Learn More, Forget Less: A Gradient-Aware Data Selection Approach for LLM

Anonymous ACL submission

Abstract

Despite large language models (LLMs) have achieved impressive achievements across numerous tasks, supervised fine-tuning (SFT) remains essential for adapting these models to specialized domains. However, SFT for domain specialization can be resource-intensive and sometimes leads to a deterioration in performance over general capabilities due to catastrophic forgetting (CF). To address these issues, we propose a self-adaptive gradient-aware data selection approach (GrADS) for supervised fine-tuning of LLMs, which identifies effective subsets of training data by analyzing gradients obtained from a preliminary training phase. Specifically, we design self-guided criteria that leverage the magnitude and statistical distribution of gradients to prioritize examples that contribute the most to the model's learning process. This approach enables the acquisition of representative samples that enhance LLMs understanding of domain-specific tasks. Through extensive experimentation with various LLMs across diverse domains such as medicine, law, and finance, GrADS has demonstrated significant efficiency and cost-effectiveness. Remarkably, utilizing merely 5% of the selected GrADS data, LLMs already surpass the performance of those fine-tuned on the entire dataset, and increasing to 50% of the data results in significant improvements! With catastrophic forgetting substantially mitigated simultaneously. We will release our code for GrADS later.

1 Introduction

Although LLMs have achieved remarkable performances in multiple tasks such as open-domain question-answering (Achiam et al., 2023; Yang et al., 2024a), logical inference (Nam et al., 2024), and long-context understanding (Chen et al., 2023c), supervised fine-tuning remains indispensable for domain-specific scenarios (Chen et al., 2023b; Yue et al., 2023; Xiong et al., 2023; Yang et al., 2023). However, incorporating domain-specific knowledge and concepts into the LLM parameters could be rather costly. For the sake of efficiency promotion, some studies have shown that not all fine-tuning data are useful (Zhou et al., 2024), and removing some of the low-quality data instead can enhance model performance (Chen et al., 2023a; Li et al., 2023; Cao et al., 2023).

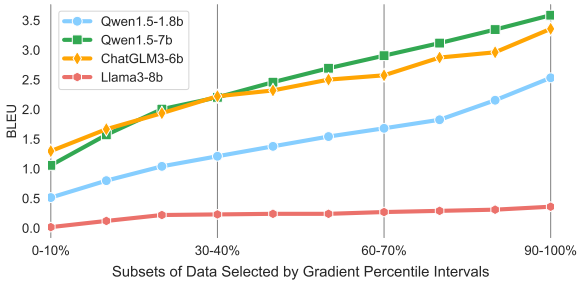

Figure 1: Pilot study: From left to right on the x-axis, we sort the CMedQA training data by gradients from largest to smallest, and select 10% by rank at each time, conducting 10 subsets, and predict their responses with untuned LLMs.

Besides, after domain-oriented fine-tuning, LLMs typically experience a decline in general capabilities, a phenomenon named Catastrophic Forgetting (CF) (Kaushik et al., 2021; Cossu et al., 2022). To address this issue, some practices use a mixture of domain-specific and general data (Luo et al., 2024), and others propose additional regularization or adaptation techniques (Ke, 2024; Diao et al., 2023). However, these approaches either increase the computational cost or compromise domain expertise to preserve more general capabilities (Lin et al., 2023).

To promote training efficiency and mitigate CF, we focus on leveraging LLMs to select high-quality subsets of data for training. Analogous to a well-educated student who can discern the most suitable college courses through trial classes, we posit that a sufficiently pre-trained LLM is capable of identify-

ing data that is more beneficial for its learning during the fine-tuning phase. Inspired by past works estimating the influence of training instances with gradient information (Pruthi et al., 2020; Han et al., 2023; Xia et al., 2024), we design a gradient-aware approach to select such data.

Therefore, we conduct a pilot study that illustrates the performance of vanilla LLMs in predicting outputs for slices of training data, each selected from different gradient intervals (Figure 1). The results show that the LLMs have higher accuracy when predicting data characterized by smaller gradients (right side) as opposed to larger gradients (left side), which confirms the potential of gradients in training data selection.

To effectively identify crucial data from candidate training sets, we propose an adaptive **Gradient-Aware Data Selection** method, namely **GrADS**. First, the entire candidate data would be trained for one epoch with the LLMs to extract gradients for each training instance. Then, a self-adaptive criterion based on the gradient distribution is used to select a subset from the full data of expected volume. This method eschews reliance on expensive, more advanced LLMs like GPT-4 for inference (Chen et al., 2023a; Du et al., 2023; Liu et al., 2023) and the need for manual intervention in creating high-quality seed data (Pan et al., 2024; Ge et al., 2024), thereby offering a cost-effective and pragmatic solution.

To carefully examine the effectiveness of GrADS, we conducted comprehensive experiments on various LLMs including Qwen (Bai et al., 2023), ChatGLM (Zeng et al., 2023), and Llama (AI@Meta, 2024) scaled from 1.8B, around 7B, to 14B, within typical knowledge-intensive and high-demand application domains like medicine (Zhang et al., 2023), law (Cui et al., 2024), and finance (Zhang and Yang, 2023). GrADS exhibits superior advantages in terms of efficiency, cost-effectiveness, and performance. In summary, our contributions are three folds:

- We introduce a novel self-adaptive Gradient-Aware Data Selection method (GrADS), which operates independently of manual intervention.
- Extensive experiments across different LLMs, model scales, and domains validate the efficacy of GrADS in facilitating target task performance.
- GrADS substantially mitigates the catastrophic forgetting problem, achieving an outstanding balance between domain specialization and general capabilities.

2 Related Work

2.1 Data Selection

The recent research by Zhou et al.(Zhou et al., 2024) indicates that most of the knowledge in LLMs is acquired during the pre-training phase, and a limited amount of instruction data is often sufficient to activate the models' capacity to follow instructions. Similarly, through interactions with SoTA LLMs such as GPT-4, Chen et al. (Chen et al., 2023a) introduced ALPAGASUS, while Li et al. (Li et al., 2023) proposed the Instruction-Following Difficulty (IFD) metric to select samples with desired characteristics to enhance LLM instruction tuning. Liu et al. (Liu et al., 2023) and Du et al. (Du et al., 2023) further delineated a series of criteria including quality, complexity, diversity, coverage, and necessity to select data. Additionally, some researchers constructed expert-aligned datasets (Ge et al., 2024) or curated high-quality seed data (Pan et al., 2024). By facilitating interaction between the LLM and these datasets alongside the original data, they aim to obtain feedback on the quality of the data and improve the models' performance.

However, the majority of the previous works necessitate human intervention or the involvement of SoTA LLMs such as GPT-4 for data filtering, which require substantial API quota budgets or significant human labor investments. In contrast, our data selection method addresses efficiency and cost-effectiveness, which is easy to implement and substantially reduces labor and API expenditures.

2.2 Catastrophic Forgetting

Domain-specific fine-tuned LLMs (Ouyang et al., 2022; Chung et al., 2024) have demonstrated substantial potential for knowledge-based question-answering (QA), auxiliary consulting, and personalized solution recommendation in various fields, such as medicine(Zhang et al., 2023), finance(Zhang and Yang, 2023) and law (Cui et al., 2024). However, as expertise within the domain advances, CF emerges along with a sharp decline in the general capabilities that the LLM had previously mastered during pre-training (Kaushik et al., 2021; Cossu et al., 2022; Luo et al., 2024).

To address this issue, from a data-driven perspective, some researchers trained both domain data and general data to reduce the forgetting of general knowledge (Chen et al., 2020), while others proposed self-distillation (Yang et al., 2024b), which guides the generation of task data through the model itself to minimize the disparity between the information distribution of the generated data and that of the initial model. On the other hand, from the model's perspective, some established end-to-end alignment of modules through shared attention mechanisms (Zhao et al., 2024), while others modified the adapter architecture by reducing the interference caused by fine-tuning tasks in different orthogonal low-rank subspaces (Wang et al., 2023) or by self-regulating the adapter's attention to different parts of the context (Liu et al., 2024).

3 Backgrounds

The **Embedding** layer and the **language model head (LM Head)** layer of LLMs play critical roles in capturing the semantics of input tokens and generating meaningful predictions, respectively. The Embedding layer maps each discrete token into a high-dimensional vector space, where the vectors capture the semantic and syntactic properties of the words they represent. On the other hand, the LM Head layer converts the final hidden states produced by the model into a probability distribution over the vocabulary and directly influences the model's accuracy in predicting the next token.

During back-propagation, the gradients computed for the Embedding layer indicate how the word vectors need to be updated to optimize the discriminative and context-aware token representations, which enhance the model's overall performance. Therefore, instances with larger gradients for the Embedding layer could imply the existence of unfamiliar information the model attempts to learn, while those with smaller gradients are rather stable and already well-presented.

In contrast, the gradients computed for the LM Head provide insights into how the model should adjust its parameters to minimize the prediction error in the decoding process, thereby improving its predictive capabilities. High-magnitude gradients show uncertainty and lack of confidence in the model's prediction, which reveals potential high complexity and perplexity of the data, whereas low-gradient tokens are well-understood and straight forward to the model.

With the insight that gradients help discover characteristics of each training instance, we raise a deduction that in a given training dataset D, the actual "effective" data points D' should depend on $feature\ importance$ (F), $information\ values$ (I), and $complexity$ (C):

$$D' \propto f(F, I, C) \tag{1}$$

4 GrADS: Gradient-Aware Data Selection

In this section, we introduce GrADS, which can adaptively select beneficial subsets of the data through gradient distribution, integrating both the Embedding layer gradients and LM Head gradients. The method consists of two major steps: gradient extraction from LLM learning, and data selection with a self-adaptive criterion. Specifically, in the first step, we obtain the integrated gradients of each training instance by conducting a single-epoch SFT. Subsequently, we select desired subsets of data based on an adaptive criterion derived from the gradient distribution. Our model architecture is illustrated in Figure 2.

4.1 Gradient Extraction from LLM Learning

Given the entire training data $\mathbf{D}$, we denote the input tokens of each data point $\mathbf{x} = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^T$, where T is the length of the input sequence. In the Transformer embedding layer, tokens are mapped to the corresponding embedding vectors $\mathbf{e}$:

$$\mathbf{e} = Embed(\mathbf{x}) \tag{2}$$

where $\mathbf{e} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T\} \in \mathbb{R}^{T*d}$ is the combined vector for input tokens, d denoting the dimension.

Then the embedded vectors $\mathbf{e}$ are passed through multiple Transformer layers, and produce the final hidden states $\mathbf{h} \in \mathbb{R}^{T*d}$ for all training instances.

$$\mathbf{h} = Transformer(\mathbf{e}) \tag{3}$$

The LM Head layer takes the final hidden states $\mathbf{h}$ and generates the probability distribution over the vocabulary for predicting the next token:

$$\mathbf{o} = softmax(LMHead(\mathbf{h})) \tag{4}$$

where $\mathbf{o} \in \mathbb{R}^{T*V}$ is the probability distribution over the vocabulary for the next token, and V is the size of the vocabulary.

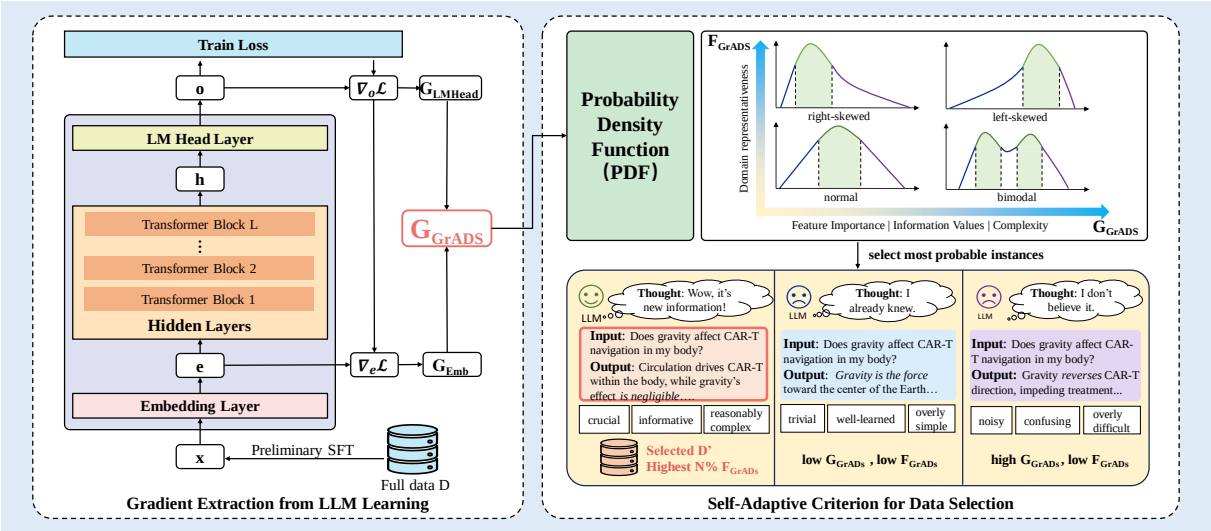

Figure 2: The illustration of the GrADS method.

Given a standard cross entropy loss function $\mathcal{L}(\cdot)$ that measures the difference between the model's predictions and the ground truth, the gradients of the loss for the embeddings can be obtained in the forward pass by:

$$\mathbf{g}_{\text{Emb}} = \nabla_{\mathbf{e}}\mathcal{L} = \left(\frac{\partial \mathbf{h}}{\partial \mathbf{e}}\right)^{\top} \nabla_{\mathbf{h}}\mathcal{L} \tag{5}$$

where $\nabla_{\mathbf{h}}\mathcal{L}$ is the gradient of the loss for the hidden state $\mathbf{h}$ and $\frac{\partial \mathbf{h}}{\partial \mathbf{e}}$ represents the Jacobian matrix (Wilamowski et al., 2008) of the hidden state for the embedding. The gradients $\nabla_{\mathbf{h}}\mathcal{L}$ can be obtained through backpropagation through the model.

Similarly, we can compute the gradients for the LM Head layer during the back-propagation step:

$$\mathbf{g}_{\text{LM}} = \nabla_{\mathbf{o}}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial \mathbf{o}} \tag{6}$$

After extracting the gradients for the input tokens in the Embedding and LM Head layers, we exclude special tokens like $[CLS]$, $[SEP]$, $[PAD]$, $[UNK]$, etc. Since the gradients of the Embedding layer reflect the LLMs' understanding of the input sequence whereas the gradients of the LM Head layer reflect the LLMs' certainty of the output tokens, we take all input tokens for the Embedding layer and only the output token for the LM Head layer into account. Meanwhile, to rule out the impact of input sequence length, we average the token-wise gradients for each training instance. Thus, the combined instance-level gradients for two layers are denoted as $\mathbf{G}^{i}_{\text{Emb}}$ and $\mathbf{G}^{i}_{\text{LM}}$, where $\mathbf{i} = \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{K}\}$ and $\mathbf{K}$ being the size of the data $\mathbf{D}$.

Thereafter, we integrate these gradients by adding $\mathbf{G}_{\text{Emb}}$ and $\mathbf{G}_{\text{LM}}$ linearly to derive a GrADS gradient vector $\mathbf{G}_{\text{GrADS}}$, which ultimately serves as the metric for selecting training instances

$$G_{\text{GrADS}} = G_{\text{Emb}} + G_{\text{LM}} \tag{7}$$

4.2 Self-Adaptive Criterion for Data Selection

To select the subset of training instances that best represents domain knowledge, we introduce the probability density function (PDF) to depict the distribution feature of $\mathbf{G}_{\text{GrADS}}$. The PDF uses a non-parametric method, such as kernel density estimation (KDE) to represent the density of $\mathbf{G}_{\text{GrADS}}$ at different values. A higher density signifies that there are more instances whose $\mathbf{G}_{\text{GrADS}}$ fall close, indicating instances more likely to share common domain characteristics. Just as one can quickly gain an understanding of a domain by reading its classical papers, prioritizing the fine-tuning process on these typical training instances can also enable LLMs to learn domain knowledge more efficiently and effectively.

Therefore, we compute the PDF function of $\mathbf{G}_{\text{GrADS}}$ gradients:

$$F_{\text{GrADS}} = PDF(G_{\text{GrADS}}) \tag{8}$$

where $\mathbf{F}_{\text{GrADS}} \in \mathbb{R}^{\mathbf{K}}$ implies the domain representativeness of the instance. Finally, an effectively refined subset of the full data $\mathbf{D}$ can be obtained by calculating the Top N% of instances with the highest PDF values:

$$D^{'} = \text{quantile}\left(F_{\text{GrADS}}, N/100\right) \tag{9}$$

GrADS operates in a self-adaptive manner, as it selects the most probable training instances regardless of the gradient distribution, whether it is left-skewed, right-skewed, normal, bimodal, or otherwise. The selected subset $\mathbf{D}'$ always has the highest $\mathbf{F}_{\text{GrADS}}$, thereby best capturing the critical characteristics of the domain. Also, $\mathbf{D}'$ carries crucial, informative, and reasonably challenging instances that guide the model to learn and acquire domain expertise. Nonetheless, as Figure 2 illustrates, training instances with low $\mathbf{G}_{\text{GrADS}}$ and low $\mathbf{F}_{\text{GrADS}}$ are typically less representative, often characterized as trivial, well-learned, or simple, and thus fail to "surprise" the model with already-known knowledge. In contrast, instances with high $\mathbf{G}_{\text{GrADS}}$ and low $\mathbf{F}_{\text{GrADS}}$ are often noisy, confusing, or overly difficult, and they might introduce misleading information that contradicts the model's established common sense. GrADS excludes these suboptimal instances by automatically adapting to the distribution of domains.

5 Experiments

In this section, we present the experiment results to verify the effectiveness of GrADS. Apart from the main results, we also try to validate the generalizability of GrADS by addressing the following research questions (RQs):(1) Generalizability: Can the GrADS approach be scaled up to larger LLMs and applied across different models? (2) Robustness: Do GrADS consistently perform well with smaller subsets selected?

5.1 Datasets

Our study incorporates three domains-specific datasets from three typical domains: CMedQA (Zhang et al., 2018b) for medicine, LawQA (Huang et al., 2023b) for law, and FinQA [1] for finance. The CMedQA dataset is provided by qualified experts, the LawQA dataset is generated by advanced LLMs, and the FinQA dataset is sourced from the open web and undergone post-cleaning. These datasets encompass the primary methodologies for fine-tuning data collection currently used, making experimental conclusions derived from those datasets representative, and can be reasonably expected to generalize to a wider range of data.

Specifically, CMedQA includes 20k instances for training and 0.5k instances for testing. For LawQA, we use the law article-based QA pairs from the Lawyer-LLama project (Huang et al., 2023a) and split 1.6k and 0.4k data for training and testing, respectively. Since FinQA's QA pairs are sourced from webpages, we retained only those designated as "best answers" in the original dataset. Additionally, we removed all duplicate questions and answers, resulting in a training set of 40k and a testing set of 2k.

5.2 Evaluation Metrics

We follow Pan et al. (Pan et al., 2024) to include BLEU (Papineni et al., 2002), along with ROUGE-L (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) to evaluate the response quality. In addition, we employed GPT-4o to score on a 1-5 scale of the response quality. We also provide the consistency test between GPT-4o evaluation and human judgement in Appendix A.

Furthermore, we delve into the CF problem in general capabilities following supervised fine-tuning on domain-specific instances. To this end, we follow the work of Liu et al. (Liu et al., 2024) and collect C-Eval (Huang et al., 2023c) for common sense understanding, GSM8K (Yu et al., 2023) for mathematics, ALPACA (Peng et al., 2023) for instruction following and SafetyPrompts (Sun et al., 2023) for instruction attack and typical safety scenarios awareness.

For C-Eval, we write a rule-based method to extract the options predicted by LLMs, and report the accuracy and whether the LLMs follow the instruction of "Single-choice questions". For GSK8k, we apply its publicly released Chinese version which is translated by GPT3.5-Turbo [2]. We follow the previous work [3] to extract the numerical results predicted by LLMs and report the accuracy, BLEU, and ROUGE-L. For ALPACA, we report BLEU and ROUGE-L. For SafetyPrompts (Typical Safety and Instruction Attack subdata), we write a few-shot prompt to instruct GPT-4o to conduct a 2 choice task on whether the LLMs' responses are safe or not. The responses are considered as correct if GPT-4o labels them as "safe".

5.3 Foundation Models

To validate GrADS' efficiency across different model scales and model architectures, we selected Qwen1.5-7B-Chat (Bai et al., 2023), ChatGLM3-6B-Chat (Zeng et al., 2023) and Llama3-8B-Instruct (AI@Meta, 2024) as our base LLMs. We

[1] https://aistudio.baidu.com/datasetdetail/34744

[2] https://huggingface.co/datasets/meta-math/GSM8K_zh

[3] https://github.com/QwenLM/Qwen

Base Model	Method	CMedQA			LawQA			FinQA		
		BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR
Qwen1.5-7B	base	2.627	12.180	10.860	9.066	20.050	21.392	3.188	11.194	14.669
	all	3.813	17.327	12.276	16.090	27.603	27.472	10.120	24.067	18.757
	rdn	3.548	16.776	11.954	15.856	27.288	26.810	9.686	22.621	17.276
	bm25	4.133	18.152	13.260	16.667	27.538	28.264	10.419	23.837	20.645
	dsir	3.650	17.636	12.314	15.987	27.362	27.644	9.876	23.463	19.142
	rds	3.826	17.980	12.744	16.203	27.862	28.017	10.133	24.135	20.075
	ppl	4.871	18.285	14.689	18.013	27.776	30.660	11.419	23.304	23.325
	less	5.126	18.214	14.896	19.473	**29.314**	33.727	12.884	**25.135**	23.976
	grads	**5.372**	**18.496**	**15.396**	**20.270**	28.026	**35.985**	**13.364**	24.822	**24.872**
ChatGLM3-6B	base	2.568	11.274	15.634	7.966	19.733	19.011	3.174	11.437	14.926
	all	4.297	17.432	16.722	16.673	28.016	28.519	11.454	22.918	24.898
	rdn	4.512	16.674	16.482	16.453	27.576	27.864	11.216	22.450	24.233
	bm25	4.824	17.015	17.163	16.929	27.798	28.316	11.636	23.412	25.170
	dsir	4.330	16.488	15.856	16.215	26.943	27.534	11.328	22.390	24.421
	rds	4.607	17.216	16.754	17.036	27.689	28.525	11.596	23.538	24.427
	ppl	5.031	17.503	17.637	18.865	**28.411**	33.068	11.957	24.214	26.682
	less	5.283	**18.425**	**18.529**	19.002	28.214	33.337	12.216	23.790	26.394
	grads	**5.488**	17.813	18.375	**20.288**	28.067	**34.932**	**13.165**	**24.281**	**28.567**
Llama3-8B	base	0.026	0.249	0.291	0.259	1.905	2.164	0.178	1.293	1.225
	all	3.332	16.415	11.061	15.272	24.301	27.033	9.116	21.190	16.913
	rdn	3.265	15.884	10.798	15.552	24.688	26.476	9.337	22.654	16.870
	bm25	3.474	16.763	12.018	15.859	24.803	28.165	10.225	22.387	18.244
	dsir	3.206	15.817	11.001	14.643	24.112	25.386	9.640	22.818	17.266
	rds	3.399	16.352	12.679	15.704	24.638	27.766	10.413	22.694	18.375
	ppl	4.183	**17.809**	13.632	16.390	25.122	30.378	11.863	22.817	22.469
	less	4.213	17.130	13.845	16.737	25.015	31.408	11.480	22.526	23.425
	grads	**4.472**	17.365	**14.089**	**18.751**	**26.613**	**34.620**	**12.288**	**23.678**	**23.437**

Table 1: Main Results. *base* denotes no further training implemented, *all* denotes full dataset, and otherwise we select 50% of the data for training.

Base Model	Method	CMedQA	LawQA	FinQA
Qwen1.5-7B	all	2.712	3.318	2.679
	grads	**3.159**	**4.202**	**3.295**
ChatGLM3-6B	all	2.587	3.254	2.826
	grads	**3.215**	**4.034**	**3.336**
Llama3-8B	all	2.553	3.110	2.547
	grads	**2.887**	**3.823**	**2.914**

Table 2: Results by GPT-4o's evaluation, scores range from 1-5.

also selected Qwen1.5-1.8B-Chat and Qwen1.5-14B-Chat for the RQ1 investigation. Besides, while GrADS permits any proportion of data selection from the original training sets, we uniformly select 50% in the main experiments for simplicity, the exploration of varying proportions will be conducted in RQ2.

To have a thorough understanding of GrADS performance regarding different training methods, we implement full-parameter fine-tuning in our main results and investigate LoRA training in Appendix F.

5.4 Baselines

Despite the existence of numerous data selection methods, we automatically excluded those requiring manual intervention (Pan et al., 2024; Ge et al., 2024) or extensive use of advanced LLMs (like GPT-4) (Chen et al., 2023a; Liu et al., 2023). Consequently, we mainly follow the settings of Less (Xia et al., 2024) and select **Random Selection**, **BM25** (Robertson et al., 2009), **DSIR** (Xie et al., 2023), **RDS** (Zhang et al., 2018a; Hanawa et al., 2020), **LESS** (Xia et al., 2024) as baselines. The implementation of RDS also follows the setting in Xia et al. (2024). Apart from the above methods, to validate the effectiveness of gradient in GrADS, we also replace gradient with perplexity score for each training instance, denoted as **PPL**. We have some further illustration regarding those baselines in Appendix G.

5.5 Main Results

5.5.1 Domain Performance

Results in Table 1 and Table 2 show that (1) **GrADS obtained the best or second-best per-**

Domain	Method	C-Eval		GSM8k			ALPACA		Safety	Attack
		Acc.	Instruct	Acc.	BLEU	ROUGE	BLEU	ROUGE	Acc.	Acc.
	base	65.189	87.427	55.497	14.967	29.207	15.097	27.529	43.807	51.365
CMedQA	all	11.285	22.674	1.895	2.286	12.809	1.815	12.252	13.594	22.007
	rdn	14.628	**35.107**	2.880	3.006	13.410	2.443	13.305	14.446	28.503
	grads	**21.345**	33.293	**4.700**	**5.572**	**18.082**	**3.340**	**15.626**	**24.560**	**33.656**
LawQA	all	24.201	8.067	15.466	9.985	20.917	8.987	20.055	23.571	39.961
	rdn	30.305	11.846	26.384	10.772	22.049	9.486	19.889	29.378	44.938
	grads	**31.206**	**12.762**	**26.547**	**10.913**	**22.368**	**9.506**	**20.145**	**32.596**	**48.452**
FinQA	all	10.756	21.802	0.758	0.855	8.266	0.895	8.864	6.480	10.921
	rdn	15.77	28.488	0.758	0.795	7.665	1.073	9.563	9.688	15.141
	grads	**25.250**	**47.359**	**1.373**	**2.416**	**10.838**	**1.873**	**10.974**	**19.139**	**25.889**

Table 3: Catastrophic forgetting results of Qwen1.5-7B. We select 50% of data for *rdn* and *grads*.

Method	Qwen1.5-1.8B Gradients			Qwen1.5-14B Gradients		
	BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR
base	3.308	11.443	14.954	3.308	11.443	14.954
all	12.518	25.451	20.183	12.518	25.451	20.183
rdn	10.674	23.758	18.337	10.674	23.758	18.337
ppl	11.913	24.272	22.136	12.370	24.854	21.348
less	13.549	24.877	24.863	13.838	24.895	24.572
grads	**14.169**	**25.844**	**25.739**	**14.371**	**25.925**	**26.673**

Table 4: The left side selects data via Qwen1.5-1.8B gradients and fine-tuned on Qwen1.5-14B. The right side is selected via Qwen1.5-14B and fine-tuned on Qwen1.5-14B. The *base*, *all*, and *rdn* are all based on Qwen1.5-14B, so they share the same results. We select 50% of the data for *rdn*, *ppl*, *less*, and *grads*.

formance over almost all domains in the experiments. Notably, with only 50% of the data, GrADS has achieved remarkable improvements on BLEU and METEOR metrics, registering an average gain of 28.08% and 25.57% respectively, compared to LLMs fine-tuned on the entire dataset. Considering that the question-answering tasks require domain expertise, the higher BLEU, and METEOR indicate that the LLMs advance in both accuracy and richness of professional expression. Apart from that, the improvement on ROUGE-L also indicates that the LLMs have considerable enhancements in terms of long-sequence content coherence and comprehensive information coverage.

Besides, we found that (2) **GrADS is not sensitive to model initialization and model architecture, demonstrating very strong robustness.** Although Llama3 was mainly pre-trained on English datasets and perform poorly when it comes to Chinese set (as the *base* experiment of Llama3 indicates), GrADS also substantially improve its performance as what it did for those well-pretrained Chinese background LLMs (Qwen and ChatGLM).

Meanwhile, (3) **GrADS has attention on the domain characteristics when selecting data, namely self-adaptive**. When we sort all training instances by gradient magnitude in ascending order, the average percentiles of selected data for CMedQA, LawQA, and FinQA are 35.8%, 27.4%, and 28.9%, respectively. In more specialized domains such as medicine where all base LLMs perform poorly, GrADS inclines to select instances with larger gradients (the harder ones).

5.5.2 Catastrophic Forgetting

To keep the paper reasonably concise, we only present the results of Qwen1.5-7B-Chat regarding the catastrophic forgetting problem on the general capabilities evaluation datasets in the main text, Table 3. For the results of other models, please refer to Appendix F. Compared to LLMs fine-tuned on the entire dataset, **GrADS brings substantial mitigation on CF**, i.e. 82.2%, 79.5%, 41.8%, 104.8%, 70.4% improvements for C-Eval, GSM8K, ALPACA Instruct, Typical Safety, and Instruct Attack.

From the domain perspective, we observe improvements of 79.3%, 28.8%, and 112.5% on Medical, Legal, and Financial, respectively. Nevertheless, in medical (20k) and financial domains (40k) with larger training volumes, the gain of GrADS in alleviating CF problems is extremely significant.

5.6 Indepth Analysis

5.6.1 RQ1: GrADS Generalizability

The scaling law indicates that Larger models are significantly more sample-efficient (Kaplan et al., 2020; Zhang et al., 2024), so it is crucial to investigate whether GrADS is still valid in larger LLMs. Therefore, we selected Qwen1.5-14B and FinQA as our illustrative case. Meanwhile, to validate the transferability of GrADS, we initially train on Qwen1.5-1.8B for one epoch to acquire gradi-

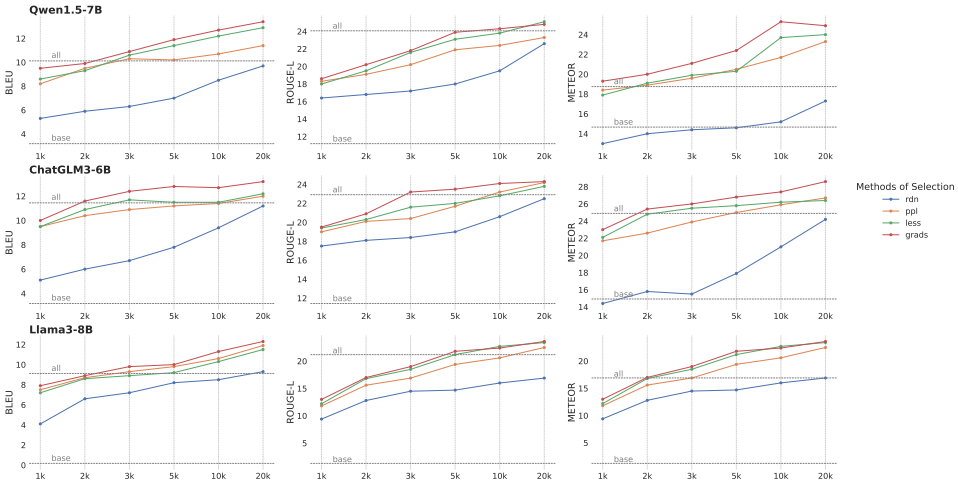

Figure 3: Experiments of fine-tuning Qwen1.5-7B, ChatGLM3-6B, Llama3-8B with subsets of different sizes selected from FinQA. Baselines 'base' and 'all' represent performances of the base models without SFT and the models fine-tuned on the entire data.

ents for each instance and subsequently employ GrADS and other gradient-based baseline methods for data selection. Thereafter, the selected subdata is fine-tuned on Qwen1.5-14B. We present more experiments regarding the transferability of GrADS across various LLMs in Appendix C.

The results presented in Table 4 demonstrate that GrADS not only **remains effective for larger LLMs** such as Qwen1.5-14B but also **can be applied across different LLMs**. Both findings highlight the strong generalizability of the GrADS method, offering exciting insights for researchers in the field of LLMs.

For instance, when confronted with voluminous training data, researchers can first leverage smaller LLMs, applying the GrADS strategy, before refining with relatively more efficacious larger LLMs. This strategy optimizes resource utilization, enabling the attainment of superior model performance while mitigating computational expenses – a pivotal consideration in large-scale machine-learning endeavors.

5.6.2 RQ2: GrADS Robustness

This section extends the main results by selecting 1k, 2k, 3k, 5k, and 10k training instances from FinQA with GrADS alongside other baseline approaches. These new experiments extend our prior analysis that was based on a 20k (50%) selection, offering a broader perspective on GrADS' performance across varying data volumes. The experiment results are provided in Figure 3.

The results in Figure 3 indicate that **the advantage of GrADS becomes even more evident in identifying subsets of smaller proportions**. In most cases, with merely 2.5%-5% (1k-2k) training instances, the GrADS has obtained comparable performance with those on full dataset. This finding holds immense implications for practical applications, showcasing a staggering efficiency-cost benefit ratio that could significantly transform the landscape of data utilization in language model tuning.

6 Conclusion

In this paper, to improve the fine-tuning efficiency and mitigate catastrophic forgetting simultaneously, we develop an adaptive gradient-aware data selection method, GrADS. Inspired by the insight that not all training data are helpful, GrADS integrates gradients extracted from the Embedding layer and LM Head layer and introduces self-guided criteria embracing statistic distributions to recognize the model's most desired data. Experimental results carried out on various LLMs and domain-specific datasets provide empirical evidence of the efficacy, efficiency, and cost-effectiveness of GrADS. Apart from extraordinary performance on domain-specific specialities, GrADS substantially mitigates catastrophic forgetting to preserve the general capabilities that the base LLMs mastered. Moreover, extensive analyses reveal that GrADS is also valid in the learning process of LoRA training, and can be scaled up to larger LLMs, delineating its great versatility and potential for generalizability.

Limitations

In this paper, we introduce the GrADS method, which aims to enhance the efficiency of domain-specific fine-tuning. While extensive experiments validate the effectiveness of GrADS, our implementation was constrained by computational resource limitations, preventing us from applying GrADS to larger-scale language models (LLMs) with parameter sizes of 30B or 72B. Nevertheless, our focus primarily lies in resource-constrained scenarios; thus, experiments conducted with models ranging from 1.8B to 14B parameters are deemed sufficiently informative for our study. Investigations of GrADS on larger LLMs can be considered for future research endeavors.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. *arXiv preprint arXiv:2307.06290*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023a. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *Preprint*, arXiv:2004.12651.

Wei Chen, Qiushi Wang, Zefei Long, Xianyin Zhang, Zhongtian Lu, Bingxuan Li, Siyuan Wang, Jiarong Xu, Xiang Bai, Xuanjing Huang, and Zhongyu Wei. 2023b. Disc-finllm: A chinese financial large language model based on multiple experts fine-tuning. *arXiv preprint arXiv:2310.15205*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023c. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, 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. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual pre-training mitigates forgetting in language and vision. *Preprint*, arXiv:2205.09357.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *Preprint*, arXiv:2306.16092.

Shizhe Diao, Tianyang Xu, Ruijia Xu, Jiawei Wang, and Tong Zhang. 2023. Mixture-of-domain-adapters: Decoupling and injecting domain knowledge to pretrained language models memories. *arXiv preprint arXiv:2306.05406*.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Hongxia Ma, Li Zhang, Hao Yang, and Tong Xiao. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. *arXiv preprint arXiv:2402.18191*.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. 2023. Understanding in-context learning via supportive pretraining data. *arXiv preprint arXiv:2306.15091*.

Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. 2020. Evaluation of similarity-based explanations. *arXiv preprint arXiv:2006.04528*.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023a. Lawyer llama. https://github.com/AndrewZhe/lawyer-llama.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023b. Lawyer llama technical report. *Preprint*, arXiv:2305.15062.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023c. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Prakhar Kaushik, Alex Gain, Adam Kortylewski, and Alan Yuille. 2021. Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping. *Preprint*, arXiv:2102.11343.

Zixuan Ke. 2024. *Continual Learning with Language Models*. Ph.D. thesis, University of Illinois at Chicago.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, et al. 2023. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *arXiv preprint arXiv:2309.06256*.

Chengyuan Liu, Shihang Wang, Yangyang Kang, Lizhi Qing, Fubang Zhao, Changlong Sun, Kun Kuang, and Fei Wu. 2024. More than catastrophic forgetting: Integrating general capabilities for domain-specific llms. *arXiv preprint arXiv:2405.17830*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2024. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *Preprint*, arXiv:2308.08747.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Xingyuan Pan, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, and Shanbo Cheng. 2024. G-dig: Towards gradient-based diverse and high-quality instruction data selection for machine translation. *arXiv preprint arXiv:2405.12915*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023. Safety assessment of chinese large language models. *arXiv preprint arXiv:2304.10436*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. *Preprint*, arXiv:2310.14152.

Bogdan M Wilamowski, Nicholas J Cotton, Okyay Kaynak, and GÜnhan Dundar. 2008. Computing gradient vector and jacobian matrix in arbitrarily connected

neural networks. *IEEE Transactions on Industrial Electronics*, 55(10):3784–3790.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227.

Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Qian Wang, and Dinggang Shen. 2023. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Kailai Yang, Tianlin Zhang, Ziyan Kuang, Qianqian Xie, and Sophia Ananiadou. 2023. Mentalllama: Interpretable mental health analysis on social media with large language models. *arXiv preprint arXiv:2309.13567*.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. 2024b. Self-distillation bridges distribution gap in language model fine-tuning. *Preprint*, arXiv:2402.13669.

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*.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *Preprint*, arXiv:2309.11325.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Huatuogpt, towards taming language model to be a doctor. *Preprint*, arXiv:2305.15075.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018a. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

Sheng Zhang, Xin Zhang, Hui Wang, Lixiang Guo, and Shanshan Liu. 2018b. Multi-scale attentive interaction networks for chinese medical question answer selection. *IEEE Access*, 6:74061–74071.

Xuanyu Zhang and Qing Yang. 2023. Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 4435–4439, New York, NY, USA. Association for Computing Machinery.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. Sapt: A shared attention framework for parameter-efficient continual learning of large language models. *Preprint*, arXiv:2401.08295.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Consistency between GPT-4o and Human Evaluation

To ensure the validity of GPT-4o's assessments, we conduct supplementary manual evaluations. For the GPT-4o's response quality scores in the main results, we sampled 200 question-answer pairs rated by GPT-4o and enlisted the expertise of three professional data annotators to independently score the response quality on a scale of 1 to 5, in alignment with GPT-4o's scoring criteria. Subsequently, we calculated the average score for the three individuals and applied rounding to the nearest integer. We find that there are 147 samples for which the scores given by GPT-4o completely align with those of the annotators, and the scores from both sides yields a Pearson product-moment correlation coefficient of 0.79. This indicates a substantial agreement between the GPT-4o and human evaluations of response quality.

Meanwhile, for the safety judgement of catastrophic forgetting experiments in main results, we also sample 200 instances for each category, i.e.,

Base Model	Method	CMedQA			LawQA			FinQA		
		BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR
Qwen1.5-1.8B	base	1.547	8.228	11.169	9.860	19.178	24.505	1.888	7.911	12.061
	all	3.339	16.318	11.792	15.606	26.124	25.950	9.358	21.014	17.127
	rdn	3.515	16.005	11.233	14.973	25.587	24.653	8.491	20.726	16.440
	ppl	4.147	17.074	13.996	16.735	26.024	30.613	11.186	21.673	20.360
	less	4.392	16.950	14.011	17.067	25.962	31.116	11.031	21.144	20.468
	grads	**4.852**	**18.218**	**14.439**	**18.754**	**26.339**	**33.688**	**11.875**	**22.106**	**21.732**
Qwen1.5-7B	base	2.627	12.180	10.860	9.066	20.050	21.392	3.188	11.194	14.669
	all	3.813	17.327	12.276	16.090	27.603	27.472	10.120	24.067	18.757
	rdn	3.548	16.776	11.954	15.856	27.288	26.810	9.686	22.621	17.276
	ppl	4.832	18.215	**15.012**	18.874	27.229	**35.680**	11.302	22.276	22.295
	less	4.711	**18.976**	13.690	17.822	27.144	29.539	10.157	22.531	21.464
	grads	**5.012**	18.664	14.787	**19.914**	**29.018**	34.457	**12.782**	**24.064**	**23.980**
Qwen1.5-14B	base	2.934	12.338	11.458	9.968	20.356	25.472	3.308	11.443	14.954
	all	4.034	17.577	12.760	16.688	27.763	28.381	12.518	25.451	20.183
	rdn	3.738	17.224	12.278	16.169	27.234	27.758	10.674	23.758	18.337
	ppl	5.038	18.215	15.339	19.359	28.080	**35.954**	11.913	24.272	22.136
	less	5.214	**19.565**	14.874	18.920	28.352	32.826	13.549	24.877	24.863
	grads	**5.766**	19.018	**15.862**	**20.314**	**30.523**	35.877	**14.169**	**25.844**	**25.739**
ChatGLM3-6B	base	2.568	11.274	15.634	7.966	19.733	19.011	3.174	11.437	14.926
	all	4.297	17.432	16.722	16.673	**28.016**	28.519	11.454	22.918	24.898
	rdn	4.512	16.674	16.482	16.453	27.576	27.864	11.216	22.450	24.233
	ppl	5.035	17.765	17.930	18.886	27.569	31.783	12.295	22.719	26.327
	less	4.888	18.026	17.651	17.379	27.468	29.136	11.843	23.170	25.089
	grads	**5.656**	**18.375**	**19.016**	**19.918**	27.779	**34.154**	**13.328**	**24.434**	**27.977**
Llama3-8B	base	0.026	0.249	0.291	0.259	1.905	2.164	0.178	1.293	1.225
	all	3.332	16.415	11.061	15.272	24.301	27.033	9.116	21.190	16.913
	rdn	3.265	15.884	10.798	15.552	24.688	26.476	9.337	22.654	16.870
	ppl	4.365	17.328	**14.426**	18.225	25.964	32.387	11.454	21.998	22.759
	less	4.186	17.684	13.631	17.271	**26.754**	31.850	11.048	21.753	22.833
	grads	**4.774**	**18.125**	14.116	**18.941**	26.376	**33.385**	**12.028**	**23.366**	**23.300**

Table 5: Experiment results of implementing GrADS with **Qwen1.5-1.8B**, and leverage the selected data for SFT on **Qwen1.5-1.8B** itself and **other larger LLMs**. We select 50% of data for training with *rdn*, *ppl*, *less* and *grads*.

Typical Safety and Instruction Attack, respectively. With the same approach, we measure the consistency between GPT-4o's judgments and the three annotators evaluations. We obtain a correlation coefficient score of 0.879 for Typical Safety and a correlation coefficient score of 0.815 for Instruction Attack.

B GrADS Transferability on Different LLMs

In this section, we adhere to the setting in RQ1 which initially train on Qwen1.5-1.8B for one epoch to acquire gradients and employ Qwen1.5-1.8B itself or other LLMs for fine-tuning. Our findings in Table 5 reveal that data selected using Qwen1.5-1.8B in conjunction with GrADS not only prove effective for larger LLMs of the same architecture (Qwen-1.5-7B and Qwen1.5-14B), but also yield substantial improvements for larger LLMs of different architectures, including ChatGLM3-6B and Llama3-8B. This experimentation **further validates the transferability of the GrADS methodology**.

C Ablation Study

In this section, we conducted an ablation study using Qwen1.5-7B as a case example across three domains. Specifically, we examined the impacts of incorporating only the gradients from the Embed Layer or only from the LM Head Layer, namely *w/o lmhead* and *w/o embed*. Additionally, we also investigate how LLMs perform when they are trained on data selected from the half with the smallest gradients (*tail*), the largest half (*top*), and the middle half (*mid*). Meanwhile, as in GrADS we add the gradients from the LM Head Layer and Embed Layer directly, and we also explore substitute integration methods. In Table6, *weight* refers to

Method	Ablation	CMedQA			LawQA			FinQA		
		BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR
	base	2.627	12.180	10.860	9.066	20.050	21.392	3.188	11.194	14.669
	all	3.813	17.327	12.276	16.090	27.603	27.472	10.120	24.067	18.757
	rdn	3.548	16.776	11.954	15.856	27.288	26.810	9.686	22.621	17.276
GrADS	w/o lmhead	4.435	17.019	14.455	18.866	25.759	33.887	11.945	**25.012**	23.130
	w/o embed	5.011	17.875	14.986	19.305	26.874	32.491	12.455	24.170	23.843
	top grad	2.986	15.874	10.039	14.012	25.183	23.456	6.758	20.417	14.009
	tail grad	4.736	16.689	15.006	19.424	26.780	33.699	12.274	23.510	23.356
	mid grad	4.630	17.492	14.890	17.764	27.733	33.383	10.429	22.304	22.285
	weight	4.832	17.316	14.284	19.259	26.671	33.034	12.218	23.313	23.409
	weightr	4.727	17.134	13.855	18.736	26.033	32.682	11.769	22.237	23.030
GRADS	ours	**5.372**	**18.496**	**15.396**	**20.270**	**28.026**	**35.985**	**13.364**	24.822	**24.872**

Table 6: Ablation Study. We select 50% of data for training except for *base* and *all*.

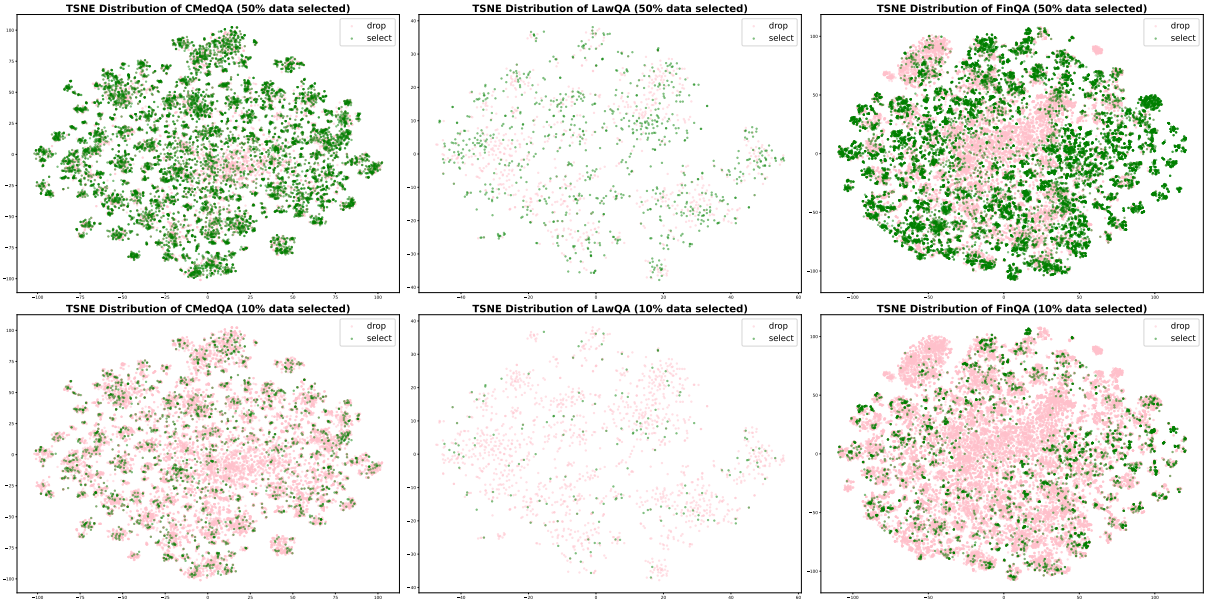

Figure 4: Semantic distribution of training instances. The green dots indicate selected instances whereas the red dots indicate dropped instance.

the gradients from the Embed Layer and LM Head Layer that are normalized and summed to derive a gradient distribution. Besides, *weightr* entails ranking the gradients of each instance from the Embed Layer and LM Head Layer in descending order and summing their ranks' reciprocals to obtain the distribution. Subsequently, both *weight* and *weightr* utilize the same data selection criteria as GrADS.

The experimental results indicate that our original GrADS consistently achieves optimal or suboptimal performance, thereby validating the rationale behind our methodological design.

D Data Diversity

One concern is that selecting data based on the highest probability density might compromise the diversity of the chosen dataset, an aspect that is essential for effective large language model (LLM) training. Therefore, in this section, we apply GrADS with Qwen1.5-7B for data selection across three domains. To obtain the semantic distribution of training instances, we apply Text_Embedding_V3 [4] for embedding representation and TSNE (Van der Maaten and Hinton, 2008) technique for dimensionality reduction and visualization.

The results illustrated in Figure 4 suggest that the probability density of gradients has few relevance to semantic meanings. Notably, the data selected by GrADS maintain considerable diversity, regardless of the situation of 50% or 10% selection. As we have discussed in the Introduction section, LLMs can perform like absolutely rational college

[4] https://www.alibabacloud.com/help/en/model-studio/developer-reference/text-embedding-synchronous-api

Domain	Method	C-Eval		GSM8k			ALPACA		Safety	Attack
		Acc	Instruct	Acc	BLEU	ROUGE	BLEU	ROUGE	Acc	Acc
	base	54.360	73.328	46.020	18.593	33.002	16.362	28.321	44.681	50.686
CMedQA	all	8.794	5.523	3.942	4.271	16.108	4.782	16.971	16.783	23.068
	rdn	15.480	18.023	6.823	5.811	18.120	6.636	19.507	16.879	27.863
	grads	**16.480**	**19.695**	**10.008**	**6.468**	**19.098**	**7.810**	**20.765**	**25.679**	**31.484**
LawQA	all	30.451	25.363	34.572	14.833	28.227	13.966	25.468	26.137	40.017
	rdn	32.756	**38.227**	36.012	**16.103**	**29.974**	**15.284**	**27.242**	28.995	41.983
	grads	**33.717**	37.974	**37.225**	15.709	29.080	13.637	26.671	**37.681**	**42.036**
FinQA	all	13.953	2.947	1.365	1.418	9.869	2.832	12.448	7.416	11.678
	rdn	18.823	5.794	2.578	2.032	11.751	3.638	14.495	10.861	14.884
	grads	**26.017**	**19.089**	**4.250**	**3.717**	**15.649**	**4.911**	**16.776**	**21.025**	**24.481**

Table 7: Supplementary experiments of **Catastrophic Forgetting** on **ChatGLM3-6B**. We select 50% of data for training with *rdn* and *grads*

Domain	Method	C-Eval		GSM8k			ALPACA		Safety	Attack
		Acc	Instruct	Acc	BLEU	ROUGE	BLEU	ROUGE	Acc	Acc
	base	46.657	97.832	58.226	3.903	16.089	3.229	7.443	27.150	44.167
CMedQA	all	0.291	0.000	0.682	0.894	9.215	0.861	9.317	7.061	6.333
	rdn	0.390	0.036	0.076	0.777	8.749	0.860	9.310	10.535	5.583
	grads	**0.509**	**0.073**	**0.758**	**1.679**	**10.827**	**1.131**	**10.800**	**14.672**	**10.250**
LawQA	all	3.634	3.343	2.729	4.438	13.415	4.624	14.708	15.643	32.333
	rdn	4.506	0.727	3.033	4.105	12.859	4.298	13.869	22.714	37.250
	grads	**4.869**	**6.017**	**4.701**	**6.311**	**15.585**	**5.326**	**16.081**	**25.143**	**41.583**
FinQA	all	0.363	1.817	0.227	0.491	6.985	0.558	10.297	1.500	2.750
	rdn	1.438	9.084	0.455	0.683	7.019	0.667	8.091	**2.571**	**3.917**
	grads	**6.541**	**21.148**	**0.607**	**1.019**	**7.514**	**0.739**	**8.473**	1.929	3.833

Table 8: Supplementary experiments of **Catastrophic Forgetting** on **Llama3-8B**. We select 50% of data for training with *rdn* and *grads*

students who select courses they need not just what they like.

E Supplementary Experiments of Catastrophic Forgetting

In this section, we provide supplementary experimental results regarding catastrophic forgetting problem. Table 7 and table 8 illustrate the results of ChatGLM3-6B and Llama3-8B, which validate that GrADS not only substantially alleviate catastrophic forgetting for Qwen1.5-7B, but also for ChatGLM3-6B and Llama3-8B.

F Supplementary Experiments of LoRA Tuning

Apart from full parameter fine-tuning, we also investigate how GrADS would facilitate LoRA tuning. Table 9 provides the results of LoRA tuning whereas table 10, table 11, and table 12 provide the results of the catastrophic forgetting problem of Qwen1.5-7B, ChatGLM3-6B, and Llama3-8B after LoRA tuning, respectively.

Those experiments validate GrADS's effectiveness across full parameter fine-tuning and LoRA tuning. In the meantime, for those who seeking a balance between domain capabilities and general capabilities (less catastrophic forgetting), the combination of GrADS and LoRA tuning should be a good choice.

G Baseline Illustration

We present a brief introduction of our baselines in this section. **BM25** (Robertson et al., 2009) featurizes examples by their word frequency statistics (i.e., TF-IDF) to rank the training instances, and select the top k% of the training instances with the highest scores to construct Dtrain. **DSIR** (Xie et al., 2023) uses n-gram features to weight candidate training data D. We resample k% of the training instances according to the importance weights. **RDS** (Representation-based Data Selection) (Zhang et al., 2018a; Hanawa et al., 2020) uses the model's hidden representations as features for data selection. We follow the settings in Xia

Base Model	Method	CMedQA			LawQA			FinQA		
		BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR	BLEU	ROUGE	METEOR
Qwen1.5-7B	base	2.627	12.180	10.860	9.066	20.050	21.392	3.188	11.194	14.669
	all	4.075	17.739	12.966	14.580	**27.382**	31.207	7.316	20.192	15.365
	rdn	3.839	17.219	12.250	14.293	25.581	27.769	6.372	19.415	14.108
	bm25	3.555	16.875	11.208	12.937	24.837	25.981	5.709	18.793	12.283
	dsir	3.840	16.698	11.475	13.057	24.880	24.512	5.716	17.397	12.388
	rds	3.818	17.022	11.549	13.235	24.320	23.898	6.875	20.051	12.648
	ppl	4.526	17.481	13.569	14.862	24.383	25.485	7.769	20.651	16.866
	less	4.757	17.596	**14.984**	16.012	26.057	30.136	7.892	20.135	17.200
	grads	**5.018**	**18.243**	14.696	**17.963**	26.755	**32.802**	**9.103**	**21.154**	**18.848**
ChatGLM3-6B	base	2.568	11.274	10.634	7.966	19.733	19.011	3.174	11.437	14.926
	all	3.551	15.960	12.124	**12.903**	**22.514**	23.174	8.047	20.820	17.159
	rdn	3.498	15.824	11.970	10.010	20.038	21.166	8.155	19.896	17.032
	bm25	3.539	16.296	12.035	10.457	20.745	20.899	8.100	20.043	16.747
	dsir	3.667	16.187	11.892	9.964	20.819	20.451	8.269	19.803	16.760
	rds	3.256	15.517	11.389	9.854	19.899	20.016	7.079	19.266	16.148
	ppl	4.286	17.536	13.492	11.914	20.188	22.358	8.177	20.375	16.658
	less	3.932	16.774	13.758	11.616	21.089	21.648	8.524	20.793	17.617
	grads50	**4.483**	**18.216**	**14.447**	12.724	22.214	**23.857**	**8.896**	**21.301**	**17.966**
Llama3-8B	base	0.026	0.249	0.291	0.259	1.905	2.164	0.178	1.293	1.225
	all	3.138	**16.695**	11.782	16.125	**25.588**	28.327	9.336	22.480	18.751
	rdn	2.851	16.030	10.956	14.478	24.515	27.160	8.931	21.267	16.922
	bm25	2.543	15.381	9.075	13.308	21.629	25.584	7.856	20.639	14.487
	dsir	2.738	15.683	10.719	13.985	24.205	26.650	8.857	20.977	17.356
	rds	2.918	15.984	10.270	14.041	23.388	26.986	8.844	20.074	17.706
	ppl	3.326	16.540	12.016	15.427	23.958	29.836	9.328	20.890	**19.027**
	less	**3.517**	16.310	12.022	15.811	24.018	28.895	9.085	20.228	18.750
	grads	3.446	16.019	**12.527**	**16.475**	25.487	**30.699**	**9.919**	**22.807**	18.940

Table 9: Supplementary experiments of **LoRA** tuning. *base* denotes no further training implemented, *all* denotes full dataset, and otherwise we select 50% of the data for training.

et al. (2024), which computes the similarity score using Equation (2) of Xia et al. (2024) but replace the gradient features with the final layer representations of the last token of each sequence. **LESS** (Low-rank gradiEnt Similarity Search) (Xia et al., 2024) utilizes gradients as well and selects training instances based on their similarity to few-shot examples embodying a specific capability.

H Implementation Details

Our experiment is conducted on 8 A100 GPUs, each with 80G memories. All experiments are conducted with LLaMA-Factory[5] training architecture and deepspeed_z3. For all methods, we set the learning rate of 3e-5, warmup ratio of 0.1, and batch size of 8. Regarding LLMs' API, we adopt GPT-4o. For LoRA experiments, the rank is set to 16. For all randomly selected data, we set the random seed of 42. To maintain some basic instruction following capabilities for more precise evaluation (especially for *rdn* and *all*), for all catastrophic forgetting related experiments, we only report the score on the test set after 1 training epoch. For the rest of the experiments, we report the average scores on the test set after the training epochs of 1, 2, and 3.

[5]https://github.com/hiyouga/LLaMA-Factory/tree/main

Domain	Method	C-Eval		GSM8k			ALPACA		Safety	Attack
		Acc	Instruct	Acc	BLEU	ROUGE	BLEU	ROUGE	Acc	Acc
	base	65.189	87.427	55.497	14.967	29.207	15.097	27.529	43.807	51.365
CMedQA	all	**35.512**	**42.124**	22.592	7.368	23.634	5.681	19.265	23.087	31.415
	rdn	29.420	29.940	33.131	10.441	27.717	7.044	21.033	28.596	37.847
	grads	34.101	40.638	**44.806**	**14.600**	31.424	**8.644**	**22.966**	**31.138**	**42.636**
LawQA	all	34.323	32.615	53.373	14.408	28.252	14.031	26.393	28.650	41.684
	rdn	39.673	**39.598**	53.146	**14.841**	**29.121**	**14.970**	**27.211**	33.757	**50.220**
	grads	**41.307**	38.484	**53.980**	12.970	26.678	13.504	25.160	**35.766**	49.814
FinQA	all	48.365	68.870	17.664	3.528	18.015	4.178	15.660	17.174	24.269
	rdn	**50.817**	**70.653**	20.849	3.976	19.035	4.356	16.408	21.235	33.471
	grads	27.637	22.956	**28.506**	**8.582**	**26.069**	**6.000**	**18.344**	**27.451**	**36.045**

Table 10: Supplementary experiments of **Catastrophic Forgetting** after **LoRA** tuning on **Qwen1.5-7B**. We select 50% of data for training with *rdn* and *grads*

Domain	Method	C-Eval		GSM8k			ALPACA		Safety	Attack
		Acc	Instruct	Acc	BLEU	ROUGE	BLEU	ROUGE	Acc	Acc
	base	54.360	73.328	46.020	18.593	33.002	16.362	28.321	44.681	50.686
CMedQA	all	25.186	28.232	25.929	11.616	27.839	11.079	24.465	25.318	33.572
	rdn	**30.163**	29.822	30.857	13.497	29.993	12.162	25.685	31.664	38.055
	grads	28.678	**40.416**	**33.207**	**14.455**	**31.127**	**12.426**	**25.974**	**32.042**	**39.776**
LawQA	all	39.673	60.327	42.077	17.661	32.429	14.814	27.301	30.285	40.069
	rdn	39.004	**64.859**	43.821	**17.929**	**32.610**	**15.249**	**27.640**	35.460	46.734
	grads	**40.119**	64.636	**44.655**	17.895	32.517	15.065	27.446	**37.261**	**48.588**
FinQA	all	23.031	42.422	22.214	10.278	26.400	10.022	22.785	16.292	25.106
	rdn	28.158	40.416	**26.005**	11.090	27.576	10.744	23.197	22.234	30.217
	grads	**31.055**	**45.840**	24.867	**11.863**	**28.318**	**12.641**	**25.519**	**28.656**	**35.785**

Table 11: Supplementary experiments of **Catastrophic Forgetting** after **LoRA** tuning on **ChatGLM3-6B**. We select 50% of data for training with *rdn* and *grads*

Domain	Method	C-Eval		GSM8k			ALPACA		Safety	Attack
		Acc	Instruct	Acc	BLEU	ROUGE	BLEU	ROUGE	Acc	Acc
	base	46.657	97.832	58.226	3.903	16.089	3.299	7.443	27.150	44.167
CMedQA	all	2.674	1.783	15.693	6.706	21.267	3.343	14.913	15.714	20.333
	rdn	9.212	14.413	17.664	6.666	21.489	3.149	15.131	21.071	26.750
	grads	**9.509**	**18.127**	**23.730**	**8.970**	**25.342**	**5.402**	**18.493**	**22.286**	**29.167**
LawQA	all	39.376	81.278	54.814	17.542	**33.502**	**10.398**	**21.001**	24.643	31.250
	rdn	**44.428**	**92.422**	55.800	16.637	32.829	9.611	19.168	26.143	34.333
	grads	43.908	91.976	**58.302**	**17.660**	33.339	8.545	17.339	**28.643**	**40.667**
FinQA	all	22.140	28.826	30.857	6.806	25.282	5.200	17.511	7.214	14.250
	rdn	**28.158**	**54.309**	32.980	7.548	26.324	5.536	18.784	11.857	19.417
	grads	23.626	35.364	**34.117**	**9.022**	**27.687**	**6.566**	**19.203**	**14.143**	**24.083**

Table 12: Supplementary experiments of **Catastrophic Forgetting** after **LoRA** tuning on **Llama3-8B**. We select 50% of data for training with *rdn* and *grads*