Beyond Similarity: A Gradient-based Graph Method for Instruction Tuning Data Selection

Yang Zhao[1], **Li Du**[2], **Xiao Ding**[1†], **Yangou Ouyang**[1], **Hepeng Wang**[1],
Kai Xiong[1], **Jinglong Gao**[1], **Zhouhao Sun**[1], **Dongliang Xu**[3],
Qing Yang[3], **Dongchen Li**[3], **Bing Qin**[1], **Ting Liu**[1]
[1]Research Center for Social Computing and Interactive Robotics
Harbin Institute of Technology, China
[2]Beijing Academy of Artificial Intelligence, China
[3]Du Xiaoman Technology (Beijing) Co., Ltd., China
{yangzhao, xding, yooy, hpwang, kxiong}@ir.hit.edu.cn

Abstract

Large language models (LLMs) have shown great potential across various industries due to their remarkable ability to generalize through instruction tuning. However, the limited availability of domain-specific data significantly hampers their performance on specialized tasks. While existing methods primarily focus on selecting training data from general datasets that are similar to the target domain, they often fail to consider the joint distribution of instructions, resulting in inefficient learning and suboptimal knowledge transfer. To address these challenges, we introduce **G2IS** (**G**radient-based **G**raph **I**nstruction **S**election), a novel method that constructs a mixed gradient-based instruction graph to capture the joint distribution and interdependencies between instructions. By accounting for the relationships between instructions, G2IS improves domain adaptation efficiency. Additionally, we propose a gradient walk algorithm to refine the data selection process, enhancing both training effectiveness and efficiency. Our experiments demonstrate that G2IS outperforms traditional methods across various domain adaptation tasks, yielding significant performance gains, particularly in complex, data-scarce scenarios. These results underscore the potential of G2IS in advancing the development of large, domain-specific models. The source code for this work is publicly available at: https://github.com/zy125413/G2IS.

1 Introduction

The increasing demand for personalized and domain-specific applications has driven the rapid advancement of domain-specific LLMs (Wu et al., 2023; Zhang and Yang, 2023). Unlike general-purpose models, these models should not only develop general expertise but also continuously adapt to evolving domain knowledge. However, the effectiveness of these models critically depends on their ability to efficiently acquire and apply relevant, domain-specific knowledge.

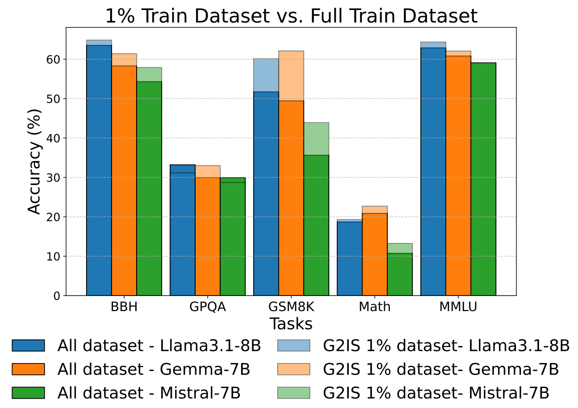

Figure 1: On the FLAN-V2 dataset, our method selects 1% of the data and compares it with the full dataset across three models. In most tasks, our approach using only 1% outperforms instruction tuning on the full dataset.

Instruction tuning has emerged as a crucial method for adapting LLMs to specialized domains through the use of curated, task-specific instruction datasets (Peng et al., 2023; Zhang et al., 2023). By exposing models to domain-relevant instructions, this approach significantly improves their ability to generalize across various tasks, including those requiring domain adaptation and complex reasoning. However, a key challenge in domain-specific instruction tuning is the limited availability of high-quality annotated data. Additionally, concerns such as user privacy have further exacerbated the data limitations in these domain-specific areas. To address this challenge, a promising solution is data selection, which involves identifying, from large-scale, general instruction datasets, the most relevant and impactful training samples that closely align with the target task (Zhao et al., 2024b). This process can substantially enhance the model's performance within the target domain. Therefore, the

[†]Corresponding Author.

effectiveness of the data selection method is critical, as it directly influences the success of domain-specific instruction tuning.

A common approach to data selection involves choosing samples that closely resemble the validation set (Xia et al., 2024; Joaquin et al., 2024). However, training data typically exhibits complex interdependencies (Yang et al., 2024; Zhao et al., 2024a), which previous methods overlook by treating relationships between data points independently. As a result, similarity-based selection methods fail to achieve optimal performance, limiting the effectiveness of instruction tuning (Hübotter et al., 2024) . This failure to account for interdependencies hinders the construction of an optimal dataset, restricting the model's ability to generalize. Fundamentally, data selection aims to identify the data from the training set that possesses the capabilities required to complete the target task. Instruction data distributions form joint distributions, where relationships between instructions should be considered (Zhao et al., 2024a). However,earlier methods (Xia et al., 2024; Joaquin et al., 2024) failed to account for this joint distribution, leading to suboptimal data selection.

To address these challenges, we propose G2IS, an innovative approach leveraging gradient-based knowledge representation for more efficient data selection. G2IS employs a mixed instruction gradient-based graph that models the complex relationships between instructions. This graph captures the joint distribution of instruction data, enabling more effective data selection. Model gradients capture the informational content of training samples (Park et al., 2023; Jain et al., 2024; Zhao et al., 2024b), influencing parameter updates and implicitly encoding how each sample contributes knowledge to the model (Hammoudeh and Lowd, 2024). Building on this, we enhance the robustness of the validation set by applying Principal Component Analysis (PCA) (Kurita, 2021) to the gradients, extracting core knowledge representations that guide data selection. For the training set, we introduce a gradient walk algorithm that refines sample selection by leveraging these gradients. This approach considers the joint distribution of instruction data, progressively selecting samples that align with the core knowledge identified in the validation set. By structuring the data selection process this way, we ensure that the selected training data is efficient and aligned with key knowledge, improving the overall quality of the training process.

We validate G2IS across multiple domain benchmarks. As shown in Figure 1, our method achieves exceptional performance with only 1% of the training data, outperforming full-data instruction tuning on most tasks. Notably, on the GSM8K dataset using the Gemma-7B model, our approach improves **accuracy by 12.66%** compared to full instruction tuning. These results highlight the efficiency of G2IS in reducing data requirements while maintaining or even surpassing baseline performance.

2 Gradient-based Knowledge Representation

During instruction tuning, LLMs update their parameters through gradient-based optimization, making gradients a natural representation of model knowledge. Gradients reflect the influence of individual training samples on parameter updates, revealing which data points contribute most to model learning. In this section, we explain how gradients are computed for the training and validation sets and how they are used to construct a **gradient-based graph** for efficient data selection. In the next section, we introduce how to utilize the Gradient-based Graph for Instruction Selection.

2.1 Gradient-based Knowledge Representation

Gradients not only drive parameter updates during instruction tuning but also implicitly encode knowledge about how training data influences model learning (Choe et al., 2024). By capturing the directional influence of each sample on parameter updates, gradients naturally reveal the relationships between instructions. Unlike static embeddings or similarity-based methods, gradients provide a dynamic, task-sensitive representation of knowledge (Hammoudeh and Lowd, 2024), effectively capturing both similarities and deeper interdependencies within the training data.

This can be understood through a first-order Taylor expansion of the loss function, where the model parameters update as:

$$\theta' = \theta - \eta \nabla \mathcal{L}(z, \theta), \tag{1}$$

where η represents the learning rate, and the gradient $\nabla \mathcal{L}(z, \theta)$ determines how each sample modifies the model, encapsulating its contribution to learning. Thus, the relationships between gradients reflect the dependencies between instruction data, with the similarity of instruction gradients enabling joint modeling and revealing the complex relationships within instruction tuning data.

2.2 Gradient Computation on Training and Validation Sets

To accurately capture the knowledge representation of different data samples, we compute gradients separately for both the training and validation sets.

For the training set, we define the **momentum-adjusted gradient** $\nabla\Gamma(z, \theta_t)$, which represents the effective gradient used for parameter updates in modern LLMs trained with the Adam optimizer (Kingma, 2014). Unlike standard gradients, it incorporates both first-order and second-order momentum terms, providing a more accurate representation of how training samples contribute to model optimization. Directly computing raw gradients without considering momentum effects can lead to imprecise knowledge representations. To address this, we use the **warmup method** proposed by Xia et al. (2024) and Liu et al. (2024), where the model is pre-trained on a small subset of instructions to initialize the optimizer's momentum states. This ensures that $\nabla\Gamma(z, \theta_t)$ reflects the actual optimization dynamics, improving data selection precision and instruction representation.

For the validation set, we compute the first-order gradient of the loss function with respect to model parameters for each sample. This gradient directly measures how each sample influences parameter updates. To avoid momentum interference, prior research (Xia et al., 2024) showed that using stochastic gradient descent (SGD) for validation gradients and Adam for training gradients improves data selection accuracy. Therefore, we compute these gradients using standard SGD without modifications.

Given the high computational cost of calculating full gradients for all model parameters, we leverage LoRA (Hu et al., 2021) to compute gradients within LoRA layers, significantly reducing memory overhead while preserving key gradient information. Additionally, we apply **Random Projection** (Johnson, 1984; Park et al., 2023) dimensionality reduction techniques to efficiently extract low-dimensional gradient features, ensuring computational efficiency without sacrificing essential knowledge representation.

2.3 Gradient-based Graph Construction

After obtaining gradient representations, we construct a structured **gradient-based graph**, consisting of nodes N_z and edges R_{ij}. Each node N_z in the graph is defined as:

$$N_z = \nabla\Gamma(z, \theta_t), \tag{2}$$

where $\nabla\Gamma(z, \theta_t)$ encapsulates the sample's contribution to model updates, effectively encoding its role in the instruction tuning process. Edges R_{ij} between nodes z_i and z_j are weighted by their cosine similarity:

$$R_{ij} = \cos\left(\nabla\Gamma(z_i, \theta_t), \nabla\Gamma(z_j, \theta_t)\right), \tag{3}$$

where higher values indicate stronger alignment in their learning impact, and negative values suggest potential conflicts. This structure captures not only direct similarities but also complementary and conflicting relationships between samples. By modeling the training data as a gradient-based graph, we capture the interdependencies overlooked by traditional similarity-based methods, laying the foundation for further capturing joint training samples that meet specific task requirements.

3 Gradient-based Graph for Instruction Selection

As illustrated in Figure 2, our method begins with a gradient-based graph. First, we apply PCA to the validation set to reduce noise and extract core knowledge. Using this core knowledge, we identify 'walk anchors'. Finally, a walk algorithm utilizes these anchors to select the training data most relevant to the validation set.

3.1 Core Knowledge Extraction from the Validation Set and Selection of Walk Anchors

The validation set, which is typically small and designed to resemble the test set, is often assumed to be independent and identically distributed in previous studies (Xia et al., 2024; Joaquin et al., 2024), where the average gradient is used as a proxy for the core knowledge it encapsulates. However, the main goal of the validation set is not only to represent the overall knowledge but also to capture the essential capabilities required to solve specific tasks. To more accurately extract this core knowledge, we employ PCA on the gradient distribution of the validation set. PCA identifies the principal components of the gradients, which correspond to the most critical task-related capabilities. Based on these principal components, we select the most relevant training samples as anchors for the walk algorithm, using them as the starting point for the

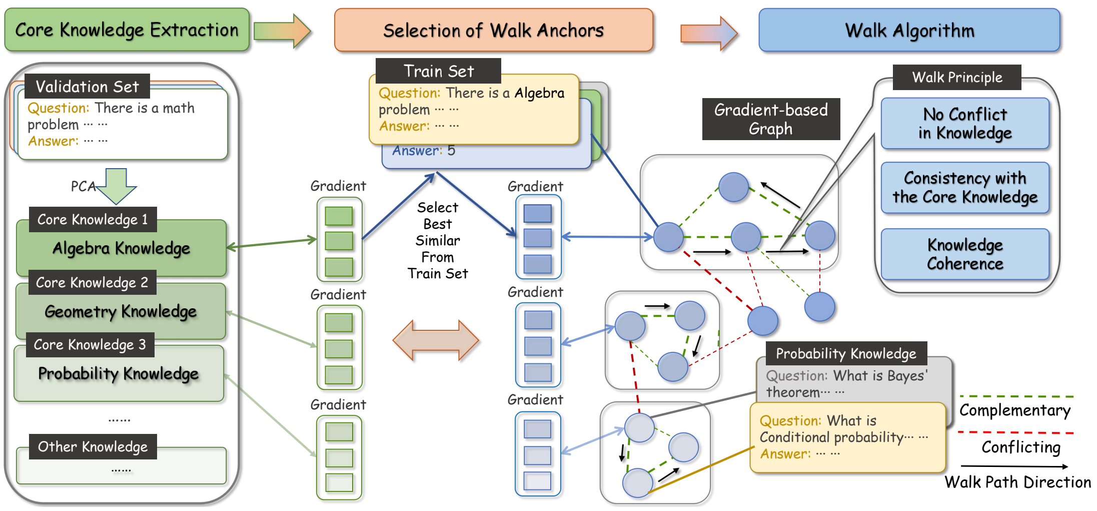

Figure 2: The left side illustrates the core knowledge extracted from the validation set. In the center, anchor selection for the gradient walk is performed by identifying the most similar data points from the training set, based on the core knowledge. These anchors are then used to conduct a gradient walk in the graph (right side), ensuring three key conditions: No Conflict in Knowledge, Consistency with Core Knowledge, and Knowledge Coherence. Finally, a gradient subgraph is selected in the lower-right corner, representing a subset of knowledge from the training set (e.g., probabilistic knowledge).

gradient walk. Unless otherwise stated, we use the first 50% of principal components in our experiments, as they capture the majority of the variance in the data, ensuring that the selected samples align with the core knowledge required for the task.

3.2 Data Selection with a Walk Algorithm on the Gradient-based Graph

To ensure that the selected training data effectively supports the target task, we adopt a structured data selection process based on a gradient-based graph, rather than relying solely on similarity-based methods. By utilizing the gradient graph constructed in the previous section, we capture both complementary and conflicting knowledge structures.

The gradient walk algorithm begins with an anchor that closely resembles the validation set. Using the weights corresponding to the principal components of the validation set, the number of training instructions required for each core knowledge component is determined. The algorithm then expands the instruction set by selecting data that contributes positively to model learning while maintaining consistency with the validation set.

The data selection process is governed by three heuristic principles. First, **No Conflict in Knowledge** ensures that new samples do not introduce contradictions. The similarity between the new node z and all existing nodes s should be non-negative. Second, **Consistency with Core Knowledge** prevents the selected training data from deviating significantly from the core knowledge by ensuring that the alignment (cosine similarity) of the growing instruction set with the Core Knowledge $K_{\mathcal{V}}$ does not degrade substantially. Specifically, the magnitude of the cosine similarity after adding a new sample must be at least a factor δ of the similarity before adding it. Third, **Knowledge Coherence** ensures that, building on the previous two principles, the selected data is most similar to the most recent instruction s^*. Formally, the data selection process is defined as:

$$z^* = \arg\max_{z \in \mathcal{Z}} \cos\left(\nabla\Gamma(z, \theta_t), \nabla\Gamma(s^*, \theta_t)\right), \quad (4)$$

subject to the following constraints:

$$\cos\left(\Gamma(z, \theta_t), \Gamma(s, \theta_t)\right) \geq 0, \quad \forall s \in \mathcal{S}, \quad (5)$$

$$\begin{aligned} &\left|\cos\left(\Gamma(\mathcal{S} \cup \{z\}, \theta_t), K_{\mathcal{V}}\right)\right| \geq \\ &\quad \delta \left|\cos\left(\Gamma(\mathcal{S}, \theta_t), K_{\mathcal{V}}\right)\right|, \end{aligned} \quad (6)$$

where z^* is the next selected node, $\mathcal{Z}$ is the nodes in the gradient-based graph, $\mathcal{S}$ is the current instruction set, and $K_{\mathcal{V}}$ is a core knowledge from the validation set.

The function $\cos(\Gamma(\mathcal{S}, \theta_t), K_{\mathcal{V}})$ represents the cosine similarity between the average gradient of the current instruction set $\mathcal{S}$ and the core knowledge from the validation set. s^* refers to the most recently added node in the instruction set, ensuring new selections align with the evolving training

		Llama3.1-8B					Gemma-7B					Mistral-7B				
ratio	Ins	BBH	GPQA	GSM8K	Math	MMLU	BBH	GPQA	GSM8K	Math	MMLU	BBH	GPQA	GSM8K	Math	MMLU
	BERT	63.05	30.55	56.79	20.22	61.87	49.79	34.22	53.15	20.92	60.75	56.77	24.85	43.44	13.30	58.30
1%	LESS	63.46	29.94	60.65	18.66	63.15	58.95	30.96	58.45	20.02	59.85	57.93	27.49	44.05	14.04	55.26
	G2IS	**64.78**	**31.57**	**62.02**	**20.96**	**63.42**	**60.25**	**35.03**	**61.64**	**22.88**	**62.17**	**58.59**	**29.44**	**46.40**	**14.68**	**58.64**
	BERT	59.65	26.68	50.95	20.46	62.33	44.39	31.57	55.80	22.16	59.85	53.72	26.27	45.79	14.02	58.00
5%	LESS	64.17	29.94	63.08	18.90	62.87	58.65	31.16	60.58	20.26	59.86	56.06	28.51	52.24	13.78	58.20
	G2IS	**65.07**	**34.22**	**64.06**	**21.00**	**63.36**	**59.38**	**34.42**	**62.70**	**23.00**	**62.13**	**57.40**	**31.57**	**54.44**	**14.14**	**59.28**
All		64.71	29.74	58.30	20.26	62.75	58.12	28.72	63.31	23.64	59.66	56.95	29.74	52.99	18.52	58.04
ratio	COT	BBH	GPQA	GSM8K	Math	MMLU	BBH	GPQA	GSM8K	Math	MMLU	BBH	GPQA	GSM8K	Math	MMLU
	BERT	63.68	28.51	53.90	16.36	63.04	58.50	28.92	57.77	21.08	59.61	56.11	23.42	40.79	12.30	57.87
1%	LESS	64.29	27.90	57.70	18.54	63.42	59.47	32.18	57.47	21.28	59.43	56.38	27.49	44.81	11.54	57.04
	G2IS	**65.66**	**32.59**	**62.70**	**21.38**	**64.22**	**61.60**	**33.20**	**63.76**	**23.44**	**62.34**	**58.04**	**28.72**	**49.20**	**12.64**	**59.09**
	BERT	63.60	29.33	61.11	17.60	62.48	56.77	29.94	60.12	21.40	58.94	54.42	28.51	44.58	11.68	56.22
5%	LESS	62.76	29.74	60.73	17.74	63.02	59.15	29.74	60.88	21.08	59.31	56.87	27.29	48.45	11.32	58.05
	G2IS	**65.14**	**30.55**	**62.47**	**20.68**	**64.01**	**60.44**	**35.23**	**64.90**	**23.10**	**61.85**	**57.68**	**28.92**	**49.41**	**13.36**	**59.48**
All		60.18	30.35	60.58	17.06	60.35	56.58	29.53	60.35	19.46	59.01	54.58	27.29	51.78	10.42	57.79
ratio	FLAN	BBH	GPQA	GSM8K	Math	MMLU	BBH	GPQA	GSM8K	Math	MMLU	BBH	GPQA	GSM8K	Math	MMLU
	BERT	64.03	29.53	50.99	15.58	63.50	59.88	30.75	52.16	19.14	59.95	55.60	23.01	39.50	12.50	56.89
1%	LESS	64.81	28.72	51.55	14.92	60.79	57.76	**32.99**	56.18	21.68	59.92	56.47	26.07	37.60	12.38	51.64
	G2IS	**64.84**	**31.16**	**60.12**	**19.30**	**64.36**	**61.39**	**32.99**	**62.09**	**22.70**	**62.06**	**57.89**	**28.72**	**43.90**	**13.26**	**59.19**
	BERT	62.94	29.53	50.57	17.26	62.90	59.15	30.96	52.54	17.62	60.31	54.83	26.68	37.07	12.98	53.87
5%	LESS	62.76	29.33	52.31	18.32	63.77	59.21	30.75	54.44	21.46	60.72	55.55	26.68	38.36	11.04	58.74
	G2IS	**65.27**	**30.96**	**62.40**	**20.40**	**64.03**	**60.70**	**34.01**	**63.68**	**22.44**	**61.78**	**59.13**	**27.29**	**48.98**	**13.46**	**59.77**
All		63.52	33.2	51.71	18.70	62.88	58.29	29.94	49.43	20.88	60.80	54.29	29.94	35.63	10.74	58.99

Table 1: Performance comparison of data selected from the top 1% and 5% most beneficial samples for enhancing performance on BBH, GPQA, GSM8K, Math, and MMLU tasks. Experiments were conducted using Llama3.1-8B, Gemma-7B, and Mistral-7B v0.3, fine-tuned on Infinity-Instruct, COT, and FLAN-v2 for single-task objectives.

set. The parameter δ acts as a threshold controlling the allowable degradation in alignment with core knowledge. Unless specified otherwise, we set $\delta = 0.8$ to balance diversity in selected data with alignment to the validation set.

For a detailed description of the algorithm, refer to Appendix C. By incorporating the joint distribution between instructions, it maximizes consistency with both the validation set and its inherent knowledge, while simultaneously avoiding conflicts, thereby enhancing the model's learning efficiency.

4 Experiment

To validate our method, we focus on two key areas: data selection for single-task optimization and gradient-based selection for multi-task instruction tuning. By comparing with baseline methods, we demonstrate the robustness and generalizability of our approach in both settings.

4.1 Experimental Setup and Baselines

To ensure a fair evaluation, we selected state-of-the-art language models: Llama3.1-8B (Dubey et al., 2024), Gemma-7B (Team et al., 2024), and Mistral-7Bv0.3 (Jiang et al., 2023). Details of the training and testing data are provided in Appendix A.

We compared our method with two baselines: Less (Xia et al., 2024), a leading data selection approach, and Sentence-BERT (Reimers and Gurevych, 2019), a widely used training data selection method. Training was performed using the Llama-Factory (Zheng et al., 2024), and model performance was evaluated with the Open LLM Leaderboard Harness (Gao et al., 2024), which provides a standardized assessment methodology. Further details on experimental configurations are in Appendix B.

4.2 Optimizing Data Selection for Single-Task Performance

Table 1 presents a comparative analysis of G2IS against baseline methods across multiple benchmark datasets. The results show that G2IS consistently outperforms Less and Sentence-BERT, regardless of whether 1% or 5% of the training data is selected. Notably, our method excels on multi-task datasets like FLAN, which is consistent with Wang et al. (2023), suggesting that targeted data selection outperforms full-dataset instruction tuning. Compared to the other two datasets, FLAN-V2 covers a broader range of tasks, better simulating the process of extracting specific domain instructions from large-scale data. Furthermore, G2IS demonstrates significant improvements on complex reasoning tasks like GSM8K and BBH, showcasing the advantages of using a gradient-based graph structure to jointly model instruction tuning data, leading to better performance on complex tasks.

A particularly noteworthy observation is that selecting just 1% of the training data often yields better performance than using 5% or even the entire dataset. This suggests that a small subset of highly relevant data plays a crucial role in enhancing task-specific performance, while the presence of excessive or irrelevant data can introduce noise and hinder the model's generalization capabilities.

Additionally, our findings show that models trained on the full dataset generally underperform compared to those trained on the top 1% and 5% subsets selected by G2IS. This further validates that blindly instruction tuning on the entire dataset does not yield optimal results. Instead, carefully curated, high-quality subsets align better with task requirements, especially for complex or specialized tasks. These results are consistent with the conclusions drawn by Tsai et al. (2024) and Zhou et al. (2024). They confirm the effectiveness of gradient-based, graph-driven data selection in enhancing model performance and training efficiency.

4.3 Enhancing Multi-Task Instruction tuning with Gradient-based Data Selection

To further assess the robustness of our method, we evaluated its performance in multi-objective optimization by combining the validation sets of GSM8K and BBH. GSM8K focuses on mathematical reasoning, requiring step-by-step problem-solving, while BBH is a general complex reasoning dataset. Although distinct, these tasks share significant structural similarities, making them suitable for multi-task optimization.

ratio		**Llama**		**Gemma**		**Mistral**	
	FLAN	BBH	GSM8K	BBH	GSM8K	BBH	GSM8K
	BERT	63.74	52.16	60.82	53.15	56.37	37.53
1%	LESS	64.05	53.53	56.87	53.53	56.70	39.95
	G2IS	**65.01**	**58.98**	**61.05**	**62.32**	**58.45**	**43.44**
	BERT	62.40	50.27	60.07	50.57	56.09	35.10
5%	LESS	59.07	46.55	56.14	51.10	53.89	34.50
	G2IS	**65.81**	**57.85**	**61.36**	**59.97**	**58.09**	**46.40**
All		63.52	51.71	58.29	49.43	54.29	35.63
	COT	BBH	GSM8K	BBH	GSM8K	BBH	GSM8K
	BERT	63.23	55.88	60.57	51.78	55.64	39.20
1%	LESS	63.69	58.91	59.35	60.05	56.01	44.73
	G2IS	**64.60**	**60.12**	**61.36**	**62.93**	**58.93**	**49.51**
	BERT	63.40	59.21	56.54	58.38	55.55	46.70
5%	LESS	63.46	57.16	58.65	56.48	55.24	45.03
	G2IS	**63.69**	**60.88**	**60.74**	**59.29**	**58.04**	**47.08**
All		60.18	60.58	56.58	60.35	54.58	51.78

Table 2: Performance of different methods in selecting data from FLAN or COT to simultaneously improve performance on BBH and GSM8K complex reasoning tasks.

As shown in Table 2, traditional methods like Less struggle with multi-task optimization, showing performance degradation in multi-task scenarios. In contrast, G2IS demonstrates robustness in handling multiple objectives, consistently delivering strong results across both. These findings confirm that our gradient-based graph approach effectively balances conflicting objectives, enabling efficient multi-task instruction tuning.

5 Ablation Study

Building on our experimental results, we conduct ablation studies to identify the key factors contributing to the effectiveness of our method. Specifically, we examine two aspects: (1) the impact of varying principal component ratios on task performance by selecting different proportions of principal components and analyzing their influence on the results,

Model	ratio	w/o	COT						FLAN					
			BBH	GPQA	GSM8K	Math	MMLU	Avg	BBH	GPQA	GSM8K	Math	MMLU	Avg
Llama	1%	w/o graph	65.64	30.55	57.85	20.1	61.42	0.95	64.03	30.55	58.15	18.96	61.6	0.97
		w/o gradient	64.57	32.53	58.91	20.24	63.94	0.97	64.75	30.14	57.01	19.12	63.84	0.98
		G2IS	**65.66**	**32.59**	**62.7**	**21.38**	**64.22**	**1.0**	**64.84**	**31.16**	**60.12**	**19.3**	**64.36**	**1.0**
	5%	w/o graph	64.44	30.35	55.72	18.22	58.89	0.94	64.37	28.11	53.53	17.4	60.81	0.91
		w/o gradient	64.20	28.11	54.51	16.44	63.71	0.91	63.35	28.31	55.04	17.66	64	0.93
		G2IS	**65.14**	**30.55**	**62.47**	**20.68**	**64.01**	**1.0**	**65.27**	**30.96**	**62.4**	**20.4**	**64.03**	**1.0**
Gemma	1%	w/o graph	58.61	31.77	59.89	21.46	60.92	0.95	60.88	31.16	61.41	21.68	59.67	0.97
		w/o gradient	58.93	32.79	62.62	22.5	61.45	0.98	59.48	30.14	58.38	22.58	61.71	0.96
		G2IS	**61.6**	**33.2**	**63.76**	**23.44**	**62.34**	**1.00**	**61.39**	**32.99**	**62.09**	**22.7**	**62.06**	**1.00**
	5%	w/o graph	58.15	31.16	56.1	20.2	60.56	0.91	59.75	33.6	56.25	21.48	60.74	0.96
		w/o gradient	59.48	32.59	59.29	21.56	60.60	0.95	59.67	30.96	58.07	21.26	61.35	0.95
		G2IS	**60.44**	**35.23**	**64.9**	**23.1**	**61.85**	**1.00**	**60.7**	**34.01**	**63.68**	**22.44**	**61.78**	**1.00**
Mistral	1%	w/o graph	57.06	28.52	43.52	12.46	59.02	0.97	57.72	28.33	39.8	11.92	58.31	0.95
		w/o gradient	55.83	25.87	43.52	12.56	58.71	0.96	57.27	27.49	**43.9**	12.82	58.3	0.98
		G2IS	**58.04**	**28.72**	**49.2**	**12.64**	**59.09**	**1.00**	**57.89**	**28.72**	**43.9**	**13.26**	**59.19**	**1.00**
	5%	w/o graph	57.28	28.51	42.3	12.44	59.01	0.95	57.1	27.24	44.35	12.34	58.99	0.95
		w/o gradient	57.4	28.11	44.96	11.78	57.26	0.94	57.99	26.99	45.11	12.46	58.45	0.96
		G2IS	**57.68**	**28.92**	**49.41**	**13.36**	**59.48**	**1.00**	**59.13**	**27.29**	**48.98**	**13.46**	**59.77**	**1.00**

Table 3: An ablation study was conducted on the COT and FLAN datasets using Llama3.1-8B, Gemma-7B, and Mistral-7B models."w/o graph" refers to a variant with no graph structure,where training samples are selected solely based on principal component analysis."w/o gradient" replaces the gradient-based representation with a Sentence-BERT-based similarity measure for comparison. The experiment evaluates data selection for BBH, GPQA, GSM8K, Math, and MMLU tasks."Avg" represents the average accuracy of each method relative to our approach.

and (2) the role of key components within the structured gradient-based graph approach. To explore this, we perform controlled experiments to isolate the effects of each factor. We compare our method with variants that either use PCA without the graph structure or replace gradient-based representations with BERT-based semantic similarity measures.

5.1 Impact of Principal Component Ratios

To investigate the impact of principal component ratios on task performance, we select the top 1% of data from the Infinity-Instruct dataset and evaluate it using the MMLU and GSM8K benchmarks. As shown in Figure 3, the optimal principal component ratio varies across tasks, though its overall impact remains limited. For tasks like MMLU, which require multi-domain knowledge and exhibit high noise, filtering out lower-variance components improves performance by reducing irrelevant or conflicting information. In contrast, for tasks like GSM8K, which focus on mathematical reasoning with minimal domain conflict, retaining more components helps preserve crucial task-related knowledge, enhancing performance. Despite these task-specific variations, the effect of adjusting the principal component ratio is constrained. In most cases, our method consistently outperforms traditional selection approaches, highlighting its robustness and effectiveness across different tasks.

5.2 The Role of Graph Structure and Gradient-based Representations

To assess the contributions of the two core components of our method—the structured graph-based framework and gradient-based knowledge representation—we conduct a comparative ablation ex-

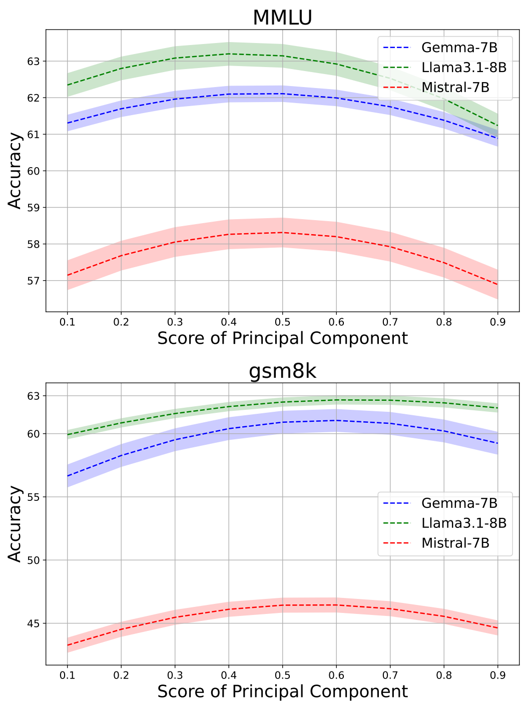

Figure 3: On the Infinity-Instruct dataset, we utilize the gradient walk algorithm based on principal components of knowledge extracted from different proportions of the validation set. The results of training after selecting data to enhance MMLU (top) and GSM8K (bottom) performance are presented.

periment. Specifically, we introduce two ablation settings: (1) **w/o graph**,where knowledge is extracted solely via PCA and selected based on gradient similarity, and (2) **w/o gradient**,where the gradient-based similarity measure is replaced with a BERT-based semantic similarity approach.

The results on the COT and FLAN datasets, shown in Table 3, demonstrate the necessity of both structured graph modeling and gradient-based representations. While both components improve performance, the structured graph is especially critical. As the number of selected instructions increases, the graph-based approach consistently outperforms by capturing interdependencies between instructions, reducing redundancy, and optimizing knowledge transfer. These findings highlight the effectiveness of our graph-based selection framework in enhancing instruction tuning.

6 Related Work

Recent studies (Park et al., 2023; Jain et al., 2024) have explored using model gradients for instruction-tuning data selection, showing that gradients capture the informational content of training samples. Xia et al. (2024) proposed a gradient similarity-based approach that selects data aligned with the validation set, achieving performance comparable to full-data instruction tuning using only 5% of the data. Joaquin et al. (2024) improved data selection for large models by leveraging the similarity between the training and validation sets in smaller models, enhancing selection efficiency.

However, these approaches focus on similarity and fail to capture the deeper interdependencies within the training data. Previous research (Zhao et al., 2024a) found that instructions are not independent, but instead exhibit interdependencies. By leveraging these dependencies, instruction tuning performance can be improved. Moreover, Hübotter et al. (2024) highlighted that similarity-based selection overlooked these dependencies, limiting the effectiveness of instruction tuning.

Several studies (Lu et al., 2023; Bhatt et al., 2024) enhance dataset complexity and diversity. While modifying data distributions for generalization, they prioritize adaptability over domain-specific optimization and omit explicit modeling of relationships between training samples.

With the rise of domain-specific LLMs (Wu et al., 2023; Zhang and Yang, 2023), instruction selection methods should account for dependencies in training data. Unlike similarity-based approaches, graph-based models capture interdependencies, reduce redundancy, and enhance knowledge transfer. This highlights the need for more effective selection strategies that extend beyond similarity, optimizing instruction tuning for specialized domains. Graph-based models, like G2IS, offer a promising solution by capturing complex relationships, leading to improved performance.

7 Conclusion

We introduced G2IS, a gradient-based instruction selection method aimed at enhancing domain-specific instruction tuning by constructing structured gradient-based graphs. Unlike traditional methods, which rely on similarity measures, G2IS captures the joint distribution and interdependencies between instructions, leading to more efficient data selection. Our experiments on state-of-the-art models demonstrate that G2IS significantly outperforms conventional instruction tuning approaches, achieving superior results with only 1% of the training data, especially on complex reasoning tasks like GSM8K and BBH. Notably, G2IS excels in selecting highly relevant data, showing that smaller, carefully curated subsets outperform larger, less relevant datasets. Ablation studies further confirm the critical role of both the structured graph framework and gradient-based representations in optimizing knowledge transfer and improving performance. These findings emphasize G2IS's potential to enhance task-specific learning, reduce data requirements, and enable more efficient instruction-tuning strategies, particularly in data-limited specialized domains.

8 Limitations

In this study, we adopt the LoRA method and compute gradients only for the LoRA layers, rather than for the full model parameters, to reduce computational cost. While our method demonstrates strong performance across various instruction-tuning tasks, we have not yet evaluated whether computing full-parameter gradients could further enhance data selection effectiveness. Additionally, due to computational constraints, our experiments are primarily conducted on 7B and 8B-scale models (e.g., Llama3-8B, Gemma-7B, Mistral-7B), and we have not yet tested our method on larger-scale LLMs (e.g., 13B, 65B, 175B). In future work, we plan to extend our study to full-gradient computation and larger-scale models to more comprehensively assess the applicability and optimization potential of the G2IS method.

9 Acknowledgements

The research in this article is supported by the New Generation Artificial Intelligence of China (2024YFE0203700), National Natural Science Foundation of China under Grants U22B2059 and 62176079.

References

Gantavya Bhatt, Yifang Chen, Arnav M Das, Jifan Zhang, Sang T Truong, Stephen Mussmann, Yinglun Zhu, Jeffrey Bilmes, Simon S Du, Kevin Jamieson, et al. 2024. An experimental design framework for label-efficient supervised finetuning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6549–6560, Bangkok, Thailand. Association for Computational Linguistics.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. 2024. What is your data worth to gpt? llm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Zayd Hammoudeh and Daniel Lowd. 2024. Training data influence analysis and estimation: A survey. *Machine Learning*, 113(5):2351–2403.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. 2024. Efficiently learning at test-time: Active fine-tuning of llms. *arXiv preprint arXiv:2410.08020*.

Saachi Jain, Kimia Hamidieh, Kristian Georgiev, Andrew Ilyas, Marzyeh Ghassemi, and Aleksander Madry. 2024. Improving subgroup robustness via data selection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Ayrton Joaquin, Bin Wang, Zhengyuan Liu, Philippe Muller, Nicholas Asher, Brian Lim, and Nancy Chen. 2024. In2core: Leveraging influence functions for coreset selection in instruction finetuning of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10324–10335.

William B Johnson. 1984. Extensions of lipshitz mapping into hilbert space. In *Conference modern analysis and probability, 1984*, pages 189–206.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Takio Kurita. 2021. Principal component analysis (pca). In *Computer vision: a reference guide*, pages 1013–1016. Springer.

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. 2024. Less is more: Data value estimation for visual instruction tuning. *arXiv preprint arXiv:2403.09559*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*.

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Yun-Da Tsai, Mingjie Liu, and Haoxing Ren. 2024. Code less, align more: Efficient llm fine-tuning for code generation with data pruning. *arXiv preprint arXiv:2407.05040*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*.

Zhao Yang, Du Li, Ding Xiao, Xiong Kai, Sun Zhouhao, Shi Jun, Liu Ting, and Qin Bing. 2024. Deciphering the impact of pretraining data on large language models through machine unlearning. *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9386–9406.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

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*, pages 4435–4439.

Hanyu Zhao, Li Du, Yiming Ju, Chengwei Wu, and Tengfei Pan. 2024a. Beyond iid: Optimizing instruction learning from the perspective of instruction interaction and dependency. *arXiv preprint arXiv:2409.07045*.

Yang Zhao, Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2024b. Supervised fine-tuning achieve rapid task adaption via alternating attention head activation patterns. *arXiv preprint arXiv:2409.15820*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

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 Appendix: Data Construction

A.1 Training Data Construction

- **Infinity-Instruct** [1] We randomly selected 1 million instruction-tuning samples from the InfInstruct-3M core subset for training.
- **COT** (Wei et al., 2022) **and FLAN-v2** (Longpre et al., 2023) We used the cleaned dataset provided by (Xia et al., 2024) to ensure data quality.

A.2 Validation and Test Data Construction & Evaluation Methods

- **BBH Dataset** (Suzgun et al., 2022)
  - Validation set: We used the original dataset's development set, which consists of 81 samples.
  - Evaluation method: 0-shot evaluation.
- **GPQA Dataset** (Rein et al., 2023)
  - Validation set: Since the original dataset does not include an official development set, we randomly selected 55 samples as the validation set, with the remaining samples used for testing.
  - Evaluation method: 0-shot evaluation.
- **GSM8K Dataset** (Cobbe et al., 2021)
  - Validation set: We randomly selected 100 samples from the training set as the validation set.
  - Evaluation method: 5-shot evaluation.
- **Math Dataset** (Hendrycks et al., 2021b)
  - Validation set: We randomly selected 200 samples from the training set as the validation set.
  - Evaluation method: 4-shot evaluation.
- **MMLU Dataset** (Hendrycks et al., 2021a)
  - Validation set: We used the original dataset's dev set, which consists of 1,531 samples.
  - Evaluation method: 0-shot evaluation.

To ensure consistency and fairness in the evaluation process, we used the harness evaluation framework (Gao et al., 2024), keeping all hyperparameters at their default settings.

[1] https://huggingface.co/datasets/BAAI/Infinity-Instruct

B Appendix: Train Setup

B.1 Experimental Setup

All experiments were conducted on an **A100-SMX4** GPU cluster. To ensure fair comparisons and reproducibility, we used the **Llama-Factory** (Zheng et al., 2024) for model training and followed the experimental configurations outlined in (Xia et al., 2024).

The model was fine-tuned using the **LoRA** method with a rank of 128, $\alpha = 512$, and a dropout rate of 0.1. We employed `bf16` precision and used the `cosine` learning rate scheduler with an initial learning rate of 2×10^{-5}. The training process spanned 3 epochs, with a batch size of 2 per device, gradient accumulation steps set to 16, and a warm-up ratio of 0.3. The maximum sequence length was set to 2048 tokens. For optimization, we used the DeepSpeed ZeRO-2 (Rajbhandari et al., 2020) configuration.

B.2 Warmup Strategy and Random Projection

To effectively initialize the momentum of the Adam optimizer, we employed a **warmup** strategy.Specifically, we randomly selected 5,000 samples from the training dataset and conducted 4 epochs of training to initialize the LoRA layer parameters and optimizer momentum.Given the high computational cost of tracking momentum at every training step, we approximate the model's momentum when encountering new data by using the momentum accumulated after these 4 epochs.Since the model sizes are similar, we adopted hyperparameters similar to those used in Xia et al. (2024).

Due to the significant magnification of gradients in the LoRA layer, we reduce the dimensionality of the LoRA gradients by applying Random Projection (Park et al., 2023), as suggested by Xia et al. (2024), to project the gradients down to 8192 dimensions.

B.3 Computational Cost Analysis

The primary computational cost of this experiment comes from gradient computation. However, since we adopt the LoRA method and compute gradients **only for the LoRA layers**, the overall computational overhead is **significantly reduced** compared to full-parameter instruction tuning.Furthermore, the additional cost compared to existing gradient-based similarity methods is negligible.In the construction of the gradient-based graph, we employ **efficient sparse computation techniques**, and the entire process can be completed **on the CPU**, eliminating the need for additional GPU resources.Overall, **G2IS introduces minimal additional computational cost** while significantly improving the efficiency and accuracy of data selection, making it both practical and scalable.

C Appendix:Gradient Walk Algorithm for Instruction Selection

Algorithm 1: Gradient Walk Algorithm for Instruction Selection

```
Input: $\mathcal{Z}$: Training dataset, $\mathcal{V}$: Validation dataset, $\delta$: Threshold for validation consistency, $k$:
       Principal component selection threshold (e.g., cumulative variance ratio), ratio: Percentage
       of training data to be selected
Output: $\mathcal{S}$: Selected instruction subset
Step 1: Compute Gradients
for each validation sample $v \in \mathcal{V}$ do
    Compute validation gradient: $\nabla_{\text{val}}(v) = \nabla\mathcal{L}(v, \theta)$
end
for each training sample $z \in \mathcal{Z}$ do
    Compute training gradient: $\nabla_{\text{train}}(z) = \nabla\Gamma(z, \theta)$ // Momentum-adjusted gradient
end
Step 2: Extract Core Knowledge from Validation Set
Perform PCA on $\{\nabla_{\text{val}}(v) | v \in \mathcal{V}\}$
Select top-$k\%$ principal components based on cumulative variance ratio
Represent validation core knowledge as $K_{\mathcal{V}}$
Normalize weights: $\alpha_i = \frac{\omega_i}{\sum_{j=1}^{k} \omega_j}$ so that $\sum_{i=1}^{k} \alpha_i = 1$
Step 3: Gradient Walk Selection
for each principal component $i$ do
    Compute selection budget: $N_{\text{select},i} = \text{ratio} \times |\mathcal{Z}| \times \alpha_i$
end
Initialize $\mathcal{S} \leftarrow \emptyset$
for each $v \in K_{\mathcal{V}}$ do
    Initialize $\mathcal{S}' \leftarrow \emptyset$
    Choose $s \in \mathcal{Z}$ maximizing $\cos(\nabla_{\text{train}}(s), v)$
    $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{s\}$
    while $|\mathcal{S}'| < N_{select,i}$ do
        Sort $z^*$ from $\mathcal{Z} \setminus \mathcal{S}'$ in descending order by $\cos(\nabla_{\text{train}}(z^*), \nabla_{\text{train}}(s))$
        Set found = False
        for each sorted $z^*$ do
            if $\forall s' \in \mathcal{S}', cos(\nabla_{train}(z^*), \nabla_{train}(s')) \geq 0$ and
             $\left|cos(\nabla_{train}(\mathcal{S}' \cup \{z^*\}), K_{\mathcal{V}})\right| \geq \delta \cdot \left|cos(\nabla_{train}(\mathcal{S}'), K_{\mathcal{V}})\right|$ then
                $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{z^*\}$
                Update $s = z^*$
                Set found = True
                break for-loop
            end
        end
        if not found then
            Choose $z^* = \arg\max_{z \in \mathcal{Z} \setminus \mathcal{S}'} \cos(\nabla_{\text{train}}(z), K_{\mathcal{V}})$
            Update $s = z^*$
            $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{z^*\}$
        end
    end
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}'$
end
return $\mathcal{S}$
```