency to compensate for data loss in the head domain. Note that for Math dataset, BalDistill can only achieve comparable results with the baseline methods on the last two tail domains (precalculus and probability), and we conjecture that the high difficulty in these two domains prevents the teacher from composing high-quality synthetic data.



(a) AbstractiveQA (b) Reuters (c) MultichoiceQA (d) Math

Figure 2: **Performance of proposed method and baselines on different domains.** X-axis represents the proportion of each domain, ranked from head to tail domains. Our proposed BalDistill method can achieve comparable results on head domains and outperform the baseline method on the tail domains.

5.2 Ablation Study

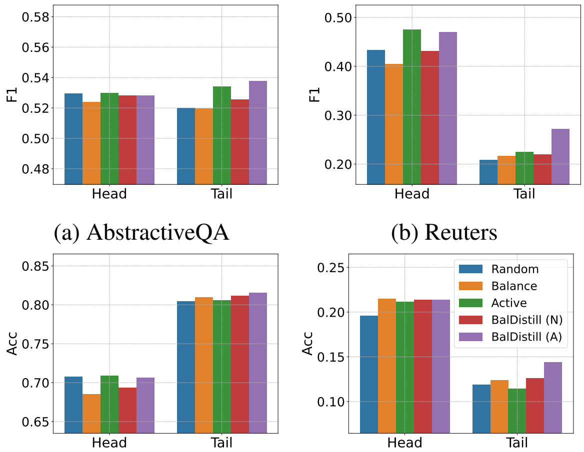
After showing the superiority of our overall framework, our next step is to verify the effectiveness of each component in the proposed method. We compare our framework with the ablated methods: (1) **Balance Finetune CoT**. We adopt a naive balancing policy to construct the training set and query the teacher model to compose inputs in the tail domains. (2) **Active Finetune CoT**. We only keep the active learning component but remove the data augmentation part. The experiment setting is similar to the setup in Section 5.1, and we present the performance of each method with two budget settings

Method	R52		Reuters		AbstractiveQA		Multi-choiceQA		Math	
	macro-f1	micro-f1	macro-f1	micro-f1	macro-f1	micro-f1	macro-acc	micro-acc	macro-acc	micro-acc
Zero-shot CoT	0.89	2.30	0.74	1.61	7.60	7.59	24.67	24.95	7.57	8.68
Random Finetune	45.95	**91.44**	28.01	**74.68**	37.62	37.21	61.23	55.96	10.12	9.48
Random Finetune CoT	**59.70**	89.46	27.35	70.53	52.57	52.88	76.09	74.12	16.62	15.20
Duplicate Finetune CoT	46.56	71.79	26.76	62.84	51.32	51.37	75.92	73.99	16.98	15.05
BalDistill (N)	59.62	82.49	28.09	62.40	52.70	**52.92**	76.60	73.43	17.90	16.34
BalDistill (A)	58.93	87.47	**32.95**	69.77	**53.20**	52.90	**77.17**	**74.73**	**18.66**	**17.42**

Table 2: **Performance of proposed BalDistill framework and other baselines across five long-tailed datasets.** The best performance is marked in bold. The performance of fine-tuned student models with our framework can outperform other baselines in macro-averages on multiple long-tailed datasets.

Method	R52	Reuters	AbsQA	MCQA	Math
Budget Setting 1					
Random FT CoT	**59.70**	27.35	52.57	76.09	15.20
Balance FT CoT	51.47	27.12	52.22	75.98	16.29
Active FT CoT	59.49	29.75	53.14	76.64	15.61
BalDistill (N)	59.62	28.09	52.70	76.60	16.34
BalDistill (A)	58.93	**32.95**	**53.20**	**77.17**	**17.42**
Budget Setting 2					
Random FT CoT	64.88	**33.42**	53.71	72.92	15.19
Balance FT CoT	60.55	32.79	50.29	76.29	15.73
Active FT CoT	64.54	31.33	53.05	76.26	15.91
BalDistill (N)	59.35	32.76	**53.86**	76.17	17.59
BalDistill (A)	**65.84**	32.77	53.49	**77.11**	**17.59**

Table 3: **Effects of active learning and adaptive balancing in BalDistill framework.** Results of fine-tuned student models on five datasets outperform methods with only balancing (Balance FT CoT), with only active learning (Active FT CoT).



(a) AbstractiveQA (b) Reuters

(c) MultichoiceQA (d) Math

Figure 3: **Performance of proposed method BalDistill and ablated methods on head and tail domains.** BalDistill (A) can achieve better results on head domains and outperform the Active FT CoT method on tail domains, which demonstrates the effectiveness of each component in our BalDistill (A) framework.

in Table 3.

Both active selection and adaptive balancing bring salient performance boost From Table 3, we find that our BalDistill (A) method obtains the best performance in 7/10 comparison cases, which demonstrates the effectiveness of each framework component. We notice that by simply adding the active learning strategy (Random Finetune CoT vs. Active Finetune CoT), the fine-tuned student model can achieve a performance boost in most cases, with an average relative improvement of 1.43%. This observation is consistent with the findings in previous work for Bert models (Devlin et al., 2019) on the long-tailed data (Dor et al., 2020).

However, when we add data augmentation from the teacher with the naive balancing policy (Balance Finetune CoT vs. Random Finetune CoT, BalDistill (N) vs. Active Finetune CoT), this operation does not substantially improve performance. This finding suggests the superiority of our adaptive balancing policy.

To probe the detailed reasons for the result patterns above, we visualize the macro-average performance of these methods on inputs from head and tail domains in Figure 3. The splitting criteria for each dataset can be found in Appendix A. We find that for methods with naive balancing policy (Balance Finetune CoT and BalDistill (N)), there exists a significant performance drop on head domains due to filtering a large proportion of data, and our method with adaptive balancing can achieve comparable performance on head domains. The observation suggests the effectiveness of active selection for head domains and the importance of adaptive balancing for the fine-tuned student to select the uncertain ones precisely.

For performance in tail domains, our proposed method with adaptive balancing and teacher augmentation could achieve the best average results, even better than the naive balancing method. We conjecture that since we do not verify the correctness of teacher-generated samples and rationales in tail domains. While teacher-generated samples induce more knowledge, more synthetic data can lead to more inevitable noise. Adaptive balancing achieves the best trade-off between inducing more knowledge and less noise in the tail domains.

5.3 Generalization Analysis

The ablation study demonstrates the effectiveness of the active learning and adaptive balancing. Then, we ask whether our proposed method is robust enough to experiment with different hyperparameters, student models, or budget settings.

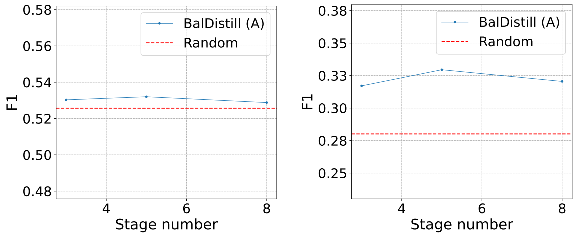
5.3.1 Generalizations on Student models

Method	R52	Reuters	AbsQA	MCQA	Math
		Llama3 Budget Setting 1			
Random FT CoT	**59.70**	27.35	52.57	76.09	15.20
Active FT CoT	59.49	29.75	53.14	76.64	15.61
BalDistill (A)	58.93	**32.95**	**53.20**	**77.17**	**17.42**
		Llama2 Budget Setting 1			
Random FT CoT	49.83	23.97	46.26	58.69	3.43
Active FT CoT	46.88	24.06	47.07	58.68	3.82
BalDistill (A)	**58.33**	**25.51**	**47.55**	**59.14**	**4.21**
		Llama3 Budget Setting 2			
Random FT CoT	64.88	**33.42**	53.71	72.92	15.19
Active FT CoT	64.54	31.33	53.05	76.26	15.91
BalDistill (A)	**65.84**	32.77	**53.49**	**77.11**	**17.59**
		Llama2 Budget Setting 2			
Random FT CoT	56.45	23.75	48.95	58.91	3.84
Active FT CoT	53.16	**27.12**	48.27	**59.20**	3.52
BalDistill (A)	**58.17**	27.07	**49.45**	58.64	**4.54**

Table 4: **Effects of student model scales and budget numbers.** Macro-averages the proposed and baseline method results when considering Llama2 and Llama3 as student models with varying two budget settings.

We first evaluate whether our method could be generalized to student models with different reasoning abilities or with different budget numbers. In this part, we additionally evaluate our BalDistill (A) on Llama2-7B models, which have a smaller model size and fewer tokens, in two budget settings (the details of each dataset are in Table 1). We present the fine-tuning results of our proposed framework and baseline methods on the Llama2 and Llama3 student models in Table 4.

BalDistill exhibits robust improvement with various budget settings or student models. From Table 4, we observe that fine-tuning with the Llama3-8B student model leads to much better performance than the Llama2-7B model, especially on tasks with complex reasoning (Math, Multi-choice QA), indicating that the student with a larger model size or a better reasoning ability will yield better fine-tuning results. This observation is consistent with previous findings in Ho et al. (2022); Hsieh et al. (2023). Our BalDistill (A) consistently outperforms other baseline methods on both Llama2-7B and Llama3-8B as student models in most cases under two budget settings, which also verifies the generalizability of our BalDistill (A) on different student models or different budget numbers.



(a) AbstractiveQA (b) Reuters

Figure 4: **Influence of stage number choices on BalDistill across two datasets.** Our proposed method consistently obtains better results than the random fine-tune baseline method with varying stage numbers.

5.3.2 Sensitivity Analysis

We next investigate how the choice of stage number: K will influence the performance of our framework. We experiment with the same setup as in Section 5.1 but with varying stage numbers among $\{3, 5, 8\}$. We visualize the results (macro-averages) of BalDistill (A) and the baseline method Random Finetune CoT in Figure 4 and 6 in Appendix.

Figure 4 shows that the fine-tuning results of BalDistill (A) could be affected by the stage number to some extent, but our proposed method can consistently outperform the baseline method with different stage numbers, demonstrating the effectiveness and robustness of BalDistill (A).

6 Conclusions

In this paper, we propose a novel framework BalDistill to enhance performance on long-tail datasets in the current teacher-student knowledge distillation process. Our framework is a multi-stage pipeline, and at each stage, we call the student models to actively select the representative examples from head domains while prompting the teacher to generate synthetic examples for tail domains. With a fixed budget restriction for calling the teacher, our extensive empirical evaluations show that our framework can significantly increase fine-tuning results across multiple datasets. Furthermore, we demonstrate the effectiveness of all framework components through ablation studies.

7 Limitations

In our work, we use the IFD score as the metric for active selection for the student model. In addition to IFD scores, we can try other metrics, such as maximum entropy (Settles, 2009) or K-means diversity (Yuan et al., 2020). However, previous work has shown that the IFD score is more effective in selecting data for sequence-level fine-tuning than other metrics (Li et al., 2024a,b).

We have verified the effectiveness of our framework on multiple student models and various long-tailed datasets. Other sequence-level KD methods still use more complex loss functions (Hsieh et al., 2023) or augment the generated rationales (Shridhar et al., 2023). Our data manipulation framework complements these KD methods, aiming to achieve more robust results on long-tailed datasets with a fixed budget. Moreover, our method focuses on sequence-level KD for black-box LLMs, so we do not incorporate the KD method for white-box LLMs as a baseline method (Gu et al., 2023; Dai et al., 2023). We will leave the exploration of combining our framework with more advanced KD methods for the future.

Furthermore, our experiments only focus on the decoder-only student models: Llama3 and Llama2. Incorporating more encoder-decoder models such as FLAN-T5 (Chung et al., 2022) would benefit future studies.

References

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. ICLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. arXiv preprint arXiv:2305.05176.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. 2021. Parametric contrastive learning. In Proceedings of the IEEE/CVF international conference on computer vision, pages 715–724.

Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, and Yongbin Li. 2023. Long-tailed question answering in an open world. arXiv preprint arXiv:2305.06557.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Liat Ein Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active learning for bert: an empirical study. In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pages 7949–7962.

Ameya Godbole and Robin Jia. 2022. Benchmarking long-tail generalization with likelihood splits. arXiv preprint arXiv:2210.06799.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. In The Twelfth International Conference on Learning Representations.

David Harbecke, Yuxuan Chen, Leonhard Hennig, and Christoph Alt. 2022. Why only micro-f1? class weighting of measures for relation classification. arXiv preprint arXiv:2205.09460.

Philip J Hayes and Steven P Weinstein. 1990. Construe/tis: A system for content-based indexing of a database of news stories. In IAAI, volume 90, pages 49–64.

Nan He, Hanyu Lai, Chenyang Zhao, Zirui Cheng, Junting Pan, Ruoyu Qin, Ruofan Lu, Rui Lu, Yunchen Zhang, Gangming Zhao, et al. 2023. Teacherlm: Teaching to fish rather than giving the fish, language modeling likewise. arXiv preprint arXiv:2310.19019.

Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. 2021. Distilling virtual examples for long-tailed recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 235–244.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874.

Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2022. A survey of methods for addressing class imbalance in deep-learning based natural language processing. arXiv preprint arXiv:2210.04675.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. arXiv preprint arXiv:2212.10071.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. arXiv preprint arXiv:2305.02301.

Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing methods for multi-label text classification with long-tailed class distribution. arXiv preprint arXiv:2109.04712.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. The Journal of the Acoustical Society of America, 62(S1):S63–S63.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. arXiv preprint arXiv:2310.06825.

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

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1317–1327.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199–22213.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024a. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. 2019. Large-scale long-tailed recognition in an open world. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2537–2546.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 5011–5034.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 650–663.

OpenAI. 2023. Gpt-4 technical report.

Ian Osband, Seyed Mohammad Asghari, Benjamin Van Roy, Nat McAleese, John Aslanides, and Geoffrey Irving. 2022. Fine-tuning language models via epistemic neural networks. arXiv preprint arXiv:2211.01568.

Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling bias in deep active classification: An empirical study. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4058–4068.

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

Hamidreza Rouzegar and Masoud Makrehchi. 2024. Enhancing text classification through llm-driven active learning and human annotation. In Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII), pages 98–111.

Dongyu Ru, Jiangtao Feng, Lin Qiu, Hao Zhou, Mingxuan Wang, Weinan Zhang, Yong Yu, and Lei Li. 2020. Active sentence learning by adversarial uncertainty sampling in discrete space. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4908–4917.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2339–2352.

Burr Settles. 2009. Active learning literature survey.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2022. Distilling reasoning capabilities into smaller language models. arXiv preprint arXiv:2212.00193.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In Findings of the Association for Computational Linguistics: ACL 2023, pages 7059–7073.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-tail: How knowledgeable are large language models (llm)? aka will llms replace knowledge graphs? arXiv preprint arXiv:2308.10168.

Naama Tepper, Esther Goldbraich, Naama Zwerdling, George Kour, Ateret Anaby Tavor, and Boaz Carmeli. 2020. Balancing via generation for multi-class text classification improvement. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1440–1452.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Dongdong Wang, Yandong Li, Liqiang Wang, and Boqing Gong. 2020. Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1498–1507.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. Advances in neural information processing systems, 30.

Ruixuan Xiao, Yiwen Dong, Junbo Zhao, Runze Wu, Minmin Lin, Gang Chen, and Haobo Wang. 2023. Freeal: Towards human-free active learning in the era of large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 14520–14535.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. arXiv preprint arXiv:2402.13116.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. arXiv preprint arXiv:2309.12284.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. arXiv preprint arXiv:2010.09535.

Chen Zhang, Lei Ren, Jingang Wang, Wei Wu, and Dawei Song. 2022. Making pretrained language models good long-tailed learners. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3298–3312.

Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. 2023. Llmaaa: Making large language models as active annotators. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 13088–13103.

Xing Zhang, Zuxuan Wu, Zejia Weng, Huazhu Fu, Jingjing Chen, Yu-Gang Jiang, and Larry S Davis. 2021. Videolt: Large-scale long-tailed video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7960–7969.

Yuhang Zhou and Wei Ai. 2024. Teaching-assistant-in-the-loop: Improving knowledge distillation from imperfect teacher models in low-budget scenarios.

Yuhang Zhou, Suraj Maharjan, and Beiye Liu. 2023. Scalable prompt generation for semi-supervised learning with language models. arXiv preprint arXiv:2302.09236.

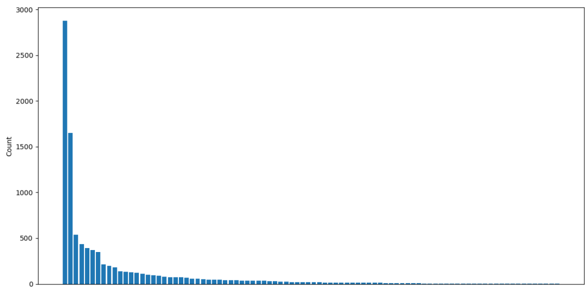
Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. 2024. Explore spurious correlations at the concept level in language models for text classification.

A Dataset Construction

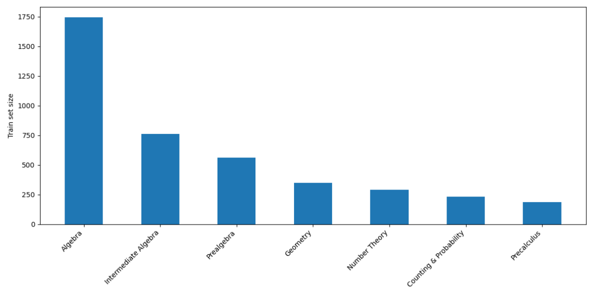
R52 & Reuters We use the original R52 and Reuters dataset. In Figure 3, we treat domains (labels) with more than 50 instances in the training dataset as the head domains and the others as tail domains.

Multi-choice QA For Multi-choice QA, we merge 10 multichoice QA datasets together, including Race, OBQA, MCTest, ARC-easy, ARC-hard, CQA, QASC, PIQA, SIQA, Winogrande (Dai et al., 2023). For training samples, we downsample the 10 datasets following a Zipf distribution with power value $\alpha = 2.0$ (Dai et al., 2023). Since Race has 5× more training samples than other datasets, we downsample its training and testing set to 1/3 of the samples using random sampling. The detailed statistics of each multichoice qa dataset is shown in Table 5.We select Race, Winogrande, SIQA and CQA as the head domains and others as tail domains for experiments in Figure 3.

Abstractive QA For Abstractive QA, we merge 5 abstractive QA datasets together, including NarQA, NQOpen, Drop, QAConv, TweetQA (Dai et al., 2023). Since the total train set and test set are very large, for efficiency concerns, we randomly sample 10000 samples from them for both train and test



(a) Reuters



(b) Math

Figure 5: Example Dataset Distribution: The datasets we use exhibit long-tail distributions.

sets. The detailed statistics of each multichoice qa dataset is shown in Table 5. We select NarQA, NQOpen, and Drop as the head domains and others as tail domains for experiments in Figure 3.

Math We use the Math dataset from (Hendrycks et al., 2021), which consists of 7 categories: Algebra, Intermediate Algebra, Prealgebra, Geometry, Number Theory, Counting & Probability and Precalculus. In order to investigate GPT4's reasoning ability on MATH problems and how much can its reasoning be taughts to the student model, we removed the reasoning procedures in Math dataset and only keep its final answer as the label. Since the original dataset distribution is as follows does not follow long tail distribution, we down-sample the training sets of all categories following a Zipf distribution with power value $\alpha = 1.1$, similar to (Dai et al., 2023). The final distribution of the datsaet is shown in Table 5. We select Algebra, Intermediate Algebra, and Prealgebra as the head domains and others as the tail domains for experiments in Figure 3.

B Implementation Details

We use greedy search in decoding for all teacher annotations, as in the previous work (Ho et al., 2022) and use stochastic temperature sampling with the same temperature value of 0.9 in synthetic data generation in Section 3.4.

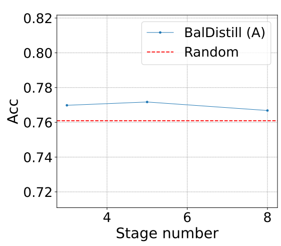
We use the zero-shot prompts for the teacher to

Table 5: Detailed statistics of each dataset per category.

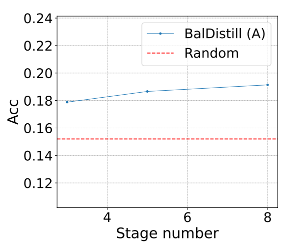
Dataset	Category	Train set size	Test set size
Multi-choice QA	Race	4735	1629
	OBQA	580	500
	MCTest	342	320
	ARC-easy	395	570
	ARC-hard	317	299
	CQA	1034	1221
	QASC	653	926
	SIQA	2077	1954
	PIQA	494	1838
	Winogrande	2634	1267
Abstractive QA	NarQA	1999	2244
	NQOpen	4441	3434
	Drop	2525	2891
	QAConv	751	1079
	TweetQA	284	352
Math	Algebra	1744	1187
	Intermediate Algebra	763	903
	Prealgebra	561	871
	Geometry	349	479
	Number Theory	290	540
	Counting & Probability	231	474
	Precalculus	187	546

Algorithm 1 Multi Stage Balanced Distillations

Input: Long tailed dataset D, Student model M_s, Teacher model M_t, prompt for generating data P_c, Stage number K, Balancing policy P, Training bucket T, Budget number B
Output: The fine-tuned student model M_s^K
for each stage $k = 0, \ldots, k-1$ **do**
    head, tail domains $= P(D, k, B)$
    for each domain tail domain j **do**
        Add remaining x_j from D to T
        $\hat{x_j} = M_t(P_c, j)$
        Add synthetic $\hat{x_j}$ to T
    for each domain head domain h **do**
        Collect all x_h from D
        $x_h = M_s^{k-1}(x_h, h)$
        Add selected x_h to T
    Use M_t to annotate x in T w/o rationales
    $M_s^k = \text{Fine-tune}(M_s, T)$



(a) MultiChoiceQA



(b) Math

Figure 6: **Influence of stage number choices on BalDistill across other two datasets.**

give the rationales and the few-shot ICL to generate the synthetic tail samples. The prompts are shown in Tables 6, 7 and 8. We call the gpt-4 function from OpenAI to obtain teacher responses.

For the fine-tuning of the student model, we base our implementation on the Pytorch[1], Huggingface transformer[2], and the Lora fine-tuning codebase [3]. We use AdamW as our optimizer with a learning rate of $2\mathrm{e}{-4}$ and a weight decay of 0.03 with linear scheduler, batch size of 16, and trained for 8 epochs. For other hyper-parameters, we set rank and dropout in Lora fine-tuning to 8 and 0.1, respectively.

[1] https://pytorch.org/

[2] https://huggingface.co/

[3] https://github.com/georgian-io/LLM-Finetuning-Toolkit/tree/main

You are provided with a dataset named R52, which is specifically designed for text classification tasks. The objective is to accurately predict the topic of news stories from a predefined list of topics. The topic of this dataset includes: copper, livestock, gold, money-fx, tea, ipi, trade, cocoa, iron-steel, reserves, zinc, nickel, ship, cotton, platinum, alum, strategic-metal, instal-debt, lead, housing, gnp, sugar, rubber, dlr, tin, interest, income, crude, coffee, jobs, meal-feed, lei, lumber, gas, nat-gas, veg-oil, orange, heat, wpi, cpi, earn, jet, potato, bop, money-supply, carcass, acq, pet-chem, grain, fuel, retail, cpu. Please write a short news story with the topic {domain} and give the step-by-step rationale. This should be a self-contained story, mirroring the style and content of real-world news articles. Here are some examples with the topic {domain}:
{demonstrations}
Please compose a news story with the topic {domain} with a similar format as the example. Paraphrase your title before outputting it. Your news story should be brief and contained within one paragraph:

(a) R52

You are provided with a dataset named reuters, which is specifically designed for text classification tasks. The objective is to accurately predict the topic of news stories from a predefined list of topics. The topic of this dataset includes: acq, rubber, lead, money-supply, income, l-cattle, crude, cpu, palmkernel, jobs, money-fx, instal-debt, rand, castor-oil, coffee, strategic-metal, nat-gas, oat, tea, corn, yen, soy-oil, grain, groundnut-oil, gas, cpi, cocoa, nzdlr, soybean, rapeseed, retail, sun-meal, coconut, jet, copper, sorghum, carcass, heat, hog, ipi, potato, lin-oil, oilseed, alum, gnp, meal-feed, fuel, barley, ship, rape-oil, cotton-oil, sunseed, palm-oil, soy-meal, naphtha, nkr, trade, palladium, lei, wheat, bop, interest, earn, reserves, housing, veg-oil, groundnut, tin, dlr, gold, copra-cake, wpi, livestock, zinc, sugar, rye, pet-chem, dmk, dfl, orange, iron-steel, nickel, sun-oil, lumber, rice, propane, platinum, silver, cotton, coconut-oil. Please write a short news story with the topic {domain} and give the step-by-step rationale. This should be a self-contained story, mirroring the style and content of real-world news articles. Here are some examples with the topic {domain}:
{demonstrations}
Please compose a news story with the topic {domain} with a similar format as the example and your news story should be brief and contained within one paragraph:

(b) Reuters

Table 6: Prompts of generating synthetic data for tail domains from the teacher for R52 and reuters datasets.

You are provided with a multiple-choice question and answering dataset composed by various QA datasets. The objective is to accurately select one from the given choices according to the question content. Please compose a question as well as the corresponding choices and answers as the examples from a QA dataset: {domain}. This should be a question, mirroring the style and content of examples with the true real-world knowledge. Here are some examples from the QA dataset: {domain}: {demonstrations} Please compose a question for the dataset: {domain} with a similar format as the example. It means if the example contains the in-context "passage", you should also write an in-context "passage" with the question information. Your question and choices should be brief and contained within one paragraph:

(a) Multi-choice QA

You are provided with an abstractive question answering dataset composed by various QA datasets. The objective is to accurately generate an answer according to the question content. Please compose a question and the corresponding answer as the examples from a QA dataset: {domain}. This should be a question and answer, mirroring the style and content of examples with the true real-world knowledge. Here are some examples from the QA dataset: {domain}: {demonstrations} Please compose a question and the corresponding answer for the dataset: {domain} with a similar format as the example. It means if the example contains the in-context "passage", you should also write an in-context "passage" with the question information. Please note that the answer should only contain a few words. Your question and answer should be brief and contained within one paragraph:

(b) Abstractive QA

You are provided with a math problem dataset with questions from various math domains. The objective is to accurately generate an answer according to the question content. Please compose a question and the corresponding answer as the examples from a math domain: {domain}. This should be a math question and answer, mirroring the style and content of examples with the true real-world knowledge. Here are some examples from the math domain: {domain}: {demonstrations} Please compose a math question and the corresponding answer for the domain: {domain}, with a similar format as the example. Please output your final digital answer (no unit) for the question with the format: "the answer is: <answer>". Your question and answer should be brief and contained within one paragraph:

(c) Math

Table 7: Prompts of generating synthetic data for tail domains from the teacher for Multi-choice QA, Abstractive QA and Math datasets.

Below is a news story from the R52 dataset. Please assign a topic to this news story. You must select the topic from this set: copper, livestock, gold, money-fx, tea, ipi, trade, cocoa, iron-steel, reserves, zinc, nickel, ship, cotton, platinum, alum, strategic-metal, instal-debt, lead, housing, gnp, sugar, rubber, dlr, tin, interest, income, crude, coffee, jobs, meal-feed, lei, lumber, gas, nat-gas, veg-oil, orange, heat, wpi, cpi, earn, jet, potato, bop, money-supply, carcass, acq, pet-chem, grain, fuel, retail, cpu. News story: {input}.
Take a step-by-step approach in your response, cite sources and give reasoning. Your answer should be brief and contained within one paragraph.

(a) R52

Below is a news story from the reuters dataset. Please assign a topic to this news story. You must select the topic from this set: acq, rubber, lead, money-supply, income, l-cattle, crude, cpu, palmkernel, jobs, money-fx, instal-debt, rand, castor-oil, coffee, strategic-metal, nat-gas, oat, tea, corn, yen, soy-oil, grain, groundnut-oil, gas, cpi, cocoa, nzdlr, soybean, rapeseed, retail, sun-meal, coconut, jet, copper, sorghum, carcass, heat, hog, ipi, potato, lin-oil, oilseed, alum, gnp, meal-feed, fuel, barley, ship, rape-oil, cotton-oil, sunseed, palm-oil, soy-meal, naphtha, nkr, trade, palladium, lei, wheat, bop, interest, earn, reserves, housing, veg-oil, groundnut, tin, dlr, gold, copra-cake, wpi, livestock, zinc, sugar, rye, pet-chem, dmk, dfl, orange, iron-steel, nickel, sun-oil, lumber, rice, propane, platinum, silver, cotton, coconut-oil. News story: {input}.
Take a step-by-step approach in your response, cite sources and give reasoning. Your answer should be brief and contained within one paragraph.

(b) Reuters

Please answer this multiple-choice question by choosing one of the given choices. If you are given a passage, please answer the question according to the passage content. If the passage is not given, please answer the question directly from your knowledge. Question: {input}
If there is no enough information, you should choose a most possible choice. Take a step-by-step approach in your response, cite sources and give reasoning before sharing final answer in the format: The answer is <selected choice>.

(c) Multi-choice QA

Here are a question and the corresponding answer for an abstractive question answering task. Please concisely clarify the rationale behind the answer for this question. If you are given a passage, please use the passage content to clarify the answer. If the passage is not given, please use your knowledge to tell why the answer is reasonable. Question: {input}. Answer {label}.
Take a step-by-step approach in your response and give reasoning. You output should be concise and in one paragraph.

(d) Abstractive QA

Here are a math question and the ground truth answer for a math problem task. Please concisely clarify the rationale behind the answer for this math question. Question: {input}. Answer: {label}
Take a step-by-step approach in your response and give reasoning. You output should be concise and in one paragraph.

(e) Math

Table 8: Teacher annotation prompts for all datasets. Note that for Math, Abstractive QA which has complex reasoning trajectories, we directly provide the ground-truth answer and ask the teacher model to explain the rationale behind the answer and for other datasets, we ask the teacher model to annotate the label and provide the reasoning steps.

Input: ENERGY GIANT <EGC> EXPANDS NATURAL GAS OPERATIONS. Energy Giant Corp (EGC) announced plans to expand its natural gas operations in the Midwest, following approval from the Federal Energy Regulatory Commission. The expansion will see EGC increase its natural gas production by 15%, aiming to meet the growing demand in the region. The company stated that the expansion will be funded by a mix of internal accruals and debt, with the project expected to be operational by the end of the next fiscal year. EGC also highlighted that the expansion aligns with its commitment to sustainable energy solutions, as natural gas is a cleaner-burning fuel compared to coal and oil. The company's shares rose 2% in early trading following the announcement.

Rationale: The news story is about Energy Giant Corp (EGC) announcing plans to expand its natural gas operations in the Midwest. The company plans to increase its natural gas production by 15% to meet the growing demand in the region. The expansion will be funded by a mix of internal accrivals and debt. The company's shares rose 2% in early trading following the announcement.The story does not mention anything about copper, livestock, gold, money-fx, tea, ipi, trade, cocoa, iron-steel, reserves, zinc, nickel, ship, cotton, platinum, alum, strategic-metal, instal-debt, lead, housing, gnp, sugar, rubber, dlr, tin, interest, income, crude, coffee, jobs, meal-feed, lei, lumber, gas, veg-oil, orange, heat, wpi, cpi, earn, jet, potato, bop, money-supply, carcass, acq, pet-chem, grain, fuel, retail, cpu. However, the story is about the expansion of natural gas operations, which falls under the topic of 'nat-gas'. The answer is nat-gas.

(a) The example of teacher synthetic data in R52 dataset for domain: nat-gas.

Input: A rectangle has a length of 14 units and a width of 8 units. A diagonal is drawn from one corner of the rectangle to the opposite corner. What is the length of the diagonal in units?

Rationale: The length of the diagonal of a rectangle can be found using the Pythagorean theorem, which relates the lengths of the sides of a right triangle. The theorem states that the square of the length of the hypotenuse (the side opposite the right angle, which is the diagonal in this case) is equal to the sum of the squares of the lengths of the other two sides (the length and width of the rectangle). For this rectangle, the length (l) is 14 units and the width (w) is 8 units. According to the Pythagorean theorem, the length of the diagonal (d) can be calculated as follows: $d^2 = l^2 + w^2$. Plugging in the values gives $d^2 = 14^2 + 8^2 = 196 + 64 = 260$. Taking the square root of both sides to solve for d gives d = $\sqrt{260}$. Factoring out the square root of 4 (which is a perfect square) from under the radical to simplify, we get $d = 2\sqrt{65}$. Thus, the length of the diagonal is $2\sqrt{65}$ units. The answer is: $2\sqrt{65}$.

(b) The example of teacher synthetic data in math dataset for domain: geometry.

Table 9: Examples of the synthetic input and rationales from the teacher model for the tail domains.