mathematics

MDPI

Article

Enhancing Large Language Model Comprehension of Material Phase Diagrams through Prompt Engineering and Benchmark Datasets

Yang Zha [1], Ying Li [2,*] and Xiao-Gang Lu [3,*]

[1] Materials Genome Institute, Shanghai University, Shanghai 200444, China; zhayang@shu.edu.cn
[2] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[3] School of Materials Science and Engineering, Shanghai University, Shanghai 200436, China
* Correspondence: yinglotus@t.shu.edu.cn (Y.L.); xglu@t.shu.edu.cn (X.-G.L.)

Abstract: Large Language Models (LLMs) excel in fields such as natural language understanding, generation, complex reasoning, and biomedicine. With advancements in materials science, traditional manual annotation methods for phase diagrams have become inadequate due to their time-consuming nature and limitations in updating thermodynamic databases. To overcome these challenges, we propose a framework based on instruction tuning, utilizing LLMs for automated end-to-end annotation of phase diagrams. High-quality phase diagram images and expert descriptions are collected from handbooks and then preprocessed to correct errors, remove redundancies, and enhance information. These preprocessed data form a golden dataset, from which a subset are used to train LLMs through hierarchical sampling. The fine-tuned LLM is then tested for automated phase diagram annotation. Results show that the fine-tuned model achieves a cosine similarity of 0.8737, improving phase diagram comprehension accuracy by 7% compared to untuned LLMs. To the best of our knowledge, this is the first paper to propose using LLMs for the automated annotation of phase diagrams, replacing traditional manual annotation methods and significantly enhancing efficiency and accuracy.

Keywords: large language models; material science; phase diagram; prompt engineering; benchmark

MSC: 68T50

Citation: Zha, Y.; Li, Y.; Lu, X.-G. Enhancing Large Language Model Comprehension of Material Phase Diagrams through Prompt Engineering and Benchmark Datasets. *Mathematics* **2024**, *12*, 3141. https://doi.org/10.3390/math12193141

Academic Editor: Florin Leon

Received: 8 September 2024
Revised: 2 October 2024
Accepted: 4 October 2024
Published: 8 October 2024

1. Introduction

Large Language Models (LLMs) are advanced artificial intelligence (AI) technologies designed to understand and generate natural language [1,2]. Recently, we have witnessed a surge in the development of LLMs, including OpenAI's GPT-3 [3] and GPT-4 [4], Google's Gemini [5], and Meta's LLaMA-1 [6] and LLaMA-2 [7]. These models excel in a variety of downstream tasks, such as natural language understanding (NLU) [8,9], natural language generation (NLG) [10,11], complex reasoning [12,13], biomedical tasks [14,15] and code generation [16,17]. Phase diagrams are critical tools in materials science for understanding the performance of materials under different conditions [18]. Currently, the annotation of phase diagrams relies mainly on manual effort. Phase diagram images are generated using thermodynamic calculation software such as Thermo-Calc [19] and Pandat [20], after which domain experts describe them using text based on their understanding of the diagrams. Eventually, these phase diagram images and textual descriptions are compiled into reference books [21] for researchers to consult and use. With the rapid development of materials science, traditional manual annotation methods can no longer meet the needs of new material development. According to current studies, no literature or research has yet been published on the use of LLMs to assist in the annotation of material phase diagrams. Therefore, the efficient recognition and annotation of material phase diagrams using LLMs is expected to become a future research trend.

Common manual phase diagram annotation methods have the following disadvantages. *(i)* These methods consume significant human and material resources and require highly skilled experts proficient in computational thermodynamics methods and theory, as well as rich experimental experience [22]. *(ii)* Manual annotation cannot achieve batch annotation of phase diagrams and is limited to focusing on commonly used systems. *(iii)* Manual annotation methods cannot ensure timeliness; the content of the annotation is limited by the current version of the thermodynamic database. After the database is updated, the phase diagrams change accordingly. If errors in the phase diagrams are not corrected in time, it can mislead users and impact the guiding role of phase diagrams in production practices [23–26].

To address these issues, users of phase diagrams often prefer to consult reference books published by authoritative or well-known research teams in the field of computational thermodynamics [27]. These publishing institutions regularly collaborate with researchers to check and revise errors and omissions in the reference books every few years to ensure their quality. However, frequent updates and annotations of reference books are very time-consuming and labor-intensive. Thus, developing a new method of phase diagram annotation that overcomes the shortcomings of manual annotation is of significant importance.

Models such as OpenAI's GPT-3 [3], GPT-4 [4], Google's Gemini [5], and Meta's LLaMA-1 [6] and LLaMA-2 [7] demonstrate significant advancements in LLMs. The pretraining of these LLMs requires extensive computational resources, making the fine-tuning of pretrained models for specialized tasks, such as phase diagram annotation, a more practical and cost-effective strategy [28]. Furthermore, multimodal learning is a critical factor in these applications. Studies indicate that GPT-4 surpasses other LLMs across several key metrics, including generation quality, semantic comprehension [29], contextual coherence, and reasoning [30]. Its instruction-based fine-tuning exhibits substantial potential for enhancing phase diagram annotation.

This paper proposes a prompting engineering framework based on instruction fine-tuning to achieve end-to-end automated annotation of phase diagrams by LLMs, thereby alleviating the time-consuming and timeliness issues of annotation. First, we obtain high-quality phase diagram images and their corresponding expert descriptions from phase diagram reference books [31]. Secondly, we preprocess these expert descriptions, correcting errors, removing redundant content, and supplementing missing information [32]. Thirdly, these 13 sets of phase diagram images and expert descriptions are integrated into a golden dataset for the phase diagram comprehension task. Through a carefully designed instruction fine-tuning framework, we use hierarchical sampling to sample some phase diagrams and descriptions, forming a training set for fine-tuning the LLM. Numerous experiments have demonstrated that our approach achieves the best performance on the phase diagram comprehension task.

The main contributions of this paper are as follows:

1. Given that manual annotation of phase diagrams is both time-consuming and labor-intensive, and cannot be updated in real time, we explored an LLM-based automated annotation method that addresses the problem of real-time updates.
2. To the best of our knowledge, this is the first paper to propose the use of LLMs for the automated annotation of phase diagrams, replacing traditional manual annotation methods. In this paper, we employ an instruction fine-tuning framework based on hierarchical sampling, which efficiently fine-tunes the LLM to achieve higher precision in end-to-end phase diagram annotation compared to an LLM without instruction fine-tuning.
3. Through extensive experiments, the proposed instruction fine-tuning framework achieves the best results in the phase diagram comprehension task, with the average cosine similarity in the test set reaching 0.8737. Compared to the LLM without tuning, the performance of the fine-tuned LLM improved by 7%. The contributions described above are illustrated in Figure 1.

The rest of the paper is divided into four sections: Section 2 introduces related work on LLMs, prompt tuning, and material phase diagrams; Section 3 introduces the golden data for the phase diagram comprehension task and the hierarchical sampling-based prompting engineering process; Section 4 details the experimental process, including hardware and software setups, benchmark testing, main experimental results, and ablation studies; and Section 5 provides the conclusion.

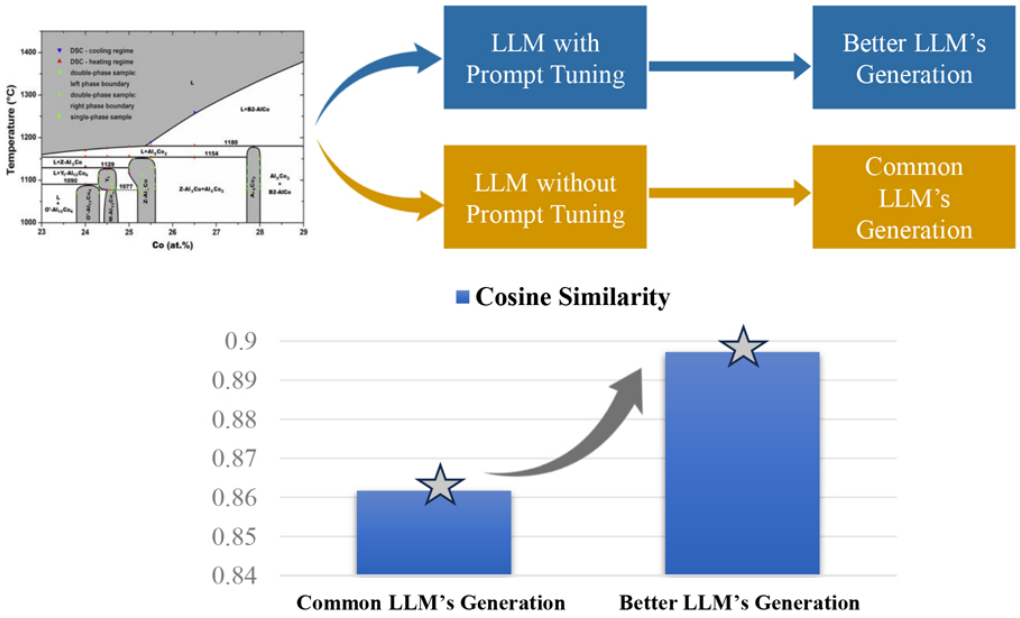


Figure 1. This figure highlights the key stages of processing phase diagrams using LLMs as described in this paper. The top section of the figure illustrates the workflow, where a phase diagram is input into two models: an un-tuned LLM and an instruction-tuned LLM. The comparison of outputs shows that the instruction-tuned LLM provides more accurate and specialized results. The bar chart below displays the cosine similarity metrics for both models on the phase diagram comprehension task, demonstrating that the instruction-tuned LLM outperforms the un-tuned LLM in task performance.

2. Related Work and Motivation

2.1. Large Language Models

Large Language Models (LLMs) represent cutting-edge AI technologies aimed at understanding and generating human language. In recent years, LLMs have undergone rapid evolution, particularly with OpenAI's introduction of GPT-3 [3] and GPT-4 [4]. OpenAI has transitioned from GPT-3 to GPT-4, a multimodal model that processes both image and text inputs, thereby enhancing natural language processing capabilities. GPT-4 is fine-tuned through pre-trained models and Reinforcement Learning with Human Feedback (RLHF) [33], achieving human-level performance in various professional and academic benchmark tests. However, significant effort is still required to adapt GPT-4 for specific domains such as materials science.

Google's Gemini multimodal model family [5] addresses the challenges of developing universal AI models through cross-modal joint pretraining and optimization, achieving significant benchmarks, including surpassing human experts in the Massive Multitask Language Understanding (MMLU) test [34]. However, research highlights limitations in reliability and reasoning, necessitating further investigation to enhance accuracy and authenticity. Open-source models such as Meta's LLaMA-1 [6] and LLaMA-2 [7], trained on public datasets, have set new performance standards. Notably, the 13-billion-parameter LLaMA model outperforms GPT-3 in most benchmarks. Models from LLaMA series have shown the effectiveness of public datasets in training high-performance models, advancing LLM research. LLaMA-2, designed for conversational scenarios, has improved multiple metrics through Safety Fine-Tuning (SFT) and RLHF, yet it remains limited in special-

ized domains such as materials science. These models excel in tasks such as NLU [8,9], NLG [10,11], complex reasoning [12,13], and code generation [16,17], which can be attributed to their extensive and diverse training data.

2.2. Prompt Tuning

Prompt engineering [35] aims at designing effective inputs, known as prompts, steering LLMs toward generating desired outputs. This method enhances inference efficiency by tailoring input prompts or queries to match the specific capabilities and nuances of LLMs. In straightforward tasks like semantic classification, prompt engineering can substitute for fine-tuning to produce high-precision outputs [36]. Few-shot prompting [28] involves supplying a small number of examples to an LLM to help it understand the task at hand. These examples, chosen from the training corpus due to their resemblance to the test cases, enable the LLM to make accurate predictions [37]. This method is a cost-effective strategy allowing LLMs to undertake various tasks without additional training or fine-tuning. The process of demonstration selection aims to identify high-quality examples for few-shot prompting. Typically, only a handful of well-chosen examples are needed to create a concise and clear prompt, facilitating efficient inference.

Demonstration selection techniques are categorized into unsupervised and supervised methods. Unsupervised methods use predefined similarity functions, such as L2 distance, cosine distance, and Minimum Description Length (MDL) [38], to identify relevant examples from training data. The KATE method (kNN-augmented Example Selection) by Liu et al. [39] enhances GPT-3's contextual learning by retrieving semantically similar examples, reducing performance variability, and improving tasks such as table-to-text generation and open-domain question answering. KATE boosts performance without fine-tuning GPT-3, making it effective for long-text generation, although GPT-3 still lags behind the fine-tuned T5 model [40] in sentiment analysis. Su et al. [41] advanced KATE with a two-step framework (Vote-K), optimizing contextual learning via selective annotation and prompt retrieval, enhancing performance and reducing annotation costs compared to random selection. Supervised methods train a domain-specific retriever for demonstration selection. Rubin et al. [42] introduces Efficient Prompt Retrieval (EPR), improving contextual learning by retrieving relevant training examples, although it sometimes underperforms compared to fine-tuned models. This underscores the need for further enhancements. Li et al. [43] presents the Unified Demonstration Retriever (UDR), a multi-task retriever trained through list-wise ranking and iterative mining, significantly improving contextual learning. UDR demonstrates strong performance and transferability across language models and datasets, but it lacks transparency. Generally, supervised methods yield better results than unsupervised ones, but they require frequent adjustments to handle out-of-domain data, reducing inference efficiency.

2.3. Phase Diagrams

Phase diagrams are essential tools in materials science, offering insights into stable phases and phase transitions across varying temperatures, pressures, and compositions [18]. These diagrams are critical for deciphering the internal structure of materials and forecasting their performance [44]. For example, the Fe-C phase diagram (Figure 2) [45] assists engineers in selecting optimal alloy compositions and thermal treatments to achieve desired microstructures and mechanical properties [21]. Under extreme conditions, such as high temperatures or pressures, materials may experience phase transitions that significantly impact their mechanical properties and longevity [46]. Phase diagrams elucidate these transitions, thereby expanding the potential applications of materials. Furthermore, phase diagrams play a pivotal role in both educational and research contexts. In materials science education, they not only convey fundamental thermodynamic principles, but also foster an intuitive understanding among students, preparing them to design experiments and analyze data.

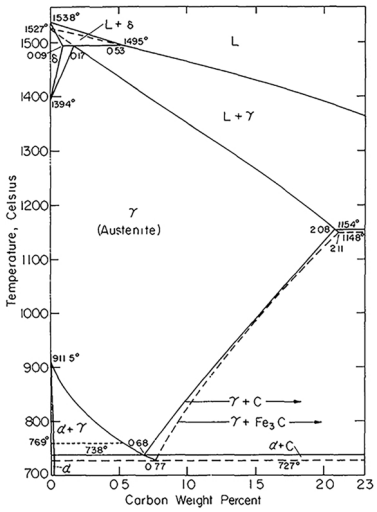


Figure 2. Fe-C phase diagram [45]. The horizontal axis represents the mass fraction of carbon in iron, while the vertical axis corresponds to temperature. By selecting the appropriate alloy composition and temperature, the microstructure and properties of the alloy can be tailored to meet specific requirements.

2.4. Motivation

Materials science encompasses a vast range of research areas, and the Materials Genome Initiative (MGI) provides a strategic framework to streamline these efforts [47]. The MGI aims to reduce both the time and cost of developing new materials by 50%. A critical tool in this process is phase diagram calculations, which help visualize simulation outcomes; however, non-experts often require assistance from specialists to interpret these results. This challenge of communication also extends to other simulation techniques, such as first-principles calculations [48] and molecular dynamics [49]. AI, particularly LLMs, offers potential to aid in interpreting these complex outputs, thus bridging the gap between simulations and practical applications [4]. This paper explores the use of LLMs to automate phase diagram annotations, thereby enhancing the integration between materials simulations and research development. Due to the resource-intensive nature of retraining models, we employed a prompt-engineering approach to fine-tune LLMs [28]. A key challenge was acquiring expert-validated manual annotations, especially given the limited dataset. By applying hierarchical sampling, we optimized the instruction fine-tuning process, addressing common interdisciplinary research challenges that require collaboration between materials scientists and AI specialists.

3. Methods

3.1. Production of Golden Data

3.1.1. Data Acquisition

The original dataset utilized in this study is sourced from the Phase Diagram Handbook, which contains numerous binary metal system phase diagrams, accompanied by expert textual descriptions [31]. These diagrams provide critical information about the phases present in various binary metal systems. In particular, we have compiled and digitized 13 commonly used binary metal phase diagrams along with their corresponding expert textual descriptions. Figure 3 provides an example of the Cr-Ni binary phase diagram and the corresponding textual information from the reference source.

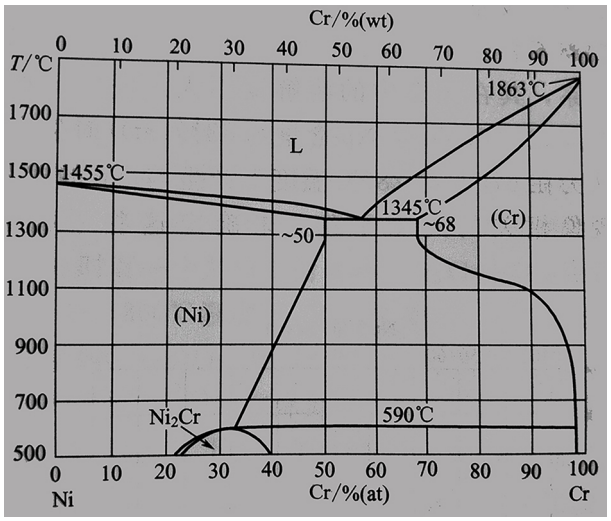


Figure 3. Cr-Ni phase diagram and its textual description from reference book: "In the system, an intermediate phase Ni2Cr exists, which forms in the solid state. The eutectic between (Ni) and (Cr) crystallizes at 1345 °C with a Cr content of 56%. The solubility of Ni in (Cr) is about 32% (at) at the eutectic temperature, about 10% (at) at 1000 °C, and about 2% (at) at 500 °C. The solubility of Cr in (Ni) is about 50% (at) at the eutectic temperature and about 36% (at) at 700 °C. The ordering phenomena in alloys near the composition of Ni2Cr have been studied in references. It has been determined that the ordering temperature for the stoichiometric composition of Ni2Cr is 590 °C. The homogeneity range of the compound Ni2Cr is 25% 36% (at) Cr. There is information on two types of metastable phases. One phase, with a σ-CrFe type structure, occurs near the alloy composition of 70% (at) Cr. The second phase, found in vacuum-sprayed thin films, has a Cr3Si type structure and occurs in compositions with over 70% (at) Cr".

3.1.2. Data Preprocessing

After acquiring the phase diagrams and textual descriptions, we conducted an initial review to ensure completeness and consistency. Certain phase diagrams, such as the Cr-Re phase diagram (Figure 4), lacked important details. For instance, the σ phase compound is not explicitly labeled on the diagram, although it should appear within the x-axis range of 58–70 and below 2356 on the y-axis. To preserve the integrity of the original data, we refrained from adding any markings that were not present in the source diagrams.

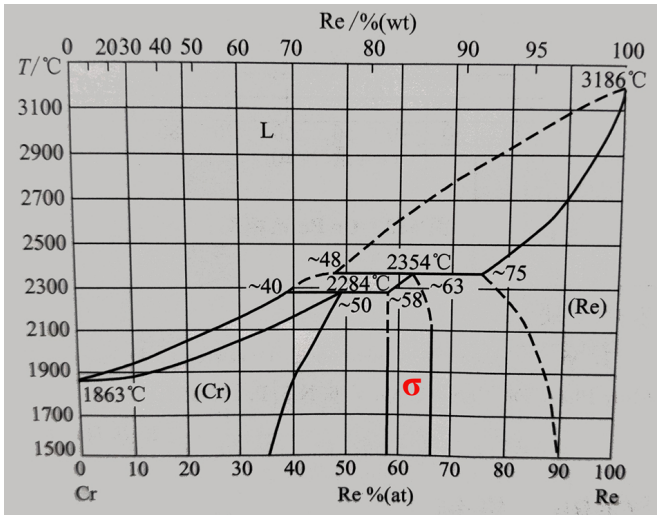


Figure 4. Phase diagram of Cr-Re system. The σ phase, which was not originally marked, has been identified and highlighted in red.

For the textual descriptions, we focused on extracting information directly relevant to the diagrams. This includes phase compositions, stability regions (i.e., phases present across various compositions and temperatures), and phase transformation reactions. Any extraneous information, such as descriptions from external references, experimental results, crystal structures, and lattice constants, was systematically excluded. Occasionally, specific details such as phase transformation processes were omitted, as in the Cr-Ni phase diagram (Figure 3), where only the name of the transformation (eutectic reaction) was provided. In such instances, we manually supplemented the missing information.

3.1.3. Data Normalization

Following the preprocessing step, we revised and normalized the expert textual descriptions to ensure clarity and uniformity. Each description was structured into two main sections. For example, the revised textual description of the Cr-Ni phase diagram (Figure 5) includes:

- Composition and Stability Regions of Phases: This section lists the phase names, the melting points of the pure components, and the solubility ranges of the solid solution phases;
- Phase Transformation Reactions: This section provides details of all phase transformations in the diagram, including the transformation temperatures, points of transformation, and reaction ranges.

By adhering to this standardized approach, we systematically reviewed and adjusted each phase diagram and its corresponding textual description. Ultimately, we produced 13 high-quality datasets suitable for phase diagram comprehension tasks. All expert textual descriptions were saved in TXT format to facilitate easy access and use.

Data Acquisition of Cr-Ni Phase Diagram

In the system, an intermediate phase Ni2Cr exists, which forms in the solid state. The eutectic between (Ni) and (Cr) crystallizes at 1345°C with a Cr content of 56%. The solubility of Ni in (Cr) is about 32% (at) at the eutectic temperature, about 10% (at) at 1000°C, and about 2% (at) at 500°C. The solubility of Cr in (Ni) is about 50% (at) at the eutectic temperature and about 36% (at) at 700°C. The ordering phenomena in alloys near the composition of Ni2Cr have been studied in references. It has been determined that the ordering temperature for the stoichiometric composition of Ni2Cr is 590°C. The homogeneity range of the compound Ni2Cr is 25%~36% (at) Cr.

There is information on two types of metastable phases. One phase, with a σ-CrFe type structure, occurs near the alloy composition of 70% (at) Cr. The second phase, found in vacuum-sprayed thin films, has a Cr3Si type structure and occurs in compositions with over 70% (at) Cr.

Data Normalization of Cr-Ni Phase Diagram

1. Composition and stable existence regions of phases:

In the Cr-Ni system, there is an intermediate phase, Ni2Cr. The system also includes a liquid phase, Ni-based solid solution (Ni), and Cr-based solid solution (Cr). At the eutectic temperature of 1345°C, with a Cr content reaching 68%, Ni achieves its maximum solubility; at the same eutectic temperature of 1345°C, the solubility of Cr in (Ni) is about 50% (atomic percent). The melting point of pure Ni is 1455°C, and the melting point of pure Cr is 1863°C.

2. Transition temperatures and sequences (2 items):

At 1345°C with a chromium content of 56%, the liquid phase (L) undergoes a eutectic reaction, solidifying into solid phases of nickel (Ni) and chromium (Cr). At a temperature of 590°C, an ordered phase, Ni2Cr, forms within the continuous solid solution (Ni,Cr).

Figure 5. Normalization of expert textual descriptions in the Cr-Ni system. The left side shows the original textual description of the Cr-Ni phase diagram, while the right side presents the normalized textual description.

3.1.4. Postprocessing of LLM's Generation

Taking the Al-Co binary system as an example, suppose an image of an Al-Co phase diagram is input into an LLM (ChatGPT-4.0) with appropriate instructions. For example: 'I will give you a phase diagram of the Al-Co binary system, give as detailed a description as possible based on the phase diagram, your description should be professional and accurate'. The LLM will provide comprehensive, accurate, and professional textual descriptions based on its knowledge of materials science databases and image recognition capabilities, as shown in Figure 6. LLM generation mainly consists of three parts. The first part is a detailed description of the range and units of the axes; the second part is a description of the content of the phase diagram; the third part is a summary of the phase diagram. We edit the

original generation, retaining only the second part, removing the redundant descriptions of the axes from the first part, and the summary descriptions from the last part. Ultimately, the LLM's textual descriptions for 13 phase diagrams are saved in TXT file format, ready to be directly used to evaluate the performance on phase diagram comprehension tasks.

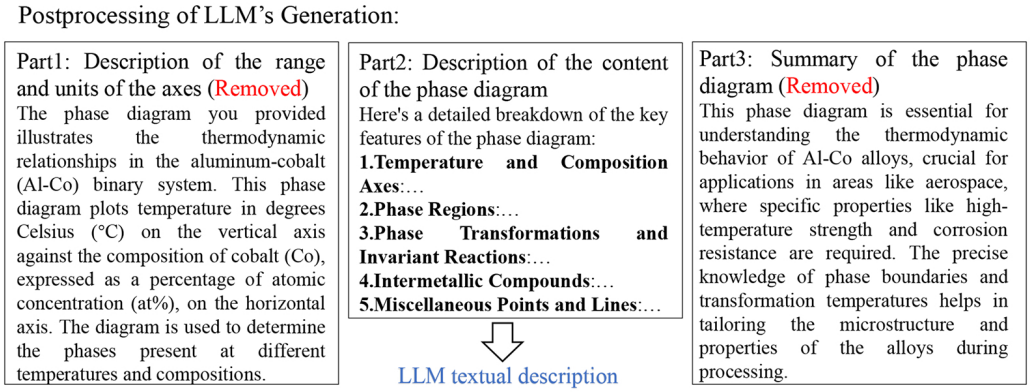


Figure 6. Textual description of the phase diagram of the Al-Co system by LLM. Output of LLM is dividied into three parts: description of the range and units of the axes, description of the content of the phase diagram, and summary of the phase diagram.

3.2. Workflow of Prompt Engineering

The ability of LLMs to interpret phase diagrams depends on domain-specific knowledge in materials science and image recognition capabilities. Retraining an LLM for material science applications is resource-intensive [28]. Therefore, this article adopts a more cost-effective and efficient few-shot prompt engineering approach to fine-tune an LLM for phase diagram comprehension tasks within the materials domain [28]. ChatGPT-4.0, a powerful multimodal LLM, has inherent knowledge of materials science and strong image recognition abilities [4]. In this study, the ChatGPT-4.0 API, provided by OpenAI, is used to adapt the model for tasks involving phase diagram comprehension.

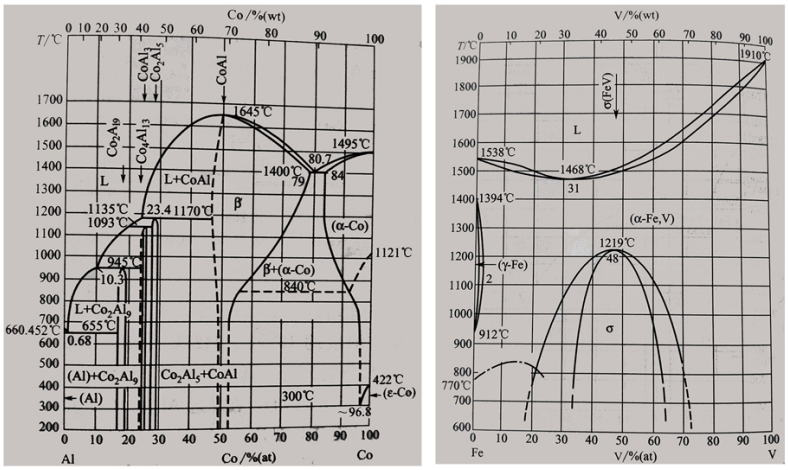
Specifically, we employ prompt tuning as a more efficient alternative to traditional model retraining methods [35]. Unlike conventional retraining, which alters the model's underlying architecture, prompt tuning focuses on refining the input prompts given to the model. This approach efficiently leverages the LLM's pre-existing knowledge, enabling faster adaptation for phase diagram interpretation in materials science. A key aspect of prompt tuning in this study is the use of few-shot learning [36]. In this method, a limited number of domain-specific examples, or "shots", are included in the prompt to guide the model toward accurate comprehension of phase diagrams. These examples are carefully chosen to improve the LLM's ability to handle the complex relationships depicted in the diagrams.

3.2.1. The Complexity of Phase Diagrams

Typically, outputs from LLMs that have not been fine-tuned on specific domain knowledge bases do not meet the professional requirements of that domain, especially when LLMs handle phase diagram comprehension tasks in the materials domain. The depth and breadth of domain knowledge contained in the input phase diagrams vary. For common, simple phase diagrams, LLMs can directly output high-quality textual descriptions from their database; however, for rare, complex phase diagrams, LLMs need robust materials science knowledge and strong reasoning abilities. Additionally, the image recognition capabilities of LLMs are crucial for accurately recognizing phase diagrams.

The golden data constructed in this paper include 13 phase diagrams and their corresponding expert textual descriptions. Based on the number of phase transition reactions, these 13 phase diagrams can be divided into simple and complex systems, as shown in Figure 7. Complex systems like Al-Co (Figure 7a) contain more information compared to simple systems like Fe-V (Figure 7b). Since it is challenging to provide a specific metric

to measure the information content of phase diagrams, we set a threshold (three in this case) based on the number of phase transition reactions to distinguish between simple and complex systems.



(a)Al-Co (b)Fe-V

Figure 7. Complex system Al-Co and simple system Fe-V: (**a**) Al-Co, with seven phase transformations. A peritectic transformation occurs at 945 °C; at 1093 °C, a peritectic reaction takes place; another peritectic reaction occurs at 1135 °C; at 1170 °C, yet another peritectic reaction occurs; at 655 °C, an eutectic reaction occurs; at 1400 °C, an additional eutectic reaction takes place; and at 300 °C, an eutectoid reaction is observed. (**b**) Fe-V, with one phase transformation. A solid-state phase transformation occurs at 1219 °C.

To further enhance the LLM's performance on phase diagram comprehension tasks, a few-shot prompting engineering process is designed. This process does not require additional training or fine-tuning of the LLM; it only needs a set of examples to guide its understanding of the task. These examples are use hierarchical sampling, randomly selecting from the simple and complex systems in the golden data.

The detailed process is shown in Figure 8. The golden data dataset contains 13 phase diagram images; however, we do not use all 13 images as the test set. Instead, we perform hierarchical sampling and select 3 images and their corresponding expert textual descriptions to pre-train the LLM through instruction tuning. The trained LLM is then be directly used for the comprehension tasks of the remaining 10 phase diagrams, and the LLM's generations are compared with the expert textual descriptions in the golden data to calculate task metrics.

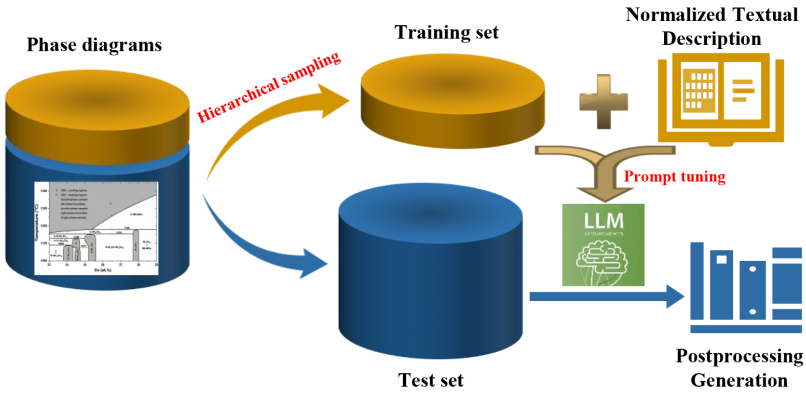


Figure 8. Workflow of prompt tuning. Phase Diagrams: Phase diagram datasets are gathered (as discussed in the previous Section 3). Hierarchical Sampling: Diagrams are sampled based on complexity; subsets are used for training and testing the LLM. Instruction Tuning: The fine-tuned model is then employed for automated phase diagram annotation.

3.2.2. Hierarchical Sampling

In the process of selecting demonstration cases from the 13 phase diagrams (the "Hierarchical sampling" stage), we do not randomly extract but use a hierarchical sampling strategy. We first divide the 13 phase diagrams into collections of complex phase diagrams systems (phase transition reactions > 3) and simple phase diagrams systems (phase transition reactions ≤ 3), as shown in Table 1. The first row lists the phase diagram names, the second row lists the number of phase transition reactions contained in the phase diagrams, the third row indicates whether the number of phase transition reactions exceeds the threshold of three, and the fourth row shows whether the phase diagram is classified as complex or simple systems. For example, the first column shows that the Al-Co phase diagram contains seven phase transition reactions, which is more than three; thus, it is classified as a complex system. Then, following a 20% ratio, we randomly select one from the complex system and two from the simple system. These selected phase diagram images and expert textual descriptions are input into the LLM as demonstration cases. The trained LLM is then applied to the phase diagram comprehension tasks on the remaining test set.

Table 1. Classification of complex and simple systems. We categorize the 13 phase diagrams into simple systems ($\leq$3 phase transitions, highlighted in yellow) and complex systems ($>$3 phase transitions, highlighted in green).

System	Al-Co	Cr-Ru	Cr-Ta	Cr-Ti	Fe-Ti	Hf-Re	Al-Ru	Co-Cr	Co-Fe	Cr-Fe	Cr-Ni	Cr-Re	Fe-V
Phase transformation numbers	7	5	4	5	4	5	3	3	3	3	2	2	1
thresholds	>3	>3	>3	>3	>3	>3	≤3	≤3	≤3	≤3	≤3	≤3	≤3
types	complex	complex	complex	complex	complex	complex	simple	simple	simple	simple	simple	simple	simple

3.2.3. Prompt Tuning

In the "prompt tuning" stage, assuming the Co-Cr phase diagram and Cr-Fe phase diagram are extracted from the simple system, and the Cr-Ti phase diagram from the complex system, these three sets of data (phase diagrams + expert textual descriptions) serve as demonstration examples for the tuning of the LLM, as shown in Figure 9.

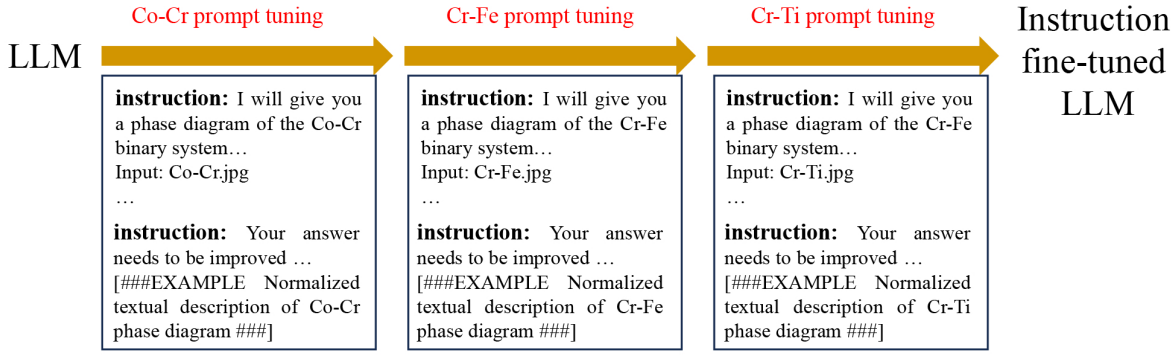


Figure 9. Examples used in instruction tuning of the LLM. The original LLM undergoes instruction tuning with three successive sets of golden data, resulting in a fine-tuned model that will be used for subsequent tasks.

Subsequently, the Al-Co phase diagram (from the test set) is input into the trained LLM. With the instructions unchanged, the output from the LLM shows significant improvement. As shown in Figure 10, the left side displays the textual description of the Al-Co phase diagram by the LLM before instruction tuning, and the right side shows the textual description after instruction tuning. The improved description exhibits noticeable enhancements in organization, standardization, and accuracy.

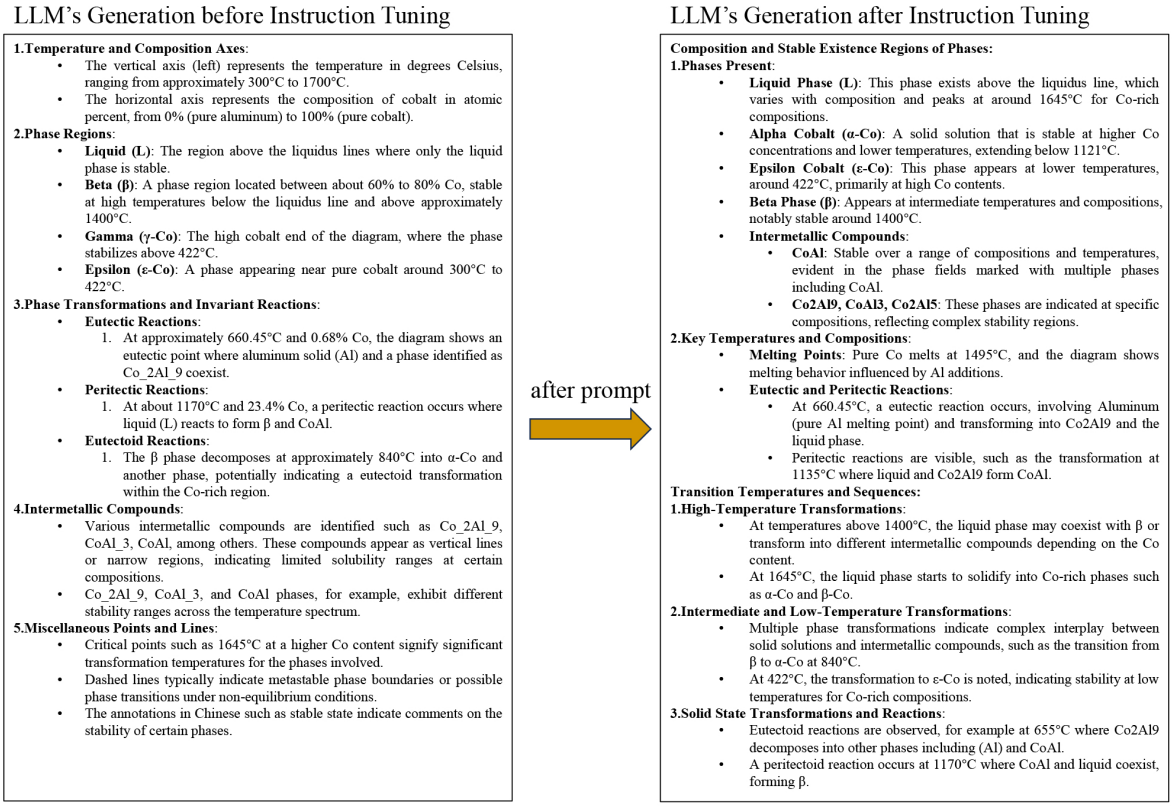


Figure 10. LLM's generations for the Al-Co system before and after instruction tuning. (**Left**) Textual description of the Al-Co phase diagram by the LLM before instruction tuning. (**Right**) Improved description after instruction tuning, showing enhanced organization, standardization, and accuracy.

4. Experiments

4.1. Experimental Setup

The experimental design of this study was executed on the Windows operating system, involving hardware configurations that included an Intel(R) Core(TM) i5-12500H 12th Gen processor, 16.0 GB RAM, and NVIDIA GeForce RTX 3060 GPU, sourced from Intel Corporation (Santa Clara, CA, USA) and NVIDIA Corporation (Santa Clara, CA, USA). The software environment comprised Anaconda version 4.12.0, a Python 3.8 environment, along with libraries such as torch 2.3.0, sentence-transformers 2.7.0, and scipy 1.10.1. Additionally, all inference results from the LLM were obtained through the ChatGPT-4.0 API provided by OpenAI. The outputs of the LLM before and after instruction tuning were used to calculate sentence embeddings using the Hugging Face open-source pre-trained model 'all-MiniLM-L6-v2'. These embeddings were then utilized to evaluate the LLM's performance on the phase diagram comprehension task.

4.2. Benchmark and Dataset

To thoroughly evaluate the effectiveness of the prompt tuning framework for phase diagram comprehension, a structured benchmark was developed using hierarchical sampling from the golden dataset. This dataset was used as the foundation for both training and evaluating the LLM.

As described in Section 3.2.2, the dataset comprised both simple and complex phase diagram systems. Specifically, two simple systems and one complex system were randomly selected from the golden dataset to serve as domain-specific examples for prompt tuning during LLM training. After the prompt-tuning, the performance of the fine-tuned LLM was tested on 10 phase diagram systems not included in the training set. These systems form

the benchmark for assessing the model's comprehension of phase diagrams and evaluating the improvements achieved through instruction tuning.

To measure the semantic similarity between LLM's textual descriptions of phase diagrams and the expert textual descriptions in the golden data, we employed three metrics: cosine similarity, Euclidean distance, and Manhattan distance.

4.2.1. Cosine Similarity

Cosine similarity evaluates the similarity in direction between two non-zero vectors by calculating the cosine of the angle between them. For two sentence vectors $\mathbf{u}$ and $\mathbf{v}$, the cosine similarity $\cos(\theta)$ can be expressed as:

$$\text{CosineSimilarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\| \mathbf{u} \| \| \mathbf{v} \|} \tag{1}$$

where $\mathbf{u} \cdot \mathbf{v}$ is the dot product of vectors $\mathbf{u}$ and $\mathbf{v}$, and $\| \mathbf{u} \|$ and $\| \mathbf{v} \|$ are the Euclidean norms of the vectors. The range of this measure is $[-1, 1]$, with 1 indicating identical directions, -1 indicating completely opposite directions, and 0 indicating orthogonality, or no correlation.

4.2.2. Euclidean Distance

Euclidean distance, or the L2 norm, is the most intuitive distance metric, measuring the straight-line distance between two points in multi-dimensional space. For two sentence vectors $\mathbf{u}$ and $\mathbf{v}$, the Euclidean distance is defined as:

$$\text{EuclideanDistance}(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{i=1}^{n} (u_{\text{i}} - v_{\text{i}})^2} \tag{2}$$

where u_{i} and v_{i} are the components of vectors $\mathbf{u}$ and $\mathbf{v}$ in the i-th dimension. This distance metric is the square root of the sum of the squared differences between corresponding dimensions of the vectors.

4.2.3. Manhattan Distance

Manhattan distance, also known as city block distance or the L1 norm, is a geometric metric that measures the distance between two points along the axes of a standard coordinate system. For two sentence vectors $\mathbf{u}$ and $\mathbf{v}$, the Manhattan distance is defined as:

$$\text{ManhattanDistance}(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} |u_{\text{i}} - v_{\text{i}}| \tag{3}$$

where u_{i} and v_{i} represents the absolute difference between the components of vectors $\mathbf{u}$ and $\mathbf{v}$ in the i-th dimension.

4.3. Main Experimental Results

4.3.1. Performance of the LLM without Instruction Tuning

Keeping the instructions unchanged, each of the 13 phase diagrams is input into the LLM, and the textual descriptions obtained are saved in TXT file format. The LLM's responses for each phase diagram are compared with the expert textual descriptions in the golden data using cosine similarity, with the final results shown in Table 2. The first row of the table lists the names of the phase diagrams, the second row classifies the system complexity, with systems containing $\leq$3 phase transition reactions marked as simple systems (highlighted in yellow) and those with $>$3 reactions marked as complex systems (highlighted in green). The third row shows the cosine similarity metrics between the traditional methods and the expert descriptions for each phase diagram, and the fourth row shows the average metrics for complex and simple systems. The fifth row shows the cosine

similarity metrics between the LLM's generation and the expert descriptions for each phase diagram, and the sixth row shows the average metrics for complex and simple systems.

Table 2. Metrics for phase diagram comprehension task for traditional methods and LLM without instruction tuning. The table presents cosine similarity metrics for each phase diagram, along with the average metrics for two sample categories.

System	Al-Co	Cr-Ru	Cr-Ta	Cr-Ti	Fe-Ti	Hf-Re	Al-Ru	Co-Cr	Co-Fe	Cr-Fe	Cr-Ni	Cr-Re	Fe-V
types	complex	complex	complex	complex	complex	complex	simple	simple	simple	simple	simple	simple	simple
Cosine Similarity (traditional methods)	0.6502	0.8319	0.8626	0.9052	0.8341	0.8263	0.8003	0.7706	0.7868	0.8409	0.8177	0.7524	0.7576
Mean Cosine Similarity (traditional methods)	0.8184									0.7895			
Cosine Similarity (LLM's generation without prompt tuning)	0.8178	0.8255	0.9	0.8511	0.8769	0.9028	0.7644	0.8281	0.7987	0.8382	0.8758	0.8591	0.7521
Mean Cosine Similarity (LLM)	0.8624									0.8166			

Taking the Al-Co system as an example, this phase diagram is classified as complex. When using traditional methods, such as outdated reference books for phase diagram interpretation [31], the cosine similarity between the results obtained from traditional methods and the golden data is 0.6502. The average cosine similarity for traditional methods on complex systems is 0.8617, while for simple systems it is 0.8155. When the phase diagram image is input into the LLM, the cosine similarity between the LLM's output and the golden data is 0.8178. The average cosine similarity for the LLM's phase diagram comprehension tasks is 0.8617 for complex systems and 0.8155 for simple systems.

Within the simple system, the LLM performs best with the Cr-Fe phase diagram, achieving a cosine similarity of 0.8758 with the expert textual description, as shown in Figure 11. This phase diagram contains only two simple solid phase transitions and is composed of common elements, enabling the LLM to accurately identify and interpret the composition of the phase diagram and provide professional descriptions based on its materials knowledge database and strong image recognition capabilities.

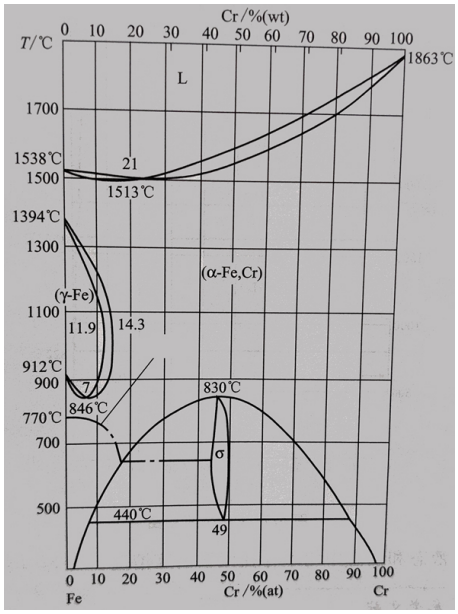
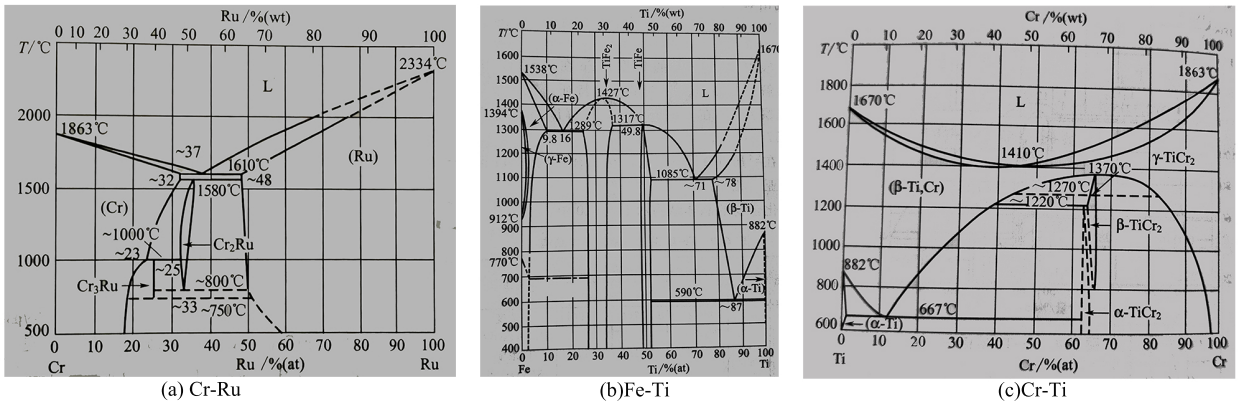


Figure 11. Phase diagram of the Cr-Fe in simple systems. Cr-Fe features two simple solid phase transitions and common elements, allowing the LLM to accurately interpret and describe it.

Among the complex systems, the LLM performs best with the Cr-Ru, Fe-Ti, and Cr-Ti systems, achieving cosine similarities with expert textual descriptions of 0.9, 0.9028, and 0.8769, respectively. As shown in Figure 12, the Cr-Ru and Cr-Ti systems each contain five phase transition reactions, while the Fe-Ti system contains four reactions, mostly common eutectic and eutectoid types of phase transitions. Overall, the LLM's performance

on complex systems phase diagram comprehension tasks is better than those on simple systems, indicating that the large model still lacks training for simple systems.



(a) Cr-Ru (b)Fe-Ti (c)Cr-Ti

Figure 12. Three phase diagrams in complex systems. Top-performing phase diagrams in complex systems: (**a**) Cr-Ru, (**b**) Fe-Ti, and (**c**) Cr-Ti, with transition reactions of 5, 5, and 4, respectively.

4.3.2. Performance of the LLM with Instruction Tuning

To enhance the LLM's capability in understanding phase diagrams and to avoid the high costs of retraining, this paper adopts a few-shot prompting engineering framework. The specific process is detailed in Section 3.2. We perform hierarchical sampling on 13 phase diagrams, extracting two samples from simple systems and one from a complex system as examples. Keeping the instructions unchanged, we compare the differences in text similarity between the LLM's output and the expert textual description before and after instruction tuning, as shown in Table 3:

Table 3. LLM performance on the phase diagram comprehension task before and after instruction tuning, showing cosine similarity metrics for each test set phase diagram.

System	Al-Co	Cr-Ru	Cr-Ta	Cr-Ti	Fe-Ti	Hf-Re	Al-Ru	Co-Cr	Co-Fe	Cr-Fe	Cr-Ni	Cr-Re	Fe-V
types	complex	complex	complex	complex	complex	complex	simple	simple	simple	simple	simple	simple	simple
Hierarchical sampling					✓				✓		✓		
Cosine Similarity (LLM's generation without prompt tuning)	0.8178	0.8255	0.9	0.8511	0.8769	0.9028	0.7644	0.8281	0.7987	0.8382	0.8758	0.8591	0.7521
Cosine Similarity (LLM's generation after prompt tuning)	0.9051	0.8746	0.884	0.8786	0.9553	0.8837	0.7926	0.8807	0.8572	0.848	0.9497	0.9017	0.8833

The first row of the table lists the phase diagram names, the second row lists the system complexity, with phase diagram containing $\leq$3 phase transition reactions marked as simple systems (highlighted in yellow), and those with >3 reactions marked as complex systems (highlighted in green). The third row indicates the samples extracted through hierarchical sampling (marked with a check), with the unmarked ones serving as the test set to evaluate the phase diagram comprehension ability of the LLM after instruction tuning. The fourth row shows the cosine similarity metrics between the LLM's phase diagram textual descriptions and expert textual descriptions before instruction tuning. Similarly, the sixth row shows the cosine similarity metrics after instruction tuning.

Taking the Al-Co system as an example, classified as the complex system, the cosine similarity between the textual description generated by the LLM and the expert textual description is 0.8178 before tuning. After prompt tuning, it improves to 0.9051.

In the simple systems, the largest improvement in metric is for the Cr-Re phase diagram, about 17.4%. As shown in Figure 4, the Cr-Re phase diagram is relatively simple, containing two peritectic reactions. The LLM, during instruction tuning, learns from two simple system examples, thus significantly improving its task performance on simple phase diagrams. Detailed analysis is presented in the ablation experiment section. In complex systems, the largest improvement in the metric is observed for the Al-Co phase diagram, about 10.7%. As shown in Figure 13, the Al-Co phase diagram is rich in information, almost marking all phase names (four compounds: Co2Al9, CoAl3, Co2Al5, CoAl

(β'), and Co4Al13; Co-based solid solutions (α-Co), (ε-Co); and the Al-based solid solution (Al)), with each phase transition reaction's temperature, composition, and involved phases detailed on the diagram. After instruction tuning, the LLM's ability to follow instructions is improved. As a result, for the Al-Co phase diagram, the trained model exhibits a significant improvement in its capacity to distill and summarize information.

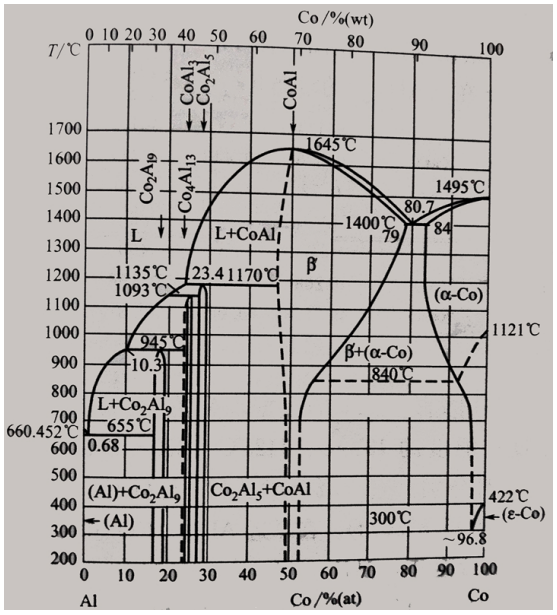


Figure 13. Al-Co phase diagram. The Al-Co phase diagram is highly detailed, labeling all phase names, including four compounds (Co2Al9, CoAl3, Co2Al5, CoAl, Co4Al13), Co-based solid solutions (α-Co, ε-Co), and Al-based solid solution (Al), along with each phase transition's temperature, composition, and involved phases.

To further quantify the contribution of the prompt engineering framework to the LLM's performance on phase diagram comprehension tasks, we analyze its contributions from a collective perspective, by examining the mean metrics for both simple and complex systems. As shown in Table 4, the first row lists two test sets—complex systems and simple systems. The second row shows the metrics of the LLM on the two test sets before instruction tuning. The third row shows the metrics after instruction tuning.

Table 4. Mean cosine similarity metrics of LLMs on phase diagram comprehension tasks before and after instruction tuning.

Types	Complex Systems	Simple Systems
Mean Cosine Similarity without prompt tuning	0.8617	0.8155
Mean Cosine Similarity after prompt tuning	0.8972	0.8721

4.4. Ablation Study

4.4.1. Effect of Sample Size Variation in Hierarchical Sampling

To assess the impact of hierarchical sampling quantity on the instruction tuning framework's effectiveness, we keep the dataset's complexity division constant: systems with over three phase transition reactions are complex; three or fewer are simple. We vary the number of hierarchical samples to see how sample quantity affects the LLM's comprehension of phase diagrams. Table 5 details this: the first row names the phase diagrams, and the second categorizes them into complex (green) and simple (yellow) based on phase transitions. The third row shows cosine similarity metrics between the LLM's responses and expert descriptions without prompt tuning. The fourth row uses hierarchical sampling to select one complex and two simple samples for prompt tuning,

with the remaining samples as the test set. Rows five and six change the sample numbers: two complex and three simple (five total), and three complex and four simple (seven total), respectively. The table presents cosine similarity metrics post-prompt tuning.

Table 5. Impact of varying hierarchical sample sizes on LLM performance in the phase diagram comprehension task.

System	Al-Co	Cr-Ru	Cr-Ta	Cr-Ti	Fe-Ti	Hf-Re	Al-Ru	Co-Cr	Co-Fe	Cr-Fe	Cr-Ni	Cr-Re	Fe-V
types	complex	complex	complex	complex	complex	complex	simple	simple	simple	simple	simple	simple	simple
Cosine Similarity (LLM's generation without prompt tuning)	0.8178	0.8255	0.9	0.8511	0.8769	0.9028	0.7644	0.8281	0.7987	0.8382	0.8758	0.8591	0.7521
Cosine Similarity (Number of hierarchical Sampling is 3)	0.9051	0.8746	0.884	0.8786	✓	0.8837	0.7926	0.8807	✓	0.848	✓	0.9017	0.8833
Cosine Similarity (Number of hierarchical Sampling is 5)	✓	0.9014	0.889	0.8912	0.8743	✓	✓	0.8702	✓	0.8274	0.8543	✓	0.8328
Cosine Similarity (Number of hierarchical Sampling is 7)	✓	0.7874	0.8963	0.8851	✓	✓	0.7806	✓	✓	0.8209	✓	✓	0.8755

For example, in the Hf-Re complex system, initial cosine similarity is 0.8255. After prompt tuning with three samples, it improves to 0.8746. With five and seven samples, similarities are 0.9014 and 0.7874, respectively. With three hierarchical samples, metrics for two of ten phase diagrams decrease, while eight improve, notably the Cr-Re system by 17.4%. The LLM initially struggles with instructions but focuses better on critical phase diagram parts post-prompt tuning, improving alignment with expert descriptions. With five samples, four out of eight metrics slightly decrease, but four improve, with Cr-Re up by 10.7% and Hf-Re by 9.2%. With seven samples, three out of six metrics decrease, while three improve, with Cr-Re again showing the highest increase at 16.4%. Overall, using three hierarchical samples (about 20% of the total) yields the best results.

4.4.2. Evaluation Metrics Comparison

Section 3.1 details the creation of golden data using the Al-Co phase diagram, processing and saving LLM outputs, and using cosine similarity to assess the LLM's performance on phase diagram comprehension. Euclidean and Manhattan distances were normalized using L2 and L1 normalization. We evaluate different similarity metrics while keeping the hierarchical sampling strategy constant, with three samples: one from complex systems and two from simple systems. These are used for prompt tuning the LLM. Performance is assessed using cosine similarity, Euclidean distance, and Manhattan distance before and after prompt tuning. Higher cosine similarity and lower Euclidean and Manhattan distances indicate better performance. The results are displayed in Table 6. The table lists phase diagram names, system complexity (complex in green, simple in yellow), and similarity metrics. Before tuning, the LLM's cosine similarity for the Al-Co phase diagram is 0.8178, improving to 0.9051 after prompt tuning. Euclidean and Manhattan distances improve from 0.6036 and 0.593 to 0.4357 and 0.4297, respectively.

Table 6. Comparison of LLM performance on the phase diagram comprehension task before and after instruction tuning, using cosine similarity, Euclidean distance, and Manhattan distance metrics.

System	Al-Co	Cr-Ru	Cr-Ta	Cr-Ti	Fe-Ti	Hf-Re	Al-Ru	Co-Cr	*Co-Fe*	Cr-Fe	Cr-Ni	Cr-Re	Fe-V
types	complex	complex	complex	complex	complex	complex	simple	simple	simple	simple	simple	simple	simple
Cosine Similarity (LLM's generation without prompt tuning)	0.8178	0.8255	0.9	0.8511	0.8769	0.9028	0.7644	0.8281	0.7987	0.8382	0.8758	0.8591	0.7521
Cosine Similarity (LLM's generation after prompt tuning)	0.9051	0.8746	0.884	0.8786	✓	0.8837	0.7926	0.8807	✓	0.848	✓	0.9017	0.8833
Euclidean Distance (LLM's generation without prompt tuning)	0.6036	0.5908	0.4471	0.5457	0.4961	0.4408	0.6865	0.5863	0.6346	0.5688	0.4985	0.5309	0.7041
Euclidean Distance (LLM's generation after prompt tuning)	0.4357	0.5007	0.4725	0.5023	✓	0.4836	0.6441	0.4885	✓	0.5515	✓	0.4434	0.483
Manhattan Distance (LLM's generation without prompt tuning)	0.593	0.595	0.4564	0.5621	0.5051	0.4428	0.6699	0.6008	0.6197	0.5755	0.4999	0.5188	0.7042
Manhattan Distance (LLM's generation after prompt tuning)	0.4297	0.5064	0.4853	0.4952	✓	0.4871	0.6342	0.483	✓	0.5518	✓	0.4357	0.4753

All three metrics quantitatively measure LLM performance trends. Post-prompt tuning, eight of ten test samples improve across all metrics, though Euclidean and Manhattan distances can be affected by noise. For instance, in the Co-Fe system, the cosine similarity increases by 1.16%, while Euclidean and Manhattan distances decrease by 3.04% and 4.12%, respectively. In the Cr-Re system, the largest improvement, cosine similarity increases by 17.44%, and Euclidean and Manhattan distances decrease by 31.4% and 32.5%. Token counts in LLM outputs increase from 306 to 377 after tuning. In high-dimensional spaces, Eu-

clidean and Manhattan distances are prone to the curse of dimensionality. Cosine similarity, unaffected by vector magnitude, better captures directional similarity in high-dimensional semantic spaces, making it more effective for comparing textual content in phase diagram comprehension tasks.

5. Conclusions and Future Works

This paper introduces a hierarchical sampling-based instruction fine-tuning framework that enables pre-trained LLMs to automatically perform phase diagram annotation tasks. This end-to-end automation process effectively addresses the shortcomings of traditional manual annotation methods in materials science, particularly in updating and batch processing. High-quality phase diagram images and their corresponding expert descriptions are collected from reference books and preprocessed, including error correction, removal of redundant information, and content supplementation. These preprocessed phase diagrams and textual descriptions are then integrated into a golden dataset. According to our design, a portion of the samples from the golden dataset is selected for LLM training. The pre-trained LLM is then used for the automated annotation of phase diagrams in the test set. Our methods significantly improve the LLM's accuracy in performing phase diagram comprehension tasks on the test set, achieving an average cosine similarity of 0.8737. Extensive experimental validation demonstrate that the pre-trained LLM's performance on phase diagram comprehension tasks improves by 7% on the test set.

Our research identifies three primary areas for future work, encompassing both broad strategies and specific technical objectives. First, we aim to develop a more versatile and general-purpose LLM based on the open-source LLaMA2 model [7]. This will be achieved by leveraging Low-Rank Adaptation (LoRA) technology [50] in conjunction with distributed training methods, particularly Fully Sharded Data Parallel (FSDP) [51]. This approach will enable efficient fine-tuning of the LLM from scratch using multiple GPUs. Initially, our focus will be on instruction fine-tuning, with a long-term goal of advancing to more sophisticated parameter fine-tuning, thereby achieving deeper model optimization. Second, we will expand and optimize the dataset for model training to improve performance on specific benchmark tasks. By increasing resources and personnel, we will enhance the golden dataset, allowing for more comprehensive and diverse data coverage. Refining the criteria for sample division will further improve categorization, making it more rational and interpretable, particularly for different system types (e.g., simple vs. complex systems). This refinement is expected to enhance the LLM's accuracy in tasks related to phase diagram analysis. Third, we aim to integrate techniques for compressing the LLM, facilitating efficient deployment across various platforms. Following compression, the model will be adaptable for use on a wide range of devices, including mobile phones and edge computing endpoints, thereby broadening accessibility and enabling researchers across disciplines to utilize the model in practical applications.

Author Contributions: Y.Z.: Conceptualization, Data Curation, Formal analysis, Investigation, Software and Writing—original draft; Y.L.: Methodology, Supervision, Validation and Writing—review & editing; X.-G.L.: Project administration, Resources and Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in [2024-AutoPhaseGPT] at [https://github.com/YangZha-2023/2024-AutoPhaseGPT] (accessed on 3 October 2024).

Acknowledgments: I would like to thank Zhenzhi Liu, Yueshan Jiang, and Genfeng Shang from our research group for their valuable expertise and guidance on phase diagrams. Their professional knowledge greatly assisted the author during the manual annotation of phase diagrams.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2023**, arXiv:2303.18223.
2. Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 39. [CrossRef]
3. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
4. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
5. Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; et al. Gemini: A family of highly capable multimodal models. *arXiv* **2023**, arXiv:2312.11805.
6. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
7. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv* **2023**, arXiv:2307.09288.
8. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-task deep neural networks for natural language understanding. *arXiv* **2019**, arXiv:1901.11504.
9. Xu, B.; Zhang, L.; Mao, Z.; Wang, Q.; Xie, H.; Zhang, Y. Curriculum learning for natural language understanding. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5 July 2020; pp. 6095–6104.
10. Dong, C.; Li, Y.; Gong, H.; Chen, M.; Li, J.; Shen, Y.; Yang, M. A survey of natural language generation. *ACM Comput. Surv.* **2022**, *55*, 173. [CrossRef]
11. Gatt, A.; Krahmer, E. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.* **2018**, *61*, 65–170. [CrossRef]
12. Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Zhong, S.; Yin, B.; Hu, X. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Trans. Knowl. Discov. Data* **2024**, *18*, 160. [CrossRef]
13. Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv* **2022**, arXiv:2205.10625.
14. He, K.; Mao, R.; Lin, Q.; Ruan, Y.; Lan, X.; Feng, M.; Cambria, E. A survey of large language models for healthcare: From data, technology, and applications to accountability and ethics. *arXiv* **2023**, arXiv:2310.05694.
15. Wan, Z.; Liu, C.; Zhang, M.; Fu, J.; Wang, B.; Cheng, S.; Ma, L.; Quilodrán-Casas, C.; Arcucci, R. Med-unic: Unifying cross-lingual medical vision-language pre-training by diminishing bias. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 56186–56197.
16. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
17. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.D.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating large language models trained on code. *arXiv* **2021**, arXiv:2107.03374.
18. Pelton, A. Metal Extraction: Phase Diagrams. In *Reference Module in Materials Science and Materials Engineering*; Elsevier: Amsterdam, The Netherlands, 2016. [CrossRef]
19. Andersson, J.O.; Helander, T.; Höglund, L.; Shi, P.; Sundman, B. Thermo-Calc & DICTRA, computational tools for materials science. *Calphad* **2002**, *26*, 273–312.
20. Cao, W.; Chen, S.L.; Zhang, F.; Wu, K.; Yang, Y.; Chang, Y.; Schmid-Fetzer, R.; Oates, W. PANDAT software with PanEngine, PanOptimizer and PanPrecipitation for multi-component phase diagram calculation and materials property simulation. *Calphad* **2009**, *33*, 328–342. [CrossRef]
21. West, D.R.F. *Ternary Phase Diagrams in Materials Science*; CRC Press: Boca Raton, FL, USA, 2017.
22. Ayer, R. Determination of Phase Diagrams by AEM. *MRS Online Proc. Libr. OPL* **1985**, *62*, 193. [CrossRef]
23. Li, L.; Zhou, H.; Yan, J.; Mandrus, D.; Keppens, V. Research update: Magnetic phase diagram of $EuTi_{1-x}B_xO_3$ (B = Zr, Nb). *APL Mater.* **2014**, *2*, 110701. [CrossRef]
24. Hidaka, H.; Yanagiya, S.; Hayasaka, E.; Kaneko, Y.; Yanagisawa, T.; Tanida, H.; Amitsuka, H. Magnetic Field–Temperature Phase Diagram of CeCoSi Constructed on the Basis of Specific Heat, Magnetoresistivity, and Magnetization Measurements: Single Crystal Study. *J. Phys. Soc. Jpn.* **2022**, *91*, 094701. [CrossRef]
25. Yoo, C.S.; Sengupta, A.; Kim, M. Phase diagram of carbon dioxide: Update and challenges. *High Press. Res.* **2011**, *31*, 68–74. [CrossRef]
26. Seidzade, A.E.; Orujlu, E.N.; Doert, T.; Amiraslanov, I.R.; Aliev, Z.S.; Babanly, M.B. An updated phase diagram of the $SnTe$-Sb_2Te_3 system and the crystal structure of the new compound $SnSb_4Te_7$. *J. Phase Equilibria Diffus.* **2021**, *42*, 373–378. [CrossRef]
27. Okamoto, H.; Okamoto, H. *Phase Diagrams for Binary Alloys*; ASM International: Materials Park, OH, USA, 2000; Volume 44.
28. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent abilities of large language models. *arXiv* **2022**, arXiv:2206.07682.
29. Daneshfar, F. Enhancing Low-Resource Sentiment Analysis: A Transfer Learning Approach. *Passer J. Basic Appl. Sci.* **2024**, *6*, 265–274. [CrossRef]
30. Islam, R.; Moushi, O.M. GPT-4o: The Cutting-Edge Advancement in Multimodal LLM. *Authorea Prepr.* **2024**, 13757.

31. Lyakishev, N.P. *Handbook of Phase Diagrams of Metallic Binary Systems*; Chemical Industry Press Co., Ltd.: Beijing, China, 2009.
32. Lejun, Z.; Minghui, P.; Shen, S.; Weizheng, W.; Zilong, J.; Yansen, S.; Huiling, C.; Ran, G.; Gataullin, S. Redundant data detection and deletion to meet privacy protection requirements in blockchain-based edge computing environment. *China Commun.* **2024**, *21*, 149–159. [CrossRef]
33. Blodgett, S.L.; Barocas, S.; Daumé, H., III; Wallach, H. Language (technology) is power: A critical survey of "bias" in nlp. *arXiv* **2020**, arXiv:2005.14050.
34. Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring massive multitask language understanding. *arXiv* **2020**, arXiv:2009.03300.
35. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 195. [CrossRef]
36. Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; Raffel, C.A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 1950–1965.
37. Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; Sui, Z. A survey on in-context learning. *arXiv* **2022**, arXiv:2301.00234.
38. Wu, Z.; Wang, Y.; Ye, J.; Kong, L. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *arXiv* **2022**, arXiv:2212.10375.
39. Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; Chen, W. What Makes Good In-Context Examples for GPT-3? *arXiv* **2021**, arXiv:2101.06804.
40. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
41. Su, H.; Kasai, J.; Wu, C.H.; Shi, W.; Wang, T.; Xin, J.; Zhang, R.; Ostendorf, M.; Zettlemoyer, L.; Smith, N.A.; et al. Selective annotation makes language models better few-shot learners. *arXiv* **2022**, arXiv:2209.01975.
42. Rubin, O.; Herzig, J.; Berant, J. Learning to retrieve prompts for in-context learning. *arXiv* **2021**, arXiv:2112.08633.
43. Li, X.; Lv, K.; Yan, H.; Lin, T.; Zhu, W.; Ni, Y.; Xie, G.; Wang, X.; Qiu, X. Unified demonstration retriever for in-context learning. *arXiv* **2023**, arXiv:2305.04320.
44. Hegde, V.I.; Aykol, M.; Kirklin, S.; Wolverton, C. The phase stability network of all inorganic materials. *Sci. Adv.* **2020**, *6*, eaay5606. [CrossRef]
45. Chipman, J. Thermodynamics and phase diagram of the Fe-C system. *Metall. Mater. Trans. B* **1972**, *3*, 55–64. [CrossRef]
46. Babanly, M.; Chulkov, E.V.; Aliev, Z.S.; Shevelkov, A.; Amiraslanov, I. Phase diagrams in materials science of topological insulators based on metal chalcogenides. *Russ. J. Inorg. Chem.* **2017**, *62*, 1703–1729. [CrossRef]
47. De Pablo, J.J.; Jackson, N.E.; Webb, M.A.; Chen, L.Q.; Moore, J.E.; Morgan, D.; Jacobs, R.; Pollock, T.; Schlom, D.G.; Toberer, E.S.; et al. New frontiers for the materials genome initiative. *NPJ Comput. Mater.* **2019**, *5*, 41. [CrossRef]
48. Zha, Y.; Liu, W.; Fan, J.; Jiang, L.; Li, Y.; Lu, X.G. Applying enhanced active learning to predict formation energy. *Comput. Mater. Sci.* **2024**, *235*, 112825. [CrossRef]
49. Hospital, A.; Goñi, J.R.; Orozco, M.; Gelpí, J.L. Molecular dynamics simulations: Advances and applications. *Adv. Appl. Bioinform. Chem.* **2015**, *8*, 37–47.
50. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
51. LLaMA, M. LLaMA Recipes. 2024. Available online: https://github.com/meta-llama/llama-recipes (accessed on 1 October 2024).