Efficient Layer-wise LLM Fine-tuning for Revision Intention Prediction

Zhexiong Liu, Diane Litman
Department of Computer Science, Learning Research & Development Center
University of Pittsburgh, Pittsburgh, Pennsylvania, USA 15260
zhexiong@cs.pitt.edu, dlitman@pitt.edu

Abstract

Large Language Models (LLMs) have shown extraordinary success across various text generation tasks; however, their potential for simple yet essential text classification remains underexplored, as LLM pre-training tends to emphasize generation over classification. While LLMs with instruction tuning can transform classification into a generation task, they often struggle to categorize nuanced texts. One such example is text revision, which involves nuanced edits between pairs of texts. Although simply fine-tuning LLMs for revision classification seems plausible, it requires a large amount of revision annotations, which are exceptionally expensive and scarce in the community. To address this issue, we introduce a plug-and-play layer-wise parameter-efficient fine-tuning (PEFT) framework, i.e., IR-Tuning, which fine-tunes a subset of important LLM layers that are dynamically selected based on their gradient norm distribution, while freezing those of redundant layers. Extensive experiments suggest that IR-Tuning surpasses several layer-wise PEFT baselines over diverse text revisions, while achieving fast convergence, low GPU memory consumption, and effectiveness on small revision corpora.

1 Introduction

Revision is regarded as an important part of writing because it commonly improves the final written work (Fitzgerald, 1987); however, as complexity in the process of revision increases, it becomes more difficult to interpret whether revision is in line with the writer's actual intentions (Sommers, 1980; Hayes and Flower, 1986). Particularly, determining the intentions between nuanced revisions is challenging for computational models. For example, Figure 1 shows the same piece of text revised with different intentions, of which one added a word *furthermore* to improve coherence, and the other modified a word from *style* to *visual* to enhance clarity. Prior work (Du et al., 2022a,b) has used small language models, e.g., RoBERTa (Liu et al., 2019), to learn these intentions, which failed to identify complex revisions due to the model's limited capability (Skitalinskaya and Wachsmuth, 2023). Hence, stronger models are needed to recognize diverse revision patterns.

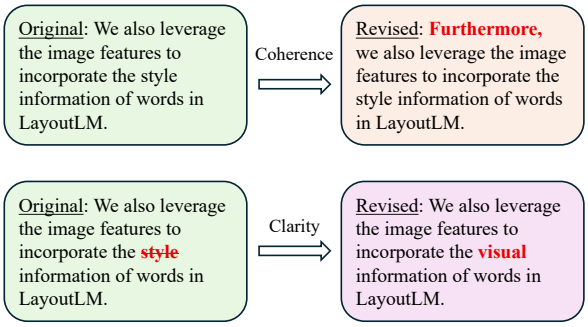

Figure 1: An example where the same original text is revised based on different intentions. The examples are from the ITERATER corpus (Du et al., 2022b).

Recently, large language models (LLMs) have achieved impressive success in multiple NLP tasks, such as text summarization (Takeshita et al., 2024), question answering (Peng et al., 2024), and concept reasoning (Li et al., 2024a). However, their application in revision tasks remains underexplored. This might be because revision tasks require LLMs to learn an iterative process involving additions, deletions, and modifications, each driven by distinct intentions, but LLMs are mostly pre-trained to generate final texts. Although Shu et al. (2024) explore LLMs with instruction tuning and reinforcement learning for text revision, they focus on supervised fine-tuning using massive datasets, which becomes inefficient and expensive as the data size grows. While Ruan et al. (2024a) fine-tune LLMs using a parameter-efficient fine-tuning (PEFT) method, i.e., QLoRA (Dettmers et al., 2023), to minimize expenses, it requires training an adapter for every LLM layer, thus becoming less efficient as the number of LLM layers increases.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 15319–15334
November 4-9, 2025

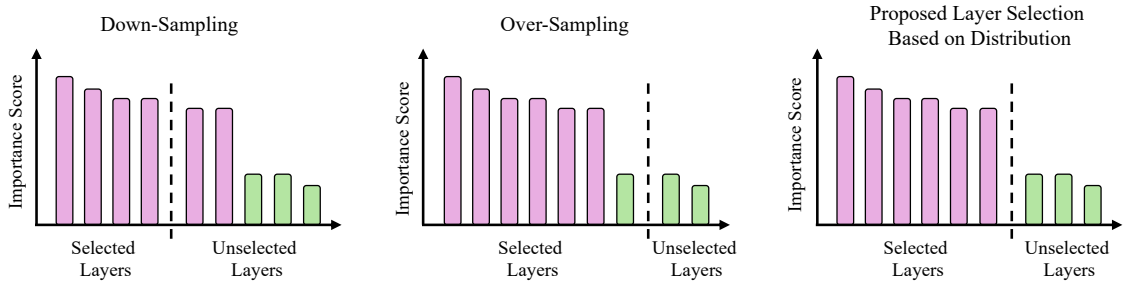

Figure 2: The illustration of over-sampling and down-sampling issues in the sampling-based layer-wise PEFT method (Yao et al., 2024) and our proposed layer selection method, where the purple and green bars represent high and low layer-wise importance scores introduced in Section 4.2, respectively.

Although standard PEFT methods (Hu et al., 2022, 2023) proved to be efficient, they attempt to learn entire LLM layers, despite prior research indicating that LLM layers contribute differently to downstream tasks (Pan et al., 2024; Zhao et al., 2024). Recently, Yao et al. (2024) have introduced an Importance-aware Sparse Tuning (IST) that samples a subset of important LLM layers for PEFT. However, the number of its important layers remains fixed throughout the fine-tuning, leading to suboptimal layer selection. For example, important layers might be overlooked (down-sampling) if sampling a small number of layers, or unimportant layers might be included (over-sampling) if selecting too many layers. Figure 2 illustrates these scenarios, where down-sampling selects four out of six important layers and over-sampling selects an extra unimportant layer. Also, the number of important layers can vary as the fine-tuning progresses, but IST failed to address it. In contrast, we propose Importance Redundancy Tuning, i.e., IR-Tuning[1], which uses an algorithm to select important layers for fine-tuning while freezing redundant ones, where the fine-tuned layers are dynamically determined based on the distribution of LLM layer-wise gradient norms throughout the fine-tuning.

To evaluate the effectiveness of the IR-Tuning on text revision tasks, we work on three research questions: **RQ1**: How do LLM layers contribute differently to revision intention prediction? **RQ2**: How can we dynamically select important layers for fine-tuning LLMs on small corpora? **RQ3**: Are contextualized instructions helpful for LLMs to learn revision intentions? In particular, we make the following contributions:

- We propose the first work that uses layer-wise PEFT to address an essential text revision task.
- We develop an algorithm to dynamically select a subset of important LLM layers for fine-tuning.
- We demonstrate the feasibility of efficiently fine-tuning LLMs with small revision corpora.

[1] https://github.com/ZhexiongLiu/IR-Tuning

2 Related Work

2.1 Revision Intention in NLP

Text revision primarily focuses on analyzing revision intention to understand human edits (Zhang and Litman, 2015; Shibani et al., 2018; Afrin et al., 2020; Kashefi et al., 2022; Du et al., 2022b; Chong et al., 2023; Mita et al., 2024; Jourdan et al., 2024). However, identifying intention is challenging, largely because collecting annotated revision corpora are expensive (Zhang et al., 2017; Anthonio et al., 2020; Spangher et al., 2022; Du et al., 2022b; D'Arcy et al., 2024). Prior work (Zhang et al., 2016; Yang et al., 2017; Afrin and Litman, 2018; Kashefi et al., 2022; Afrin et al., 2020; Afrin and Litman, 2023) has developed feature-based computational methods, which cannot generalize to other corpora. Although Du et al. (2022b); Jiang et al. (2022) have trained small language models, such as RoBERTa (Liu et al., 2019), for identifying revision intention, they have struggled to learn complex revision patterns due to the models' capabilities. While more recent work either prompts LLMs with instruction (Ruan et al., 2024b) or fine-tunes LLMs using standard PEFT (Ruan et al., 2024a), they have not investigated how LLM layers contribute to revision tasks. In contrast, we propose a novel layer-wise PEFT method that facilitates LLM fine-tuning more effectively and efficiently using small revision corpora.

2.2 Parameter-Efficient Fine-tuning

PEFT methods offer promising solutions for fine-tuning LLMs in a computationally efficient manner. They update a small fraction of LLM parameters

	Evidence Revision				Reasoning Revision				Total
	Relevant	Irrelevant	Repeated	Others	LCE	not LCE	Commentary	Others	
Add	1,774	397	225	136	1,069	317	425	299	4,642
Delete	598	102	45	56	318	121	194	70	1,504
Modify	138	26	6	53	105	23	41	55	447
Total	2,510	525	276	245	1,492	461	660	424	6,593

Table 1: The statistics of revisions in the ArgRevision corpus.

by using adapter-based techniques (Houlsby et al., 2019; Wang et al., 2022; Lei et al., 2024), low-rank adaptations (Hu et al., 2022; Edalati et al., 2025; Liu et al., 2024), and prompt-based approaches (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2022). While standard PEFT implements an identical design across all LLM layers, it cannot explore each layer's unique contribution to downstream tasks, despite prior work suggesting layer redundancy in pre-trained models (Lan et al., 2020; Sajjad et al., 2023; Zhang et al., 2023; Elhoushi et al., 2024). Recently, Kaplun et al. (2023) develop a greedy search to select useful layers, Pan et al. (2024) instead randomly select layers, and Zhu et al. (2024) pick layers based on front-to-end or end-to-front heuristics, all for fine-tuning LLMs, but these methods either need expensive computations or utilize simple strategies that could cause downgraded performance on complex tasks. While prior work (Yao et al., 2024; Zhou et al., 2025) sampled a subset of important layers based on importance scores, their methods could cause over-sampling and down-sampling issues, as shown in Figure 2. Although Wei et al. (2025) utilize an unrolled differentiation method to identify the most useful LLM layers, they require expensive computation on hyperparameter optimization. In contrast, we propose a plug-and-play PEFT framework that utilizes an efficient algorithm to dynamically select a subset of important LLM layers based on their gradient norm distribution in each fine-tuning iteration, ensuring high-gradient layers are prioritized for update.

3 Corpora

Text revision is rarely annotated because annotation is expensive; thus, we collect an argument revision corpus called ArgRevision, which includes essays written by elementary and middle school students with limited writing skills. We use this corpus to study revisions in argumentation, which is critically needed for argument writing evaluation research (Li et al., 2024b; Correnti et al., 2024). Additionally, we use a publicly available revision corpus (Du et al., 2022b) that contains articles written by experienced authors, editors, and researchers. We conduct experiments using text revisions from both skilled and less skilled writers.

	Space Essays		MVP Essays	
	RER#	Kappa	RER#	Kappa
Reasoning	148	0.86	135	0.84
Evidence	108	0.89	136	0.80

Table 2: The annotation agreement for reasoning and evidence RER annotations for a batch of 117 essays in our prior studies (Liu et al., 2023a).

3.1 Argument Revision

The ArgRevision corpus contains pairs of essays before and after revisions, collected using our deployed automated writing evaluation system (Liu et al., 2025). The corpus contains 990 essay drafts written by students in grades four to eight from schools in Pennsylvania and Louisiana, who are taking the Response to Text Assessment (Correnti et al., 2013). 172 students wrote essays in response to a prompt about the United Nations' Millennium Villages Project (MVP). Afterward, the students revised their essay drafts in response to feedback provided by the system, and each student completed three drafts, resulting in 344 pairs of essay drafts, e.g., draft1-draft2 and draft2-draft3. Another 158 students did the same tasks in response to another essay prompt about Space Exploration (Space), yielding 316 pairs of essay drafts. We combine the essays from the two prompts as students share similar writing skills, and the scoring rubric is consistent across the prompts.

We preprocess collected essays for revision annotation. First, the sentences from the original and revised drafts (e.g., draft1-draft2, draft2-draft3) are aligned into pairs of original sentences and revised sentences using a sentence alignment tool Bertalign (Liu and Zhu, 2022). The aligned pairs are automatically labeled with *no change* if the original sentence and the revised sentence are the same, *modify* if the original and the revised sentence are not empty but not the same, *add* if the original sentence is empty but the revised sentence is not, or

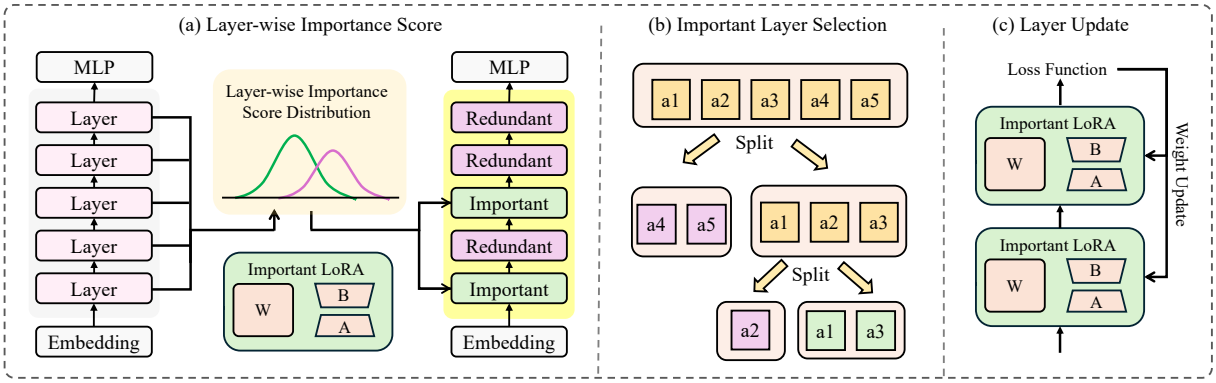

Figure 3: The components of the proposed Importance Redundancy Tuning (IR-Tuning). The IR-Tuning includes a module that updates layer-wise importance scores on all the layers, then a layer-selection module to select important layers for fine-tuning. Finally, the weights in the selected layers are updated with fine-tuning through LoRA.

delete if the revised sentence is empty but the original sentence is not. The changed alignments are classified into *surface* (meaning-preserving) and *content* (meaning-altering) revisions by a BERT-based (Devlin et al., 2019) classifier trained on a college-level revision corpus (Kashefi et al., 2022) with an F1 score of 0.96. Following Afrin et al. (2020), we use the Revisions of Evidence and Reasoning (RER) scheme to annotate *content* revisions into evidence and reasoning revisions. Specifically, evidence revisions are annotated with *relevant, irrelevant, repeated* evidence and *others*, and reasoning revisions are annotated with *linked claim-evidence (LCE), not LCE, commentary* and *others*. Here, we do not use the *others* label as it contains a mixture of revisions based on multiple rarely annotated intentions. Table 1 shows annotated intention statistics. The annotation is done by expert annotators, and each revision is annotated by one expert. Some annotators participated in the same annotation tasks in our prior studies (Liu et al., 2023a), of which the two-annotator agreements on a batch of 117 student essays about both MVP and Space prompts are shown in Table 2. The annotation examples are shown in Table 8 in the Appendix.

3.2 Article Revision

We use a publicly available revision corpus named ITERATER (Du et al., 2022b), which annotates 4,018 text revisions from Wikipedia, ArXiv, and news articles. Wikipedia articles are written by editors who often focus on improving the clarity and structure of articles. ArXiv articles are written by scientific authors who generally revise hypotheses, experimental results, and research insights. News articles are written by editors interested in clarity and readability. The ITERATER corpus contains six intention labels: *clarity*, *fluency*, *coherence*, *style*, *meaning-changed*, and *others*. Here, we do not use the *others* label as it denotes unrecognizable intentions. The revision statistics and examples are shown in Table 9 and Table 10 in the Appendix.

4 Method

We propose a novel plug-and-play layer-wise PEFT framework, i.e., IR-Tuning, which attempts to update the weights of important LLM layers while keeping those of redundant layers unchanged. Figure 3 shows the framework components.

4.1 Layer-wise Importance Score

Supposing an LLM $\mathcal{M}$ consisting of l transformer layers m_i, $\mathcal{M} = \{m_i\}_{i=1}^{l}$, our objective is to generate a subset $\mathcal{S}$ of $\mathcal{M}$ as important layers and the remaining set $\bar{\mathcal{S}}$ as redundant layers. Inspired by prior work (Zhang et al., 2024), we use layer-wise gradient norm $\mathcal{N} = \{a_i\}_{i=1}^{l}$ to denote LLM layers' importance scores for two reasons. First, layers with high gradient norms suggest large weight updates that make key contributions to the rapid optimization of the loss function along the gradient direction, which can facilitate efficient gradient descent. Second, larger gradient norms may carry more information relevant to downstream tasks, which makes LLM layers informative during fine-tuning. These hypotheses have been confirmed and utilized in prior work (Lee et al., 2023; Zhang et al., 2024). Empirically, we use a threshold γ to split $\mathcal{M}$ into $\mathcal{S}$ and $\bar{\mathcal{S}}$:

$$\mathcal{S} = \{m_i \mid a_i > \gamma\}, \quad (1)$$

$$\bar{\mathcal{S}} = \{m_i \mid a_i \leq \gamma\}, \quad (2)$$

where γ is obtained by a layer selection algorithm described in the next section.

4.2 Important Layer Selection

Unlike prior work that selects the top n important layers from a ranked score list (Yao et al., 2024), we formulate the layer selection as a distribution divergence problem, arguing that the important layers and redundant layers are from different distributions. Although this idea has been used in Liu et al. (2023b) for splitting a large amount of meta-learning tasks into easy and hard tasks, we instead focus on a smaller set of LLM layer splitting based on their importance scores. Here, we have a null hypothesis $\mathcal{H}_0$: the importance scores of layers in $\mathcal{M}$ follows a single distribution, and an alternative hypothesis $\mathcal{H}_1$: there exists a subset $\mathcal{S}$ of $\mathcal{M}$, where the importance scores of layers in $\mathcal{S}$ follow a different distribution from those of the remaining layers $\bar{\mathcal{S}}$. Therefore, the optimal set $\mathcal{S}^*$ can be obtained by solving the optimization problem:

$$\mathcal{S}^* = \text{argmax}_{\mathcal{N}} \log \frac{\text{Likelihood } (\mathcal{H}_1 \mid \mathcal{N})}{\text{Likelihood } (\mathcal{H}_0)}. \quad (3)$$

Intuitively, we need to maximize the likelihood of the alternative hypothesis that suggests the layer-wise importance scores in $\mathcal{S}$ and $\bar{\mathcal{S}}$ are different, where the former is denoted as $\mathcal{N}_{\mathcal{S}} = \{a_i \mid m_i \in \mathcal{S}\}$ and the latter is denoted as $\mathcal{N}_{\bar{\mathcal{S}}} = \{a_i \mid m_i \in \bar{\mathcal{S}}\}$. This can be solved by minimizing the variance of the importance scores in $\mathcal{N}_{\mathcal{S}}$ and $\mathcal{N}_{\bar{\mathcal{S}}}$ (Xie et al., 2021). Hence we choose an optimal threshold γ^* that leads to the minimum sum of $\text{Var}(\mathcal{N}_{\mathcal{S}})$ and $\text{Var}\,(\mathcal{N}_{\bar{\mathcal{S}}})$ such that

$$\text{Var}(\mathcal{N}_{\mathcal{S}}) + \text{Var}\,(\mathcal{N}_{\bar{\mathcal{S}}}) <= \text{Var}(\mathcal{N}). \quad (4)$$

Algorithm 1 summarizes the layer-splitting process, where the input is the full-layer importance score $\mathcal{N}$ and the output is the optimized threshold $\gamma*$, which can be used to split $\mathcal{M}$ into $\mathcal{S}$ and $\bar{\mathcal{S}}$ using Equations 1 and 2. The algorithm is efficient, as it has a time complexity of $O(l \log l)$ with respect to LLM $\mathcal{M}$, which has l layers, and is independent of the size of fine-tuning data. If we run the algorithm multiple times, we can further split the important layers $\mathcal{S}$ into more important and less important subsets. For example, a hierarchical splitting of layer-wise importance score $\mathcal{N} = \{a_1, a_2, a_3, a_4, a_5\}$ is visualized in Figure 3 (b), which yields multiple subsets, whose importance are ranked $\{a_1, a_3\} > \{a_2\} > \{a_4, a_5\}$ if the maximum splitting number is two. Hence, we have a fine-grained set of importance layers $\mathcal{S}^* = \{m_1, m_3\}$ used for fine-tuning if we select the highest importance score set $\{a_1, a_3\}$. While more hierarchical splits yield more fine-grained important layers, there is a trade-off between model performance and fine-tuning efficiency, as too many splits could cause down-sampling important layers, as described in Figure 2.

Algorithm 1: Layer splitting algorithm

Output: optimized threshold γ^*
Input: importance score array $\mathcal{N}$
Initialization: rank $\mathcal{N}$ in descending order;
set array $\mathcal{V} = \emptyset$, l the length of $\mathcal{N}$, $j = 0$
while j *is less than* l **do**
  $\mathcal{N}_{\mathcal{S}} \leftarrow \mathcal{N}[:j]$
  $\mathcal{N}_{\bar{\mathcal{S}}} \leftarrow \mathcal{N}[j:]$
  $\mathcal{V}_j \leftarrow \text{Var}(\mathcal{N}_{\mathcal{S}}) + \text{Var}(\mathcal{N}_{\bar{\mathcal{S}}})$
  $j \leftarrow j + 1$
end
$\gamma^* = \mathcal{N}[\text{ArgMin}(\mathcal{V})]$

4.3 Layer Update

Upon selection, the layers in $\mathcal{S}$ are encapsulated with LoRA (Hu et al., 2022) for fine-tuning, while the layers in $\bar{\mathcal{S}}$ are frozen. Since the importance scores $\mathcal{N}$ and splitting threshold γ keep updating as the fine-tuning progress, the sizes of $\mathcal{S}$ and $\bar{\mathcal{S}}$ might change, where important layers could become redundant and contribute less to the fine-tuning if their gradient norms descend below the optimized threshold γ^*, and vice versa. In practice, we select important and redundant layers every k step(s).

5 Experiments

5.1 Data Preprocessing

ArgRevision corpus contains substantial revisions, which may have empty original sentences $\mathcal{R}_1$ (in adding revisions) or empty revised sentences $\mathcal{R}_2$ (in deleting revisions) as shown in Table 8 in the Appendix. Revisions in ITERATER are minor, of which $\mathcal{R}_1$ and $\mathcal{R}_2$ are not null as shown in Table 10 in the Appendix. Also, ArgRevision only allows one intention for a revision, but ITERATER allows multiple, e.g., the same $\mathcal{R}_1$ could be revised to several $\mathcal{R}_2$ with different intention labels as shown in Figure 1 and more in rows 1, 2, and 3 in Table 10 in the Appendix. Additionally, we formulate revisions into **vanilla** and **instruction** data for vanilla fine-tuning and instruction tuning, respectively, where the former is a pair of $\{\mathcal{R}_1, \mathcal{R}_2\}$, and the latter are

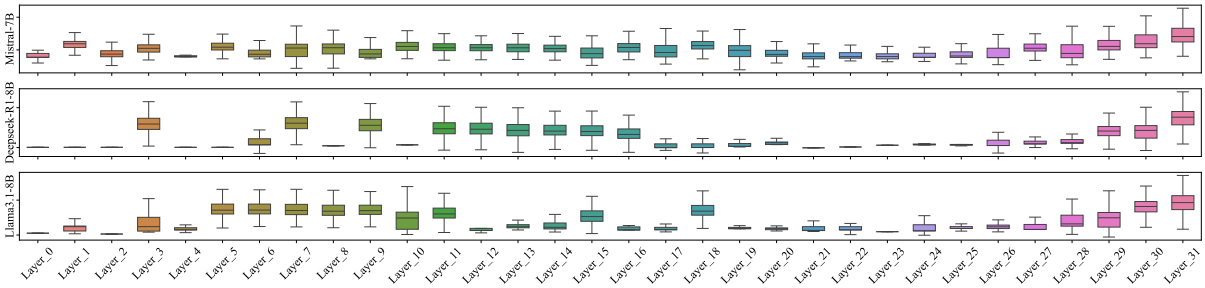

Figure 4: The importance scores (gradient norms) vary across different LLM layers on the ArgRevision corpus. The high variances indicate that the gradients have changed significantly, suggesting the layers are actively learning from the data. The layers with low variances suggest they have been frozen most of the time.

formulated with an instruction: ### *Instruction: Identify the intention of the revision between the original sentence and the revised sentence. The possible intentions include*: $\mathcal{Y}$. ### *Original Sentence*: $\mathcal{R}_1$. ### *Revised Sentence*: $\mathcal{R}_2$, where $\mathcal{Y}$ is the annotated intention labels.

5.2 Baselines and Evaluation Metrics

To evaluate the proposed method and answer our research questions, we use three LLMs, i.e., Mistral-7B, which is known for efficient inference (Jiang et al., 2023), Llama3.1-8B, which achieves good performance for general NLP tasks (Grattafiori et al., 2024), and Deepseek-R1-8B, which is a distilled version of the Llama model with emphasized reasoning capability (Guo et al., 2025). We compare our IR-Tuning to the following baselines:

- **RoBERTa**: We use a RoBERTa-large (Devlin et al., 2019) as a small language model baseline the same as Du et al. (2022b).
- **LISA-Baseline**: We fine-tune randomly selected four layers based on the algorithm in Pan et al. (2024). Here, we apply LoRA to the selected layers for PEFT based on Yao et al. (2024).
- **IST-Baseline**: We compute layer-wise importance scores based on Yao et al. (2024) and sample the top eight layers for PEFT with LoRA.
- **Full-Finetuning**: We fine-tune full LLM layers with LoRA (Hu et al., 2022) as an upper bound.

In the implementation, we build the framework with PyTorch[2], use pretrained models from Huggingface[3], and optimize multi-class cross-entropy loss with Adam optimizer on an Nvidia A100 GPU. We set the batch size as 16, the maximum text length cutoff as 256 for vanilla data and 1024 for instruction data, and the learning rate as 2e-4. We use a default one split to select important and redundant layers every fine-tuning step, and log results every 10 steps. We fine-tune LLMs for four epochs, and train the RoBERTa model for 20 epochs. In addition, we split the ArgRevision corpus into 80%, 10%, and 10% for training, validation, and test sets, respectively. We use the splits in Du et al. (2022b) for the ITERATER corpus. The detailed splits are shown in Tables 11 and 12 in the Appendix. We tune hyperparameters on the validation sets and report macro-average Precision, Recall, F1-score, and the area under the precision-recall curve (AUPRC) used for evaluating imbalanced multi-class classification, all on the test sets.

6 Results

6.1 The Importance of LLM Layers

To understand the layer-wise contributions to the revision prediction, we visualize importance score (gradient norm) changes across all the LLM layers during IR-Tuning on the ArgRevision corpus, as shown in Figure 4. For Mistral-7B, gradient updates across almost all layers have relatively high variances. This suggests that all the LLM layers are actively updating and share similar capacities for handling text revisions. Regarding Deepseek-R1-8B, several layers, such as Layers 0 to 5 except Layer 3 and Layers 21 to 25, exhibit low gradient variances, which suggests these layers are not heavily involved in the fine-tuning process. Instead, layers with relatively high gradient variances, such as Layers 3, 7, 9, and several middle and top layers (Layers 29 to 31), suggest frequent updates and active learning from the data. For Llama3.1-8B, the top and bottom layers are almost identical to those of Deepseek-R1-8B, possibly because Deepseek-R1-8B was distilled on Llama models, thus sharing similar patterns. Despite differences in layer-wise gradient norms across all the LLMs,

[2] https://pytorch.org
[3] https://huggingface.co

Models	Methods	Layer Num	ArgRevision Corpus				ITERATER Corpus			
			Precision	Recall	F1-Score	AUPRC	Precision	Recall	F1-Score	AUPRC
RoBERTa	-	-	52.00	46.35	47.95	51.47	44.99	51.19	46.31	52.69
Mistral-7B	Full-Finetuning	fixed (32)	53.60	50.09	50.54	49.08	54.95	52.48	51.56	56.81
	LISA-Baseline	fixed (4)	31.00	34.29	31.06	43.98	45.31	47.32	45.14	51.24
	IST-Baseline	fixed (8)	44.51	40.64	40.69	46.39	45.47	50.67	46.61	50.60
	IR-Tuning (ours)	dynamic	**51.45**	**46.26**	**47.20**	**49.16***	**47.61**	**52.64***	**49.38**	**54.22**
DeepSeek-R1-8B	Full-Finetuning	fixed (32)	54.13	47.94	49.33	50.78	52.04	53.03	51.62	56.00
	LISA-Baseline	fixed (4)	**54.82***	38.94	37.48	44.85	46.94	50.18	47.26	51.74
	IST-Baseline	fixed (8)	44.98	44.27	43.76	48.64	47.40	**52.11**	48.64	52.76
	IR-Tuning (ours)	dynamic	52.20	**47.19**	**48.47**	**50.14**	**52.93***	52.06	**50.66**	**54.46**
Llama3.1-8B	Full-Finetuning	fixed (32)	54.06	49.30	50.56	51.14	53.33	54.51	52.38	57.46
	LISA-Baseline	fixed (4)	35.23	35.66	33.46	42.01	49.96	51.87	48.89	54.96
	IST-Baseline	fixed (8)	44.29	42.52	42.16	47.99	**51.84**	**52.77**	**51.69**	55.81
	IR-Tuning (ours)	dynamic	**54.17***	**49.17**	**50.15**	**52.69***	48.45	51.94	49.20	**56.42**

Table 3: The performance of different LLMs and PEFT methods on the ArgRevision and ITERATER test sets. The bold numbers represent the best results, and the asterisks indicate that the results are better than full fine-tuning.

the proposed IR-Tuning can select high-gradient layers to ensure the most informative layers are effectively fine-tuned on the revision task; in contrast, less informative layers are not frequently used. Similar observations on the ITERATER corpus are shown in Figure 8 in the Appendix, which confirms **RQ1** that LLM layers contribute differently to revision intention prediction, thus fine-tuning layers that contribute more while freezing those that contribute less can potentially benefit the revision task.

6.2 The Performance of IR-Tuning

We show the intention prediction performance on the ArgRevision and ITERATER corpora in Table 3. IR-Tuning achieves the best results across almost all metrics and outperforms RoBERTa in most settings, suggesting the generalizability of IR-Tuning across different LLMs and revision corpora. Although Llama3.1-8B achieves better F1 scores using IST-Baseline instead of IR-Tuning on the ITERATER corpus, the results are reversed in terms of AUPRC. This is because unbalanced data can cause the model to poorly learn the minority classes (e.g., *Style* in Table 9 in the Appendix), which can downgrade the macro F1 scores that treat each class equally but are less sensitive to AUPRC. Thus, the higher AUPRC in IR-Tuning across all settings suggests its effectiveness compared to the baselines. Sometimes, LLMs with IR-Tuning can outperform full fine-tuning results on several metrics, which demonstrates the success of IR-Tuning over standard PEFT methods. These results answer **RQ2**, which suggests that we can dynamically select important layers for PEFT on even small corpora.

To investigate the efficiency of IR-Tuning, we plot the losses of fine-tuning Llama3.1-8B on the

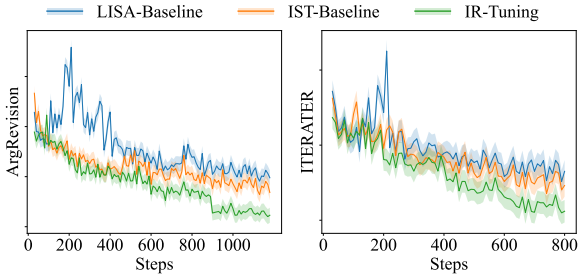

Figure 5: The Llama3.1-8B fine-tuning losses on the ArgRevision and ITERATER training sets, using IR-Tuning and baseline PEFT methods.

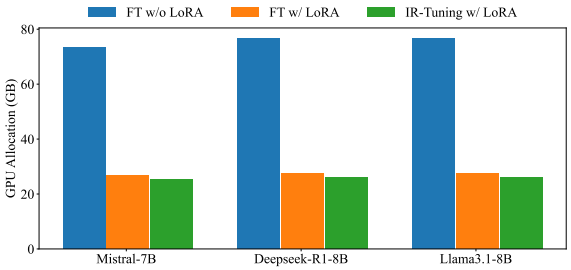

Figure 6: The GPU memory allocations for the Full-Finetuning (FT) with and without PEFT (LoRA), and IR-Tuning with PEFT (LoRA) on the ArgRevision corpus.

training sets of ArgRevision and ITERATER corpora in Figure 5. IR-Tuning exhibits fast convergence compared to other PEFT methods, of which LISA has surprisingly high fluctuations during the beginning steps, largely because it fine-tunes on random layers rather than the layers with high gradient norms. The similar patterns are observed on Mistral-7B and Deepseek-R1-8B in Figure 9 and 10 in the Appendix. These observations highlight the superiority of IR-Tuning for fast convergence. In addition, we plot its GPU memory allocations while fine-tuning on the ArgRevision corpus in Figure 6 and the ITERATER corpus in Figure 11

Models	Inputs	ArgRevision Evidence			ArgRevision Reasoning			ITERATER Corpus				
		Relevant	Irrelevant	Repeated	LCE	not LCE	Commentary	Clarity	Fluency	Coherence	Style	Meaning
RoBERTa	vanilla	78.14	42.35	13.64	66.67	25.24	61.67	66.47	81.22	30.30	11.11	54.21
Mistral-7B	vanilla	78.11	**35.96**	**16.00**	**71.52**	**42.11**	**63.72**	71.65	75.95	31.43	0.00	55.17
	instruct	**78.52**	31.46	5.00	67.52	38.33	62.39	**72.93**	**80.90**	**37.14**	0.00	**55.91**
Deepseek-R1-8B	vanilla	75.97	30.77	8.89	66.01	26.00	58.99	**74.73**	82.22	**37.93**	**16.67**	**62.79**
	instruct	**76.53**	**33.71**	**13.04**	**71.73**	**35.16**	**60.66**	72.04	**82.87**	33.33	10.53	54.55
Llama3.1-8B	vanilla	76.51	30.23	4.44	69.39	33.65	67.15	72.38	**87.64**	**31.88**	**19.05**	55.56
	instruct	**79.69**	**34.57**	**10.26**	**71.57**	**35.64**	**69.17**	**75.33**	83.62	30.51	0.00	**56.52**

Table 4: The F1 scores of IR-Tuning with vanilla inputs and instruction inputs on the ArgRevision (Evidence and Reasoning) and ITERATER corpora. The bold numbers are the best results in each setting.

in the Appendix. For a fair comparison, we set the batch size to one for both full fine-tuning and PEFT. The histogram shows that IR-Tuning with LoRA uses the smallest memory across different LLMs, compared to full-layer fine-tuning with and without LoRA, which confirms its memory efficiency.

6.3 The Effectiveness of Instruction Tuning

We evaluate IR-Tuning performance on specific intentions with and without instruction tuning for each corpus. For ArgRevision, Table 4 shows IR-Tuning is worse than RoBERTa in predicting *Irrelevant* and mostly worse for *Relevant* and *Repeated* intentions, all from evidence revision. This might be because LLMs struggle to learn evidence revision with limited data; in contrast, LLMs perform well in predicting reasoning revision. Although Llama3.1-8B is generally better than Mistral-7B in several NLP tasks, it does not hold true on argument revisions using vanilla input data, which might be because Llama3.1-8B does not learn revisions well without proper instructions. However, IR-Tuning can improve Mistral-7B without using instruction tuning. For ITERATER, LLMs are generally better than the RoBERTa baseline except for *Style*, which has limited annotations (e.g., 128 *Style* labels in Table 9 in the Appendix). While instruction tuning is generally helpful on the ArgRevision corpus, except for Mistral-7B, it sometimes downgrades the performance on the ITERATER corpus for Deepseek-R1-8B and Llama3.1-8B. This might be because ArgRevision has substantial revisions that add and delete entire sentences (e.g., rows #1 and #2 in Table 8 in the Appendix), thus IR-Tuning needs contextualized instructions to help predict intentions. In contrast, the revisions in ITERATER are minor (see Table 10 in the Appendix), which rely less on the instructions to learn intentions. These observations suggest that contextualized instructions are not always helpful for LLMs to learn revisions, which answers **RQ3**.

Models	Sizes	ArgRevision		ITERATER	
		F1-Score	AUPRC	F1-Score	AUPRC
Phi-2	2.7B	39.89	44.88	42.77	47.22
Mistral	7B	47.20	49.16	49.38	54.22
Deepseek-R1	8B	48.47	50.14	50.66	54.46
Llama3.1	8B	50.15	52.69	49.20	56.42
Llama2	13B	**52.49**	**53.11**	**52.19**	**57.45**

Table 5: The performance of IR-Tuning with LoRA PEFT for different sizes of LLMs across two corpora.

Adapters	ArgRevision		ITERATER	
	F1-Score	AUPRC	F1-Score	AUPRC
Botteneck	50.21	**54.38**	47.97	56.00
LoRA	50.15	52.69	49.20	56.42
DoRA	**52.66**	51.47	**50.94**	**58.64**

Table 6: The performance of IR-Tuning on Llama3.1-8B with different PEFT adapters across two corpora.

6.4 Generalizability and Robustness

To validate the generalizability of IR-Tuning, we evaluate its performance using LLMs with varying parameter sizes, ranging from small to large. Table 5 shows that F1 scores and AUPRC generally improve as the size of the model increases, which implies that larger LLMs might work better since larger models can capture complex revision patterns and leverage richer contextual information. In addition, we implement IR-Tuning using different PEFT adapters, i.e., classic Bottleneck (Houlsby et al., 2019) and advanced DoRA (Liu et al., 2024). Table 6 shows that DoRA generally performs better than LoRA on two corpora, except for AUPRC in ArgRevision. This suggests that IR-Tuning benefits from DoRA's ability to decompose the weight magnitude and direction within LLM layers (Liu et al., 2024).

Furthermore, we compare our gradient-based importance scoring metric with the magnitude-based method regarding parameter weights (Ma et al., 2023) and the similarity-based method that measures the hidden state before and after LLM lay-

Model	ArgRevision		ITERATER	
	F1-Score	AUPRC	F1-Score	AUPRC
Similarity	**52.36**	**53.53**	49.23	53.24
Magnitude	48.42	48.72	**52.57**	53.26
Gradient	50.15	52.69	49.20	**56.42**

Table 7: The performance of IR-Tuning on Llama3.1-8B with LoRA PEFT and different importance scoring metrics across two corpora.

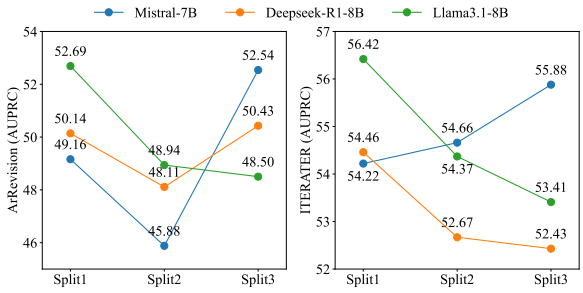

Figure 7: The performance (AUPRC) of IR-Tuning with LoRA PEFT and different splits across two corpora.

ers (Chen et al., 2025), as shown in Table 7. Although the similarity metric appears to be advantageous on ArgRevision, it is more computationally expensive than magnitude and gradient-based metrics, as it necessitates additional computations right before and after each LLM layer. In addition, the gradient-based metric has a higher AUPRC on the ITERATER data, suggesting it's more reliable for unbalanced data, while the magnitude-based metric has generally accurate predictions on ITERATER. These observations suggest that performance varies across corpora; however, gradient-based methods are informative as they measure whether LLM layers are actively engaged during fine-tuning.

Moreover, we study the parameter sensitivity of the layer-selection algorithm in Figure 7. In the ArgRevision corpus, the performance of Llama3.1-8B decreases as adding more splits, which could be due to down-sampling issues, as shown in Figure 2. Mistral-7B and Deepseek-R1-8B downgrade performance in two splits but improve in three. In the ITERATER corpus, performance typically drops when adding more splits, except for Mistral-7B. These observations suggest that the layer selection depends on both data and models, where one split is generally good for Llama-based models, and three splits work best for Mistral-based models.

7 Conclusion

We present a layer-wise PEFT framework named IR-Tuning, which dynamically selects a subset of LLM layers for fine-tuning while freezing the others. We demonstrate that IR-Tuning is effective across multiple revision corpora and LLMs, and achieves fast fine-tuning convergence and consumes low GPU memory. Although IR-Tuning is primarily used for text revision in this study, it can potentially be used for other NLP tasks. In future work, we will evaluate its performance on more revision tasks to generalize our findings.

Ethics Statement

We use our previously developed automated writing evaluation system (Liu et al., 2025) to collect the ArgRevision corpus under standard protocols approved by an Institutional Review Board (IRB). Expert annotators, familiar with both the system and the writing tasks, were employed to label revision intentions. The data collection and annotation processes do not raise any ethical concerns. Furthermore, our research on text revision, particularly for argument essay revisions, is critically needed for automated student writing evaluation, which benefits both the education and NLP communities.

Limitations

Despite the effectiveness of our proposed method, several limitations are noted. First, the integration of IR-Tuning relies on the gradient norm on every LLM layer to hypothetically measure the importance of the layers. However, more data-driven and task-specific metrics, e.g., evaluation loss, could be leveraged to score layer importance. Nevertheless, IR-Tuning does not introduce significant computational overhead compared to standard PEFT, based on our pilot studies. We will further investigate its efficiency in future work. Also, we empirically use one split for selecting LLM layers in our current settings. We will explore automated methods for splitting layers in future work. Second, computational limitation constrains our evaluation to relatively lightweight models. While larger LLMs, e.g., Llama3.3-70B, offer enhanced capabilities for contextual understanding, we are unable to incorporate them in this study. Our evaluations are based on small annotated corpora, which have limited generalizability to cover diverse revisions in real-world scenarios. Third, we do not investigate other PEFT techniques, e.g., prompt-tuning. Thus, a deeper exploration of different PEFT methods, instruction designs, and diverse LLMs may achieve more robust task-specific fine-tuning strategies.

Acknowledgments

This research was supported by National Science Foundation Award #2202347 and its REU and NAIRR Pilot Demonstration Project supplements, as well as through CloudBank allocations. The opinions expressed are those of the authors and do not represent the views of the institutes. The authors would like to thank the anonymous reviewers and the Pitt PETAL groups for their valuable feedback on this work.

References

Tazin Afrin and Diane Litman. 2018. Annotation and classification of sentence-level revision improvement. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 240–246, New Orleans, Louisiana.

Tazin Afrin and Diane Litman. 2023. Predicting desirable revisions of evidence and reasoning in argumentative writing. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2550–2561, Dubrovnik, Croatia. Association for Computational Linguistics.

Tazin Afrin, Elaine Lin Wang, Diane Litman, Lindsay Clare Matsumura, and Richard Correnti. 2020. Annotation and classification of evidence and reasoning revisions in argumentative writing. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, Seattle, Washington, USA (Remote).

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A resource and analyses on edits in instructional texts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5721–5729, Marseille, France. European Language Resources Association.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. 2025. Streamlining redundant layers to compress large language models. *The Fourteenth International Conference on Learning Representations*.

Ruining Chong, Cunliang Kong, Liu Wu, Zhenghao Liu, Ziye Jin, Liner Yang, Yange Fan, Hanghang Fan, and Erhong Yang. 2023. Leveraging prefix transfer for multi-intent text revision. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1219–1228, Toronto, Canada. Association for Computational Linguistics.

Richard Correnti, Lindsay Clare Matsumura, Laura Hamilton, and Elaine Wang. 2013. Assessing students' skills at writing analytically in response to texts. *The Elementary School Journal*, 114(2):142–177.

Rip Correnti, Elaine Lin Wang, Lindsay Clare Matsumura, Diane Litman, Zhexiong Liu, and Tianwen Li. 2024. Supporting students' text-based evidence use via formative automated writing and revision assessment. In *The Routledge international handbook of automated essay evaluation*, pages 221–243. Routledge.

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. 2024. ARIES: A corpus of scientific paper edits made in response to peer reviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6985–7001, Bangkok, Thailand. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wanyu Du, Zae Myung Kim, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022a. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 96–108, Dublin, Ireland. Association for Computational Linguistics.

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022b. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590, Dublin, Ireland. Association for Computational Linguistics.

Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, and Mehdi Rezagholizadeh. 2025. *KronA: Parameter-Efficient Tuning with Kronecker Adapter*, pages 49–65. Springer Nature Switzerland, Cham.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. 2024. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642, Bangkok, Thailand. Association for Computational Linguistics.

Jill Fitzgerald. 1987. Research on revision in writing. *Review of educational research*, 57(4):481–506.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

John R Hayes and Linda S Flower. 1986. Writing research and the writer. *American psychologist*, 41(10):1106.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276, Singapore. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arXivEdits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9420–9435, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Léane Jourdan, Florian Boudin, Nicolas Hernandez, and Richard Dufour. 2024. CASIMIR: A corpus of scientific articles enhanced with multiple author-integrated revisions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2883–2892, Torino, Italia. ELRA and ICCL.

Gal Kaplun, Andrey Gurevich, Tal Swisa, Mazor David, Shai Shalev-Shwartz, and Eran Malach. 2023. Less is more: Selective layer finetuning with subtuning. *arXiv preprint arXiv:2302.06354*.

Omid Kashefi, Tazin Afrin, Meghan Dale, Christopher Olshefski, Amanda Godley, Diane Litman, and Rebecca Hwa. 2022. Argrewrite v. 2: an annotated argumentative revisions corpus. *Language Resources and Evaluation*, pages 1–35.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representation*.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. 2023. Surgical fine-tuning improves adaptation to distribution shifts. *International Conference on Learning Representations*.

Tao Lei, Junwen Bai, Siddhartha Brahma, Joshua Ainslie, Kenton Lee, Yanqi Zhou, Nan Du, Vincent Zhao, Yuexin Wu, Bo Li, and 1 others. 2024. Conditional adapters: Parameter-efficient transfer learning with fast inference. *Advances in Neural Information Processing Systems*, 36.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Empirical Methods in Natural Language Processing*.

Nianqi Li, Jingping Liu, Sihang Jiang, Haiyun Jiang, Yanghua Xiao, Jiaqing Liang, Zujie Liang, Feng Wei, Jinglei Chen, Zhenghong Hao, and 1 others. 2024a. Cr-llm: A dataset and optimization for concept reasoning of large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13737–13747.

Tianwen Li, Zhexiong Liu, Lindsay Matsumura, Elaine Wang, Diane Litman, and Richard Correnti. 2024b. Using large language models to assess young students' writing revisions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 365–380, Mexico City, Mexico. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Lei Liu and Min Zhu. 2022. Bertalign: Improved word embedding-based sentence alignment for chinese–english parallel corpora of literary texts. *Digital Scholarship in the Humanities*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association of Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Zhexiong Liu, Diane Litman, Elaine Wang, Lindsay Matsumura, and Richard Correnti. 2023a. Predicting the quality of revisions in argumentative writing. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 275–287, Toronto, Canada. Association for Computational Linguistics.

Zhexiong Liu, Diane Litman, Elaine L Wang, Tianwen Li, Mason Gobat, Lindsay Clare Matsumura, and Richard Correnti. 2025. eRevise+RF: A writing evaluation system for assessing student essay revisions and providing formative feedback. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 173–190, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhexiong Liu, Licheng Liu, Yiqun Xie, Zhenong Jin, and Xiaowei Jia. 2023b. Task-adaptive meta-learning framework for advancing spatial generalizability. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14365–14373.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: on the structural pruning of large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2024. Towards automated document revision: Grammatical error correction, fluency edits, and beyond. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. Lisa: layer-wise importance sampling for memory-efficient large language model fine-tuning. *Advances in Neural Information Processing Systems*, 37:57018–57049.

Yixing Peng, Quan Wang, Licheng Zhang, Yi Liu, and Zhendong Mao. 2024. Chain-of-question: A progressive question decomposition approach for complex knowledge base question answering. In *ACL (Findings)*.

Qian Ruan, Ilia Kuznetsov, and Iryna Gurevych. 2024a. Are large language models good classifiers? a study on edit intent classification in scientific document revisions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15049–15067, Miami, Florida, USA. Association for Computational Linguistics.

Qian Ruan, Ilia Kuznetsov, and Iryna Gurevych. 2024b. Re3: A holistic framework and dataset for modeling collaborative document revision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4635–4655, Bangkok, Thailand. Association for Computational Linguistics.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429.

Antonette Shibani, Simon Knight, and Simon Buckingham Shum. 2018. Understanding revisions in student writing through revision graphs. In *International Conference on Artificial Intelligence in Education*, pages 332–336, Cham. Springer International Publishing.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen, and Lei Meng. 2024. Rewritelm: An instruction-tuned large language model for text rewriting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18970–18980.

Gabriella Skitalinskaya and Henning Wachsmuth. 2023. To revise or not to revise: Learning to detect improvable claims for argumentative writing support. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15799–15816, Toronto, Canada. Association for Computational Linguistics.

Nancy Sommers. 1980. Revision strategies of student writers and experienced adult writers. *College Composition & Communication*, 31(4):378–388.

Alexander Spangher, Xiang Ren, Jonathan May, and Nanyun Peng. 2022. NewsEdits: A news article revision dataset and a novel document-level reasoning challenge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 127–157, Seattle, United States. Association for Computational Linguistics.

Sotaro Takeshita, Tommaso Green, Ines Reinig, Kai Eckert, and Simone Ponzetto. 2024. ACLSum: A new dataset for aspect-based summarization of scientific publications. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6660–6675, Mexico City, Mexico. Association for Computational Linguistics.

Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. Adamix: Mixture-of-adapter for parameter-efficient tuning of large language models. *arXiv preprint arXiv:2205.12410*, 1(2):4.

Chenxing Wei, Yao Shu, Ying Tiffany He, and Fei Yu. 2025. Flexora: Flexible low-rank adaptation for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14643–14682, Vienna, Austria. Association for Computational Linguistics.

Yiqun Xie, Erhu He, Xiaowei Jia, Han Bao, Xun Zhou, Rahul Ghosh, and Praveen Ravirathinam. 2021. A statistically-guided deep network transformation and moderation framework for data with spatial heterogeneity. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 767–776. IEEE.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in Wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010, Copenhagen, Denmark. Association for Computational Linguistics.

Kai Yao, Penglei Gao, Lichun Li, Yuan Zhao, Xiaofeng Wang, Wei Wang, and Jianke Zhu. 2024. Layer-wise importance matters: Less memory for better performance in parameter-efficient fine-tuning of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1977–1992, Miami, Florida, USA. Association for Computational Linguistics.

Fan Zhang, Homa B. Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578, Vancouver, Canada. Association for Computational Linguistics.

Fan Zhang, Rebecca Hwa, Diane Litman, and Homa B. Hashemi. 2016. ArgRewrite: A web-based revision assistant for argumentative writings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 37–41, San Diego, California. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In *Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143, Denver, Colorado. Association for Computational Linguistics.

Kaiyan Zhang, Ning Ding, Biqing Qi, Xuekai Zhu, Xinwei Long, and Bowen Zhou. 2023. Crash: Clustering, removing, and sharing enhance fine-tuning without full large language model. In *Empirical Methods in Natural Language Processing*.

Zhi Zhang, Qizhe Zhang, Zijun Gao, Renrui Zhang, Ekaterina Shutova, Shiji Zhou, and Shanghang Zhang. 2024. Gradient-based parameter selection for efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28566–28577.

Zheng Zhao, Yftah Ziser, and Shay B Cohen. 2024. Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15195–15214, Miami, Florida, USA. Association for Computational Linguistics.

Hongyun Zhou, Xiangyu Lu, Wang Xu, Conghui Zhu, Tiejun Zhao, and Muyun Yang. 2025. LoRA-drop: Efficient LoRA parameter pruning based on output evaluation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5530–5543, Abu Dhabi, UAE. Association for Computational Linguistics.

Ligeng Zhu, Lanxiang Hu, Ji Lin, and Song Han. 2024. Lift: Efficient layer-wise fine-tuning for large model models.

A Appendix

ID	Sentence in Original Essay	Sentence in Revised Essay	Revision Behavior	Revision Type	Revision Intention
1	Sauri has been struggling with poverty.		Delete	Evidence	Irrelevant
2		The author did convince me that,"winning the fight against poverty is achievable in our life time"the author tells us that this is achievable in paragraph 3.in paragraph 3 it says," we are halfway to 2035, and the world is capable of meeting these goals.	Add	Evidence	Relevant
3	...	...	...	...	...
4		I can infer they want to get Sauri out of poverty before 2035.	Add	Reasoning	Other
5	According to paragraph 3 it says,"the plan is to get people out of poverty, assure them access to health care and help them stabilize the economy and quality of life in their communities."	According to paragraph 3 it says,"the plan is to get people out of poverty, assure them access to health care and help them stabilize the economy and quality of life in their communities."	N/A	N/A	N/A
6	This shows the Sauri needs as much help as they can to get out of poverty.	This shows the Sauri needs as much help as they can to get out of poverty.	N/A	N/A	N/A
7		Also it shows that people are willing to help Sauri.	Add	Reasoning	LCE
8	...	...	...	...	...
9	In 2035 I can infer that they will be financially stable and will have better housing.	In 2035 I can infer that they will be financially stable and will have better housing.	N/A	N/A	N/A
10	...	...	...	...	...
11		Local leaders take it from there."	Add	Evidence	Relevant
12		This shows how they provided Sauri with the necessities and they can see action taking place and see improvements.	Add	Reasoning	LCE

Table 8: The examples of revision intention annotations for an essay in the ArgRevision corpus.

	Clarity	Fluency	Coherence	Style	Meaning	Others	Total
Add	94	208	37	2	342	6	689
Delete	282	111	180	20	11	11	615
Modify	1,225	623	176	106	543	41	2,714
Total	1,601	942	393	128	896	58	4,018

Table 9: The statistics of revisions in the ITERATER corpus.

ID	Sentence in Original Article	Sentence in Revised Article	Revision Intention
1	In this work, we point out the inability to infer behavioral conclusions from probing results, and offer an alternative method which is focused on how the information is being used, rather than on what information is encoded.	In this work, we point out the inability to infer behavioral conclusions from probing results, and offer an alternative method which focuses on how the information is being used, rather than on what information is encoded.	Style
2	In this work, we point out the inability to infer behavioral conclusions from probing results , and offer an alternative method which focuses on how the information is being used, rather than on what information is encoded.	In this work, we point out the inability to infer behavioral conclusions from probing results and offer an alternative method which focuses on how the information is being used, rather than on what information is encoded.	Fluency
3	In this work, we point out the inability to infer behavioral conclusions from probing results , and offer an alternative method which focuses on how the information is being used, rather than on what information is encoded.	In this work, we point out the inability to infer behavioral conclusions from probing results , and offer an alternative method that focuses on how the information is being used, rather than on what information is encoded.	Clarity
4	A growing body of work makes use of probing in order to investigate the working of neural models, often considered black boxes.	A growing body of work makes use of probing to investigate the working of neural models, often considered black boxes.	Coherence
5	We explore representations from different model families (BERT, RoBERTa, GPT-2 , etc.) and find evidence for emergence of linguistic manifold across layer depth (e.g., manifolds for part-of-speech and combinatory categorical grammar tags). We further observe that different encoding schemes used to obtain the representations lead to differences in whether these linguistic manifolds emerge in earlier or later layers of the network .	We explore representations from different model families (BERT, RoBERTa, GPT-2 , etc.) and find evidence for emergence of linguistic manifold across layer depth (e.g., manifolds for part-of-speech tags), especially in ambiguous data (i.e, words with multiple part-of-speech tags, or part-of-speech classes including many words) .	Meaning-changed

Table 10: The examples of revision intention annotations in the ITERATER corpus.

	Evidence Revision			Reasoning Revision		
	Relevant	Irrelevant	Repeated	LCE	not LCE	Commentary
Train	2,029	425	216	1,200	358	506
Val	240	50	29	148	46	79
Test	240	50	31	144	56	71
Total	2,509	525	276	1,492	460	656

Table 11: The statistics of revision intentions in the training, validation, and test sets of the ArgRevision corpus.

	Clarity	Fluency	Coherence	Style	Meaning
Train	1,258	739	311	100	807
Val	157	115	46	13	54
Test	186	88	36	15	35
Total	1,601	942	393	128	896

Table 12: The statistics of revision intentions in the training, validation, and test sets of the ITERATER corpus.

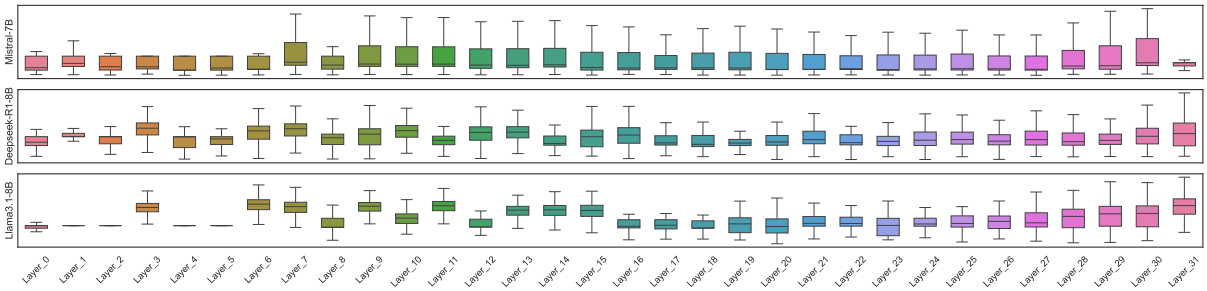

Figure 8: The importance scores (gradient norms) vary across different LLM layers on the ITERATER corpus. The high variances indicate that the gradients have changed significantly, suggesting the layers are actively learning from the data. The layers with low variances mean they have been frozen most of the time.

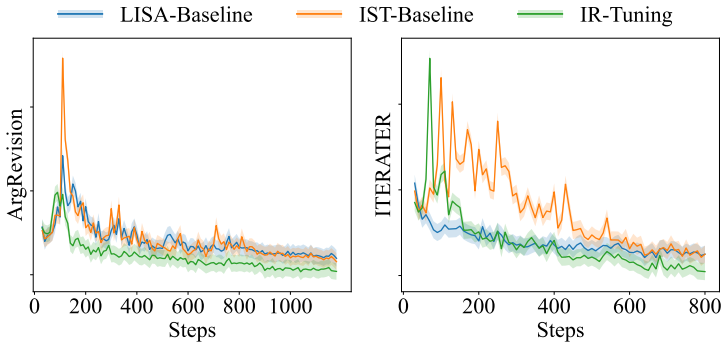

Figure 9: The Mistral-7B fine-tuning losses on the ArgRevision and ITERATER training sets, using IR-Tuning and baseline PEFT methods.

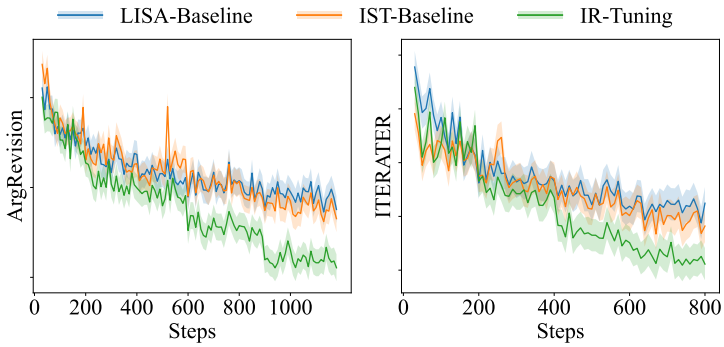

Figure 10: The Deepseek-R1-8B fine-tuning losses on the ArgRevision and ITERATER training sets, using IR-Tuning and baseline PEFT methods.

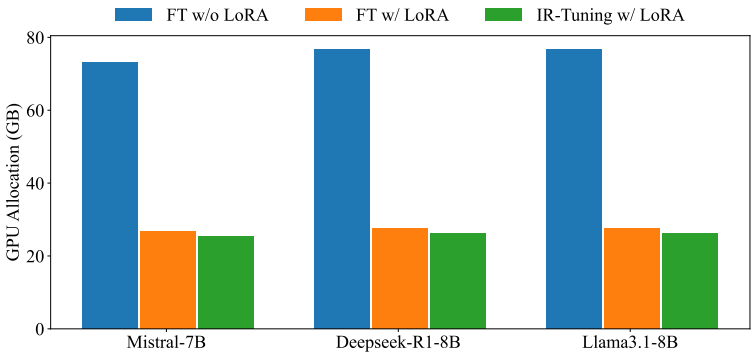

Figure 11: The GPU memory allocations for the Full-Finetuning (FT) with and without PEFT (LoRA), and IR-Tuning with PEFT (LoRA) on the ITERATER corpus.