LLM4Mat-Bench: Benchmarking Large Language Models for Materials Property Prediction

Andre Niyongabo Rubungo[1,2] **Kangming Li**[3]
Jason Hattrick-Simpers[3,4,5,6] **Adji Bousso Dieng**[1,2]
[1]Department of Computer Science, Princeton University [2]Vertaix
[3]Acceleration Consortium, University of Toronto
[4]Department of Materials Science and Engineering, University of Toronto
[5]Vector Institute for Artificial Intelligence
[6]Schwartz Reisman Institute for Technology and Society
{rn3004, adji}@princeton.edu
{kangming.li, jason.hattrick.simpers}@utoronto.ca

Abstract

Large language models (LLMs) are increasingly being used in materials science. However, little attention has been given to benchmarking and standardized evaluation for LLM-based materials property prediction, which hinders progress. We present LLM4Mat-Bench, the largest benchmark to date for evaluating the performance of LLMs in predicting the properties of crystalline materials. LLM4Mat-Bench contains about 1.9M crystal structures in total, collected from 10 publicly available materials data sources, and 45 distinct properties. LLM4Mat-Bench features different input modalities: crystal composition, CIF, and crystal text description, with 4.7M, 615.5M, and 3.1B tokens in total for each modality, respectively. We use LLM4Mat-Bench to fine-tune models with different sizes, including LLM-Prop and MatBERT, and provide zero-shot and few-shot prompts to evaluate the property prediction capabilities of LLM-chat-like models, including Llama, Gemma, and Mistral. The results highlight the challenges of general-purpose LLMs in materials science and the need for task-specific predictive models and task-specific instruction-tuned LLMs in materials property prediction [1].

1 Introduction

With the remarkable success of large language models (LLMs) in solving natural language tasks [12, 31, 32, 1, 39] and different scientific tasks [26, 14, 40, 4, 15, 27], scientists have recently started to leverage LLMs to tackle very important and challenging problems in materials science, including predicting materials properties [35, 24, 44, 11, 30, 6] and discovering new materials [2, 19, 30, 6].

The learning capabilities of LLMs have the potential to revolutionize the field of materials science. For example, recent research by Rubungo et al. [35] has demonstrated the exceptional performance of LLMs in predicting the properties of crystalline materials based on textual descriptions of their structures. In their study, they introduced a novel dataset, TextEdge, which comprises textual descriptions of crystals and their corresponding properties. This dataset was used to fine-tune the encoder component of the T5-small model for the task of materials property prediction. The findings of Rubungo et al. [35] challenge the conventional practice of heavily relying on graph neural networks and using solely either crystal composition or structure as input for property prediction. Their work underscores the significance of further investigating the extent to which LLMs can be harnessed to

[1] The Benchmark and code can be found at: https://github.com/vertaix/LLM4Mat-Bench

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

develop innovative techniques for accurately predicting the properties of crystalline materials, thereby enhancing the materials discovery pipeline. Unfortunately, the proposed TextEdge dataset is limited in scope, comprising approximately 145K samples and encompassing only three distinct properties. Furthermore, its lack of diversity, being derived from a single data source (Materials Project [21]), hinders its effectiveness in assessing the robustness of LLMs in materials property prediction.

In this work, we introduce LLM4Mat-Bench, a benchmark dataset collected to evaluate the performance of LLMs in predicting the properties of crystalline materials. To the best of our knowledge, LLM4Mat-Bench is the most extensive benchmark to date for assessing the efficacy of language models in materials property prediction. The dataset comprises approximately two million samples, sourced from ten publicly available materials sources, each containing between 10K and 1M structure samples. LLM4Mat-Bench encompasses several tasks, including the prediction of electronic, elastic, and thermodynamic properties based on a material's composition, crystal information file (CIF), or textual description of its structure. We use LLM4Mat-Bench to evaluate several LLMs of different sizes, namely LLM-Prop [35] (35M parameters), MatBERT [41] (109.5M parameters), and Llama 2 [39] (7B parameters). And we provide fixed train-valid-test splits, along with carefully designed zero-shot and few-shot prompts to ensure reproducibility. We anticipate that LLM4Mat-Bench will significantly advance the application of LLMs in addressing critical challenges in materials science, including property prediction and materials discovery.

2 LLM4Mat-Bench

2.1 Data Collection Process

We collected the data used to create LLM4Mat-Bench from 10 publicly available materials data sources. In this section, we describe each data source and discuss how we accessed its data.

2.1.1 Data sources

hMOF [42] is a publicly available database[2] consisting of about 160K Metal-Organic Frameworks (MOFs), generated by Wilmer et al. using computational approaches. Materials Project (MP) [21] is a database with around 150K materials, offering free API access[3] for data retrieval, including CIF files and material properties. The Open Quantum Materials Database (OQMD) [23] is a publicly accessible database[4] of 1.2M materials, containing DFT-calculated thermodynamic and structural properties, created at Northwestern University. OMDB [3] is an organic materials database with about 12K structures and related electronic band structure properties, freely available[5]. JARVIS-DFT [8, 9] is a repository created by NIST researchers, containing around 75.9K material structures with downloadable properties[6]. QMOF [33, 34] is a quantum-chemical property database of over 16K MOFs, accessible via GitHub[7]. JARVIS-QETB [17] is a NIST-created database[8] of nearly one million materials with tight-binding parameters for 65 elements. GNoME is a database of 381K new stable materials discovered by Merchant et al. [28] using graph networks and DFT, available on GitHub[9]. Cantor HEA [25] is a DFT dataset of formation energies for 84K alloy structures, available on Zenodo[10]. SNUMAT is a database with around 10K experimentally synthesized materials and DFT properties, accessible via API[11].

2.1.2 Collecting crystal information files (CIFs) and materials property

Crystal structure files (CIFs), material compositions, and material properties were collected from publicly accessible sources described in Section 2.1.1. Data collection was facilitated by APIs and

[2] https://mof.tech.northwestern.edu/

[3] https://next-gen.materialsproject.org/api

[4] https://www.oqmd.org/

[5] https://omdb.mathub.io/

[6] https://jarvis.nist.gov/jarvisdft

[7] https://github.com/Andrew-S-Rosen/QMOF

[8] https://jarvis.nist.gov/jarvisqetb/

[9] https://github.com/google-deepmind/materials_discovery/blob/main/DATASET.md

[10] https://doi.org/10.5281/zenodo.10854500

[11] https://www.snumat.com/apis

direct download links provided by the respective databases. For databases such as Materials Project, OMDB, SNUMAT, JARVIS-DFT, and JARVIS-QETB, user registration is required for access, while databases like hMOF, QMOF, OQMD, and GNoME allow direct data access without registration. From each source, we obtained CIFs and associated material properties. Although the Materials Project and JARVIS-DFT databases offer a broader range of properties, we selected a subset—10 and 20 properties respectively—that adequately represents the data within our benchmark, based on the number of data points available for each property. This selection was made to optimize computational efficiency when training models across the 65 properties included in LLM4Mat-Bench.

2.1.3 Generating the textual description of crystal structure

LLMs perform better with textual input, and Rubungo et al. [35], Korolev and Protsenko [24], Qu et al. [30] have demonstrated that LLMs can effectively learn the structural representation of a crystal from its textual description, outperforming graph neural network (GNN)-based models that directly utilize the crystal structure for property prediction. Crystal structures are typically described in file formats such as Crystallographic Information File (CIF) which include predominantly numbers describing lattice vectors and atomic coordinates and are less amenable to LLMs. Instead of directly using these as inputs, we use Robocrystallographer [21] to deterministically generate texts that are more descriptive of crystal structures from CIF files. Robocrystallographer was developed and has been used by the Materials Project team to auto-generate texts for their database. Given a structure, Robocrystallographer leverages predefined rules and existing libraries to extract chemical and structural information, including oxidation states, global structural descriptions (symmetry information, prototype matching, structural fingerprint calculations etc.), and local structural descriptions (e.g. bonding and neighbor analysis, connectivity). This method not only generate deterministic and human-readable texts, but also ensures no data contamination in our fine-tuned LLMs, as the data sources mentioned do not include these crystal text descriptions.

Table 1: LLM4Mat-Bench statistics.

Data source	# Structure files	# Structure-Description pairs				# Tokens (Words)			# Avg. subword tokens/Sample			# Properties
		Total	Train	Validation	Test	Composition	Structure	Description	Composition	Structure	Description	
OQMD [23]	1,008,266	964,403	771,522	96,440	96,441	964K	96M	244M	5.3	635.4	347.3	2
JARVIS-QETB [17]	829,576	623,989	499,191	62,399	62,399	624K	45M	90M	3.5	466.6	202.5	4
GNoME [28]	381,000	376,276	301,020	37,628	37,628	830K	78M	508M	9.7	1185.3	1711.3	6
Materials Project [21]	146,143	146,143	125,825	10,000	10,318	272K	37M	157M	6.8	1611.8	1467.3	10
hMOF [42]	133,524	132,743	106,194	13,274	13,275	449K	96M	581M	14.9	4583.9	5629.3	7
Cantor HEA [25]	84,024	84,019	67,215	8,402	8,402	84K	11M	251M	9.5	868.4	4988.6	4
JARVIS-DFT [8, 9]	75,965	75,965	60,772	7,596	7,597	76K	9M	25M	5.0	786.0	455.9	20
QMOF [33, 34]	16,340	7,656	6,124	766	766	8K	7M	22M	14.0	5876.4	3668.0	4
OMDB [3]	12,500	12,122	9,697	1,212	1,213	66K	8M	14M	14.8	4097.4	1496.6	1
SNUMAT[12]	10,481	10,372	8,297	1,037	1,038	16K	2M	4M	5.9	1244.5	539.1	7
Total	2,697,779	1,978,985	1,592,315	193,357	193,313	4.7M	615.5M	3.1B	7.9	1559.7	1703.6	65

Table 2: Comparing the LLM4Mat-Bench with other existing benchmarks.

Benchmark	# Data Sources	# Distinct Properties	# Properties/# Samples			# Properties/Task Type		Material Representations		
			<10k	10-100k	100k+	Regression	Classification	Composition	Structure	Description
MatBench [13]	6	10	7	3	3	10	3	✓	✓	✗
TextEdge [35]	1	3	0	0	3	2	1	✗	✗	✓
LLM4Mat-Bench (Ours)	10	45	5	31	29	60	5	✓	✓	✓

2.2 Data Statistics

As Table 1 shows, LLM4Mat-Bench comprises 2,697,779 structure files, which, after pairing with descriptions generated by Robocrystallographer and filtering out descriptions with fewer than five words, result in 1,978,985 composition-structure-description pairs[13]. The reduction in sample count is also due to Robocrystallographer's inability to describe certain CIF files. The total samples for each dataset in LLM4Mat-Bench are randomly split into 80%, 10%, and 10% for training, validation, and testing, respectively. OQMD has the highest number of samples at 964,403, while QMOF has the fewest with 7,656 samples. On average, each dataset in LLM4Mat-Bench contains approximately 200,000 samples.

In LLM4Mat-Bench, when combined, textual descriptions contain 3.1 billion tokens, crystal structures 615 million, and compositions 4.7 million[14]. OQMD leads in composition tokens (964K), while

[13] The total number of pairs were 2,433,688, after removing about 454703 duplicated pairs across datasets, it resulted to 1,978,985 pairs.

[14] We used NLTK toolkit as a tokenizer to count the number of words/tokens.

hMOF has the most description tokens (581M). For CIFs, both OQMD and hMOF have around 96M tokens. On average, compositions have 8 subword tokens per sample, CIFs 1600, and descriptions 1700. hMOF averages the longest inputs for compositions (14.9) and descriptions (5629), while QMOF leads in structures (5876.4)[15]. JARVIS-DFT has the most tasks with 20 properties, followed by Materials Project with 10, and OMDB with one. Details on sample counts are in Section 3.2.

LLM4Mat-Bench provides the most comprehensive dataset compared to existing benchmarks, with the largest number of samples, properties, and tasks, including 60 regression and 5 classification tasks (see Table 2). It also offers more diverse material representations, incorporating chemical formulas, crystal structures, and crystal text descriptions. In contrast, MatBench [13] and TextEdge [35] have fewer tasks and less representation diversity, with MatBench lacking crystal text descriptions and TextEdge missing material compositions and crystal structures.

2.3 Data Quality

Since Robocrystallographer generates crystal text descriptions in a deterministic manner following predefined and well-validated rules [21], these texts should faithfully describe the crystal structures used to generate them. Regarding the quality of labels, they are calculated from simulations and are usually considered noise-free. Properties data except those from JARVIS-QETB and hMOF are obtained from DFT, which is based on fundamental quantum mechanical equations. While DFT calculations can still be performed with different levels of approximations and fidelity, the DFT-calculated properties are usually considered to be highly reliable and are routinely used as noise-free ground truths for ML models in the materials science community.

3 Results

3.1 Experimental Details

We conducted about 845 experiments, evaluating the performance of five models and three material representations on each property for each data source. Consistent with standard practices in materials science, we evaluated performance separately for each data source rather than combining samples from different sources for the same property. This approach accounts for variations in techniques and settings used by different data sources, which can result in discrepancies, such as differing band gaps for the same material. Below, we will describe each material representation, model, and metric that we used to conduct our experiments.

3.1.1 Material Representations

LLM4Mat-Bench includes three distinct materials representations: Composition, CIF, and Description (see Table 6). The primary goal of using these diverse representations is to identify which best enhances LLM performance in predicting material properties across different data sources.

Composition (Comp.) Material composition refers to the chemical formula of a material. Though it only provides stoichiometric information, studies have shown it can still be a reliable material representation for property prediction [13, 38]. For LLMs, it offers the advantage of being a short sequence that usually fits within the model's context window, making it efficient to train. To further optimize efficiency, we set the longest sequence of material compositions from each data source as the context window, rather than using the default 512 tokens for fine-tuning while the original length is kept during inference.

CIF We represent the materials structure using CIF files, the conventional way of representing the crystal structure in crystallography [20]. CIFs are commonly used for GNN-based models, but some recent works have demonstrated that it can also work with LLMs [2, 16, 19].

Description (Descr.) As we outlined in Section 2.1.3, we also use textual descriptions of crystal structures as representations for both atomic crystals and MOFs.

[15] We used Llama 2 tokenizer to count the number of subword tokens.

3.1.2 Models

We benchmarked different LLM-based models with various sizes, and a GNN-based baseline. Herein, We provide the details of each model.

CGCNN [43] is employed as a GNN baseline which is widely used in the materials science community[16]. We trained on LLM4Mat-Bench from scratch with optimal hyperparameters: 128 hidden dimensions, batch size of 256, three message passing layers, 1e-2 learning rate, 8.0 radius cutoff, 12 nearest neighbors, and 500 training epochs, though extending to 1000 epochs improved performance in some cases.

MatBERT [41] is a BERT-base model [12] with 109 million parameters, pretrained on two million materials science articles. We fine-tuned MatBERT on LLM4Mat-Bench, following Rubungo et al. [35], and achieved optimal performance with a 512-token input length, 64-sample batch size, 5e-5 learning rate, 0.5 dropout, and 100 epochs using the Adam optimizer and onecycle learning rate scheduler [36]. Although training for 200 epochs improves performance, results are reported for 100 epochs due to computational constraints.

LLM-Prop is a model based on the encoder part of T5-small model [32] introduced by Rubungo et al. [35], with 35 million parameters, smaller than MatBERT. It predicts material properties from the textual descriptions of crystal structures. To adapt LLM-Prop on CIF, we employed xVal encoding [18], where we parse an input sequence x to extract numerical values into a list x_{num}, replace them with a [NUM] token to form x_{text}, and then embed x_{text}, followed by multiplying each [NUM] embedding by its corresponding value in x_{num} to get h_{embed} that we feed to the model. xVal encoding ensures that the quantitative value of each number is reflected in the input embedding while reducing the input length caused by the high volume of numerical values in CIF files, which extend the length of the input sequence after tokenization. We fine-tuned LLM-Prop on LLM4Mat-Bench and optimizing with a 1e-3 learning rate, 0.2 dropout, Adam optimizer, and onecycle learning rate scheduler for 100 epochs, with a 768-token input length, batch size of 64 for training, and 512 for inference. While Rubungo et al. [35] recommended that training for 200 epochs and increasing the number of input tokens improves the performance, we could not replicate this due to computational constraints.

Table 3: Prompt template. <material representation type> denotes *"chemical formula"*, *"cif structure"*, or *"structure description"*. <value> represents the input context (for example *NaCl*, etc.). <property name> denotes the name of the property (for example *band gap*, etc.). <predicted value> represents the property value generated by Llama 2 while <actual value_i> represents the ground truth of the EXAMPLE_i. FINAL PROMPT and **RESPONSE** denote the input prompt to Llama 2 and its generated output, respectively.

Prompt Type	Template
-	SYSTEM PROMPT: «SYS» You are a material scientist. Look at the <material representation type> of the given crystalline material and predict its property. The output must be in a json format. For example: {property_name:predicted_property_value}. Answer as precise as possible and in as few words as possible. «/SYS» INPUT PROMPT: <material representation type>: <value> property name: <property name>.
0-shot	FINAL PROMPT: <s>[INST] + SYSTEM PROMPT + INPUT PROMPT + [/INST] RESPONSE: <property name>:<predicted value>
5-shot	EXAMPLE_1: <material representation type>: <value_1> property name: <property name>. <property name>:<actual value_1> ... EXAMPLE_5: <material representation type>: <value_5> property name: <property name>. <property name>:<actual value_5> FINAL PROMPT: <s>[INST] + SYSTEM PROMPT + EXAMPLE_1 + ... + EXAMPLE_5 + INPUT PROMPT + [/INST] RESPONSE: <property name>:<predicted value>

Llama 2-7b-chat To assess the performance of conversational LLMs in materials property prediction, we tested Llama 2-7b-chat (7 billion parameters) using our designed zero-shot and five-shot prompts (see Table 3) without fine-tuning. For the CIF structure prompts, we removed "*# generated using*

[16] Although CGCNN is not state-of-the-art for some properties, it was faster compared to models like ALIGNN [7] and DeeperGatGNN [29], making it suitable for our extensive experiments

pymatgen" comment that is appended to each file. The maximum input length was set to 4000 tokens while the output length was set to 256, with a batch size of 256 samples, temperature of 0.8, and top-K sampling applied with $K = 10$. The details of other models that we compared with Llama 2 can be found in Appendix C. For five-shot examples, we sampled from crystals with shorter structures and descriptions to reduce the context length. We also made sure the property values for those examples are diverse (for instance, they should not all have 0.0 eV as their bandgap values).

We trained all models using NVIDIA RTX A6000 GPUs. Training MatBERT with two GPUs on about 300K data points and 100 epochs took about four days while for LLM-Prop took about 2.5 days. For CGCNN, it took about 7 hours training time on one GPU for 500 epochs. With one GPU, Llama 2 took about a half day to generate the output of 40K samples with 256 tokens maximum length each. We report the test set results averaged over five runs for predictive models and three runs for generative models.

3.1.3 Evaluation Metrics

Following Choudhary and DeCost [7], we evaluated regression tasks using the ratio between the mean absolute deviation (MAD) of the ground truth and the mean absolute error (MAE) of the predicted properties. The MAD:MAE ratio ensures an unbiased model comparison between different properties where the higher ratio the better. According to Choudhary and DeCost [7], a good predictive model should have at least 5.0 ratio. MAD values represent the performance of a random guessing model predicting the average value for each data point. To provide a comprehensive performance comparison across datasets, we also reported the weighted average of MAD:MAE across all properties in each dataset (Wtd. Avg. (MAD:MAE), see Equation 1).

For classification tasks, we reported the area under the ROC curve (AUC) for each task and provided the weighted average across all properties (Wtd. Avg. AUC, see Equation 2).

$$\text{Wtd. Avg. (MAD/MAE)} = \frac{\sum_{\mathrm{i}}^{m} \mathrm{TestSize_i} \times \frac{\mathrm{MAD_i}}{\mathrm{MAE_i}}}{\sum_{\mathrm{i}}^{m} \mathrm{TestSize_i}} \tag{1}$$

$$\text{Wtd. Avg. AUC} = \frac{\sum_{\mathrm{i}}^{m} \mathrm{TestSize_i} \times \mathrm{AUC_i}}{\sum_{\mathrm{i}}^{m} \mathrm{TestSize_i}}, \tag{2}$$

m denotes the number of regression properties in the dataset.

Table 4: The Wtd. Avg. (MAD:MAE) scores (the higher the better) for the regression tasks in the LLM4Mat-Bench are reported. **Bolded** results indicate the best model for each input format, while **bolded results with blue background** highlight the best model per each dataset. Inval. denotes cases where the Llama model failed to generate outputs with a property value or had fewer than 10 valid predictions.

Input	Model	Dataset									
		MP 8 tasks	**JARVIS-DFT** 20 tasks	**GNoME** 6 tasks	**hMOF** 7 tasks	**Cantor HEA** 4 tasks	**JARVIS-QETB** 4 tasks	**OQMD** 2 tasks	**QMOF** 4 tasks	**SNUMAT** 4 tasks	**OMDB** 1 task
CIF	CGCNN (baseline)	5.319	7.048	19.478	2.257	17.780	61.729	14.496	3.076	1.973	2.751
Comp.	Llama 2-7b-chat:0S	0.389	Inval.	0.164	0.174	0.034	0.188	0.105	0.303	0.940	0.885
	Llama 2-7b-chat:5S	0.627	0.704	0.499	0.655	0.867	1.047	1.160	0.932	1.157	1.009
	MatBERT-109M	**5.317**	**4.103**	12.834	1.430	6.769	11.952	5.772	**2.049**	**1.828**	**1.554**
	LLM-Prop-35M	4.394	2.912	**15.599**	**1.479**	**8.400**	**59.443**	**6.020**	1.958	1.509	1.507
CIF	Llama 2-7b-chat:0S	0.392	0.216	6.746	0.214	0.022	0.278	0.028	0.119	0.682	0.159
	Llama 2-7b-chat:5S	Inval.	Inval.	Inval.	Inval.	Inval.	1.152	1.391	Inval.	Inval.	0.930
	MatBERT-109M	7.452	6.211	14.227	1.514	9.958	47.687	10.521	3.024	**2.131**	**1.777**
	LLM-Prop-35M	**8.554**	**6.756**	**16.032**	**1.623**	**15.728**	**97.919**	**11.041**	**3.076**	1.829	**1.777**
Descr.	Llama 2-7b-chat:0S	0.437	0.247	0.336	0.193	0.069	0.264	0.106	0.152	0.883	0.155
	Llama 2-7b-chat:5S	0.635	0.703	0.470	0.653	0.820	0.980	1.230	0.946	1.040	1.001
	MatBERT-109M	7.651	6.083	15.558	1.558	9.976	46.586	**11.027**	**3.055**	**2.152**	**1.847**
	LLM-Prop-35M	**9.116**	**7.204**	**16.224**	**1.706**	**15.926**	**93.001**	9.995	3.016	1.950	1.656

3.2 Discussion

Table 4 and 5, and Figure 1 and 2 show the main results. The detailed results on each dataset can be found in Appendix E. The main observations are as follows:

Table 5: The Wtd. Avg. AUC scores (the higher the better) for the classification tasks in the LLM4Mat-Bench.

Input	Model	Dataset	
		MP 2 tasks	SNUMAT 3 tasks
CIF	CGCNN (baseline)	0.846	0.722
Comp.	Llama 2-7b-chat:0S	0.491	Inval.
	Llama 2-7b-chat:5S	0.507	0.466
	MatBERT-109M	**0.722**	0.712
	LLM-Prop-35M	0.691	**0.716**
CIF	Llama 2-7b-chat:0S	0.501	0.489
	Llama 2-7b-chat:5S	0.502	0.474
	MatBERT-109M	**0.750**	**0.717**
	LLM-Prop-35M	0.738	0.660
Descr.	Llama 2-7b-chat:0S	0.500	Inval.
	Llama 2-7b-chat:5S	0.502	0.568
	MatBERT-109M	0.735	0.730
	LLM-Prop-35M	**0.742**	**0.735**

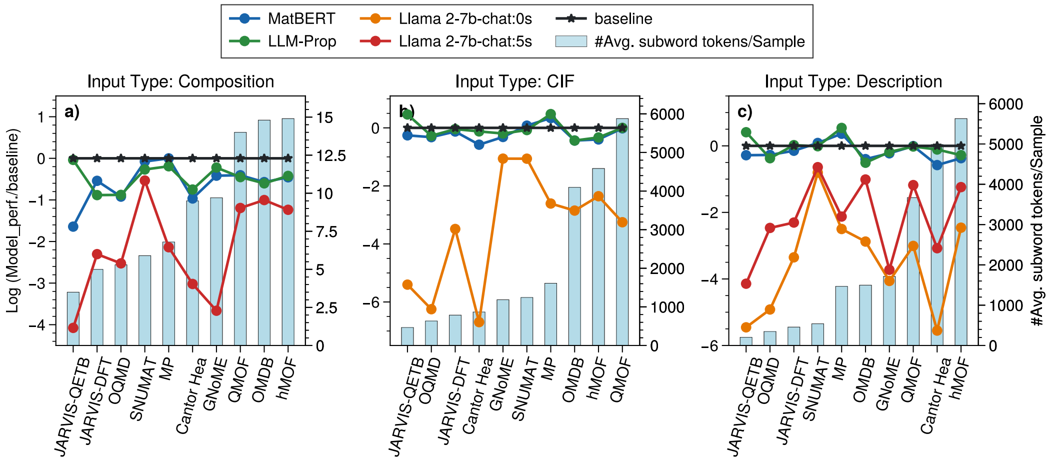


Figure 1: The performance comparison across models for each material representation is presented. The left y-axis shows the log-normalized performance of each LLM-based model relative to the baseline (CGCNN), while the right y-axis (bar plots) displays the average subword tokens per sample for each dataset. Datasets on the x-axis are ordered by increasing average subword tokens. Results for Llama 2-7b-chat:0S and Llama 2-7b-chat:5S are missing in plots (a) and (b), respectively, due to invalid outputs. Higher values in the line plots indicate better performance.

Small, task-specific predictive LLMs exhibit significantly better performance than larger, generative general-purpose LLMs. This performance disparity is evident across both regression (Table 4 and Figure 1) and classification tasks (Table 5) on all 10 datasets. Specifically, LLM-Prop and MatBERT outperform Llama 2-7b-chat:0S and Llama 2-7b-chat:5S by a substantial margin, despite being approximately 200 and 64 times smaller in size, respectively. In regression tasks, LLM-Prop achieves the highest accuracy on 8 out of 10 datasets, with MatBERT leading on the remaining 2 datasets. For classification tasks, both LLM-Prop and MatBERT deliver the best performance on 1 out of 2 datasets. LLM-Prop surpasses MatBERT by 1.8% on the SNUMAT dataset, whereas MatBERT outperforms LLM-Prop by 0.8% on the other dataset. As anticipated, a modest enhancement in average performance is observed across various datasets and input formats when the Llama 2-7b-chat

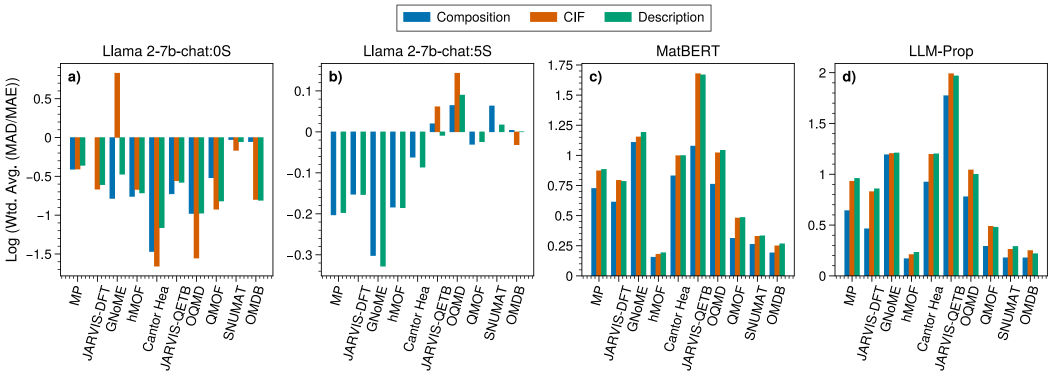


Figure 2: The performance comparison across material representations for each LLM-based model is shown. The y-axis represents the log-normalized Weighted Average (MAD/MAE) score for each representation, while the x-axis displays randomly ordered datasets. In the (a) and (b) plots, some Composition and Structure performance results are missing due to invalid outputs. A higher y-axis value indicates better performance.

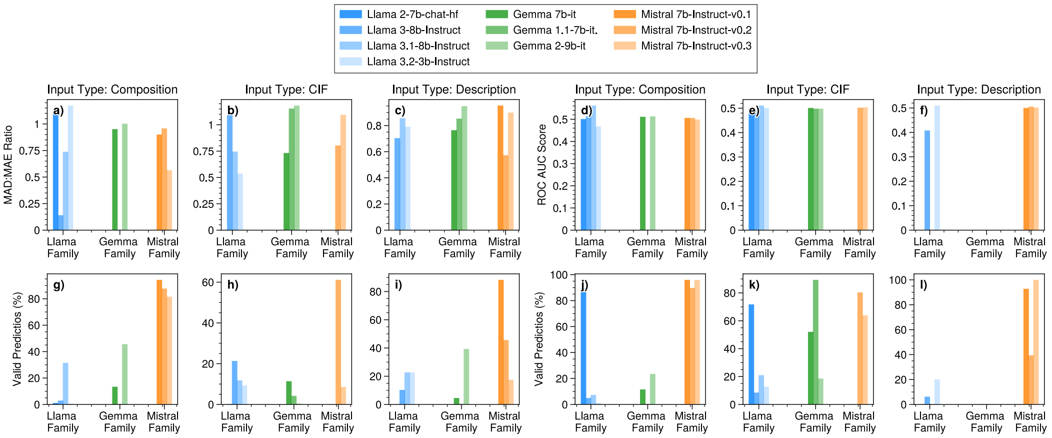


Figure 3: The performance comparison of different chat-based LLM versions is presented with results based on 5-shot prompts, averaged over three inference runs. Panels (a)–(c) and (d)–(f) show each model's accuracy in predicting band gaps and stability in the MP dataset, respectively, while panels (g)–(i) and (j)–(l) depict the percentage of valid predictions for band gap and stability on the test set.

model is evaluated using 5-shot prompts rather than 0-shot prompts. Determining the optimal number of examples required to achieve peak performance will be the focus of future work.

General-purpose generative LLMs hallucinate and often fail to generate valid property values. As shown in Table 4, Table 5, Figure 2, and Appendix E, Llama 2-7b-chat model produces invalid outputs on multiple tasks, where the expected property value is missing. This issue occurs less frequently when the input is a description or chemical formula, but more commonly when the input is a CIF file. One reason may be that descriptions and chemical formulas resemble natural language, which LLMs can more easily interpret compared to CIF files. This may also indicate that when the input modality during inference differs significantly from the modalities encountered during pretraining, fine-tuning is necessary to achieve reasonable performance. Another key observation is that Llama 2-7b-chat model often generates the same property value for different inputs (i.e. hallucinates), contributing to its poor performance across multiple tasks. These findings highlight the importance of caution when using general-purpose generative LLMs for materials property prediction and emphasize the need for fine-tuned, task-specific LLM-based models.

Representing materials with their textual descriptions improves the performance of LLM-based property predictors compared to other representations. We observe a significant performance improvement when the input is a description compared to when it is a CIF file or a chemical formula. One of the possible reasons for this might be that LLMs are more adept at learning from natural language data. On the other hand, although material compositions appear more natural to LLMs compared to CIF files, they lack sufficient structural information. This is likely why LLMs with CIF files as input significantly outperform those using chemical formulas.

More advanced, general-purpose generative LLMs do not necessarily yield better results in predicting material properties. In Figure 3, we compare the performance of Llama 2-7b-chat-hf model with advanced versions of Llama of comparable sizes when predicting material's band gap and its stability. Similar comparisons are also conducted for the Mistral [22] and Gemma [37] models. The results indicate that, despite being trained on substantially larger and higher-quality datasets, more advanced versions of generative LLMs show limited improvements in performance and validity of predictions for material properties. For instance, Llama 3 and 3.1 8b models were trained on over 15 trillion tokens—around eight times more data than the 2 trillion tokens used for the Llama 2 7b models. This finding highlights the ongoing challenges of leveraging LLMs in material property prediction and underscores the need for further research to harness the potential of these robust models in this domain.

The performance on energetic properties is consistently better across all datasets compared to other properties. This is consistent with the trend observed in the community benchmarks such as MatBench and JARVIS-Leaderboard, where energetic properties are among those that can be most accurately predicted [13, 10]. This is not surprising because energy is known to be relatively well predicted from e.g., compositions and atom coordination (bonding), which is inherently represented in GNNs and also presented in text descriptions.

Task-specific predictive LLM-based models excel with shorter textual descriptions, while CGCNN performs better on datasets with longer descriptions. While the focus on this work is on LLMs, a comparison with a simple but widely used GNN-based baseline suggests room for improvement in LLM-based property prediction. For regression tasks, LLM-Prop outperforms CGCNN on only 4 out of 10 datasets (MP, JARVIS-DFT, JARVIS-QETB, and SNUMAT), and MatBERT outperforms CGCNN on just 2 out of 10 datasets (MP and JARVIS-QETB). In contrast, CGCNN achieves the best performance on 5 out of 10 datasets (GNoME, hMOF, Cantor HEA, OQMD, and OMDB). Further analysis reveals that CGCNN tends to perform better than LLM-based models on datasets with relatively longer textual descriptions, while LLM-based models excel on datasets with shorter descriptions (see Table 1). The performance gain on shorter descriptions may stem from LLM-based models' ability to leverage more context from compact text, while CGCNN consistently benefits from training on the entire crystal structure.

4 Conclusion

LLMs are increasingly being utilized in materials science, particularly for materials property prediction and discovery. However, the absence of standardized evaluation benchmarks has impeded progress in this field. We introduced LLM4Mat-Bench, a comprehensive benchmark dataset designed to evaluate LLMs for predicting properties of atomic and molecular crystals and MOFs. Our results demonstrate the limitations of general-purpose LLMs in this domain and underscore the necessity for task-specific predictive models and instruction-tuned LLMs tailored for materials property prediction. These findings emphasize the importance of using LLM4Mat-Bench to advance the development of more effective LLMs in materials science.

5 Limitations

Due to computational constraints and the number of experiments, we were unable to conduct thorough hyperparameter searches for each property and dataset. The reported settings were optimized on the MP dataset and then fixed for other datasets. For each model, we highlighted hyperparameter settings that may improve performance (see Section 3.1.2). Additionally, we could not include results from SOTA commercial LLMs such as GPT-4o[17] or Claude 3.5 Sonnet[18] due to budget constraints.

[17] https://openai.com/index/hello-gpt-4o/

[18] https://www.anthropic.com/news/claude-3-5-sonnet

We also encountered issues with chat-based models, which sometimes failed to follow the output format, producing invalid or incomplete outputs. Extracting property values was therefore challenging. We believe further instruction-tuning chat-based models on the provided prompts could mitigate these issues.

Furthermore, we did not include comparisons with dataset-specific retrieval-augmented generation (RAG) models, such as the recently developed LLaMP [5], a RAG-based model tailored for interaction with the MP dataset. Our work aims to provide a comprehensive benchmark and baseline results to advance the evaluation of LLM-based methods for materials property prediction. Future work should address these limitations.

Acknowledgements

Adji Bousso Dieng acknowledges support from the National Science Foundation, Office of Advanced Cyberinfrastructure (OAC) #2118201, and from the Schmidt Sciences AI2050 Early Career Fellowship.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] L. M. Antunes, K. T. Butler, and R. Grau-Crespo. Crystal structure generation with autoregressive large language modeling. *arXiv preprint arXiv:2307.04340*, 2023.

[3] S. S. Borysov, R. M. Geilhufe, and A. V. Balatsky. Organic materials database: An open-access online database for data mining. *PloS one*, 12(2):e0171501, 2017.

[4] C. M. Castro Nascimento and A. S. Pimentel. Do large language models understand chemistry? a conversation with chatgpt. *Journal of Chemical Information and Modeling*, 63(6):1649–1655, 2023.

[5] Y. Chiang, C.-H. Chou, and J. Riebesell. Llamp: Large language model made powerful for high-fidelity materials knowledge retrieval and distillation. *arXiv preprint arXiv:2401.17244*, 2024.

[6] K. Choudhary. Atomgpt: Atomistic generative pretrained transformer for forward and inverse materials design. *The Journal of Physical Chemistry Letters*, 15(27):6909–6917, 2024.

[7] K. Choudhary and B. DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):1–8, 2021.

[8] K. Choudhary, I. Kalish, R. Beams, and F. Tavazza. High-throughput identification and characterization of two-dimensional materials using density functional theory. *Scientific reports*, 7(1):5179, 2017.

[9] K. Choudhary, Q. Zhang, A. C. Reid, S. Chowdhury, N. Van Nguyen, Z. Trautt, M. W. Newrock, F. Y. Congo, and F. Tavazza. Computational screening of high-performance optoelectronic materials using optb88vdw and tb-mbj formalisms. *Scientific data*, 5(1):1–12, 2018.

[10] K. Choudhary, D. Wines, K. Li, K. F. Garrity, V. Gupta, A. H. Romero, J. T. Krogel, K. Saritas, A. Fuhr, P. Ganesh, et al. Jarvis-leaderboard: a large scale benchmark of materials design methods. *npj Computational Materials*, 10(1):93, 2024.

[11] K. Das, P. Goyal, S.-C. Lee, S. Bhattacharjee, and N. Ganguly. Crysmmnet: multimodal representation for crystal property prediction. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR, 2023.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):138, 2020.

[14] C. Edwards, T. Lai, K. Ros, G. Honke, K. Cho, and H. Ji. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, 2022.

[15] Y. Fang, X. Liang, N. Zhang, K. Liu, R. Huang, Z. Chen, X. Fan, and H. Chen. Mol-instructions-a large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[16] D. Flam-Shepherd and A. Aspuru-Guzik. Language models can generate molecules, materials, and protein binding sites directly in three dimensions as xyz, cif, and pdb files. *arXiv preprint arXiv:2305.05708*, 2023.

[17] K. F. Garrity and K. Choudhary. Fast and accurate prediction of material properties with three-body tight-binding model for the periodic table. *Physical review materials*, 7(4):044603, 2023.

[18] S. Golkar, M. Pettee, M. Eickenberg, A. Bietti, M. Cranmer, G. Krawezik, F. Lanusse, M. McCabe, R. Ohana, L. H. Parker, et al. xval: A continuous number encoding for large language models. In *NeurIPS 2023 AI for Science Workshop*.

[19] N. Gruver, A. Sriram, A. Madotto, A. G. Wilson, C. L. Zitnick, and Z. W. Ulissi. Fine-tuned language models generate stable inorganic materials as text. In *The Twelfth International Conference on Learning Representations*, 2023.

[20] S. R. Hall, F. H. Allen, and I. D. Brown. The crystallographic information file (cif): a new standard archive file for crystallography. *Foundations of Crystallography*, 47(6):655–685, 1991.

[21] A. Jain, S. Ong, G. Hautier, W. Chen, W. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, et al. The materials project: A materials genome approach to accelerating materials innovation. apl materials, 1 (1): 011002, 2013, 2013.

[22] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[23] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton. The open quantum materials database (oqmd): assessing the accuracy of dft formation energies. *npj Computational Materials*, 1(1):1–15, 2015.

[24] V. Korolev and P. Protsenko. Accurate, interpretable predictions of materials properties within transformer language models. *Patterns*, 4(10), 2023.

[25] K. Li, K. Choudhary, B. DeCost, M. Greenwood, and J. Hattrick-Simpers. Efficient first principles based modeling via machine learning: from simple representations to high entropy materials. *Journal of Materials Chemistry A*, 12(21):12412–12422, 2024.

[26] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

[27] L. Lv, Z. Lin, H. Li, Y. Liu, J. Cui, C. Y.-C. Chen, L. Yuan, and Y. Tian. Prollama: A protein large language model for multi-task protein language processing. *arXiv preprint arXiv:2402.16445*, 2024.

[28] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon, and E. D. Cubuk. Scaling deep learning for materials discovery. *Nature*, 2023. doi: 10.1038/s41586-023-06735-9.

[29] S. S. Omee, S.-Y. Louis, N. Fu, L. Wei, S. Dey, R. Dong, Q. Li, and J. Hu. Scalable deeper graph neural networks for high-performance materials property prediction. *Patterns*, 3(5), 2022.

[30] J. Qu, Y. R. Xie, K. M. Ciesielski, C. E. Porter, E. S. Toberer, and E. Ertekin. Leveraging language representation for materials exploration and discovery. *npj Computational Materials*, 10(1):58, 2024.

[31] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[32] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[33] A. S. Rosen, S. M. Iyer, D. Ray, Z. Yao, A. Aspuru-Guzik, L. Gagliardi, J. M. Notestein, and R. Q. Snurr. Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter*, 4(5):1578–1597, 2021.

[34] A. S. Rosen, V. Fung, P. Huck, C. T. O'Donnell, M. K. Horton, D. G. Truhlar, K. A. Persson, J. M. Notestein, and R. Q. Snurr. High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration. *npj Computational Materials*, 8(1):112, 2022.

[35] A. N. Rubungo, C. Arnold, B. P. Rand, and A. B. Dieng. Llm-prop: Predicting physical and electronic properties of crystalline solids from their text descriptions. *arXiv preprint arXiv:2310.14029*, 2023.

[36] L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[37] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[38] S. I. P. Tian, A. Walsh, Z. Ren, Q. Li, and T. Buonassisi. What information is necessary and sufficient to predict materials properties using machine learning? *arXiv preprint arXiv:2206.04968*, 2022.

[39] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[40] G. Valentini, D. Malchiodi, J. Gliozzo, M. Mesiti, M. Soto-Gomez, A. Cabri, J. Reese, E. Casiraghi, and P. N. Robinson. The promises of large language models for protein design and modeling. *Frontiers in Bioinformatics*, 3, 2023.

[41] N. Walker, A. Trewartha, H. Huo, S. Lee, K. Cruse, J. Dagdelen, A. Dunn, K. Persson, G. Ceder, and A. Jain. The impact of domain-specific pre-training on named entity recognition tasks in materials science. *Available at SSRN 3950755*, 2021.

[42] C. E. Wilmer, M. Leaf, C. Y. Lee, O. K. Farha, B. G. Hauser, J. T. Hupp, and R. Q. Snurr. Large-scale screening of hypothetical metal–organic frameworks. *Nature chemistry*, 4(2):83–89, 2012.

[43] T. Xie and J. C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018.

[44] T. Xie, Y. Wan, W. Huang, Z. Yin, Y. Liu, S. Wang, Q. Linghu, C. Kit, C. Grazian, W. Zhang, et al. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv:2308.13565*, 2023.

Appendices

A Materials Representations

Table 6: LLM4Mat-Bench material representations of Sodium Chloride (NaCl).

Crystal Information File (CIF)

```
# generated using pymatgen
data_NaCl
_symmetry_space_group_name_H-M   ’P 1’
_cell_length_a   3.50219000
_cell_length_b   3.50219000
_cell_length_c   3.50219000
_cell_angle_alpha   90.00000000
_cell_angle_beta   90.00000000
_cell_angle_gamma   90.00000000
_symmetry_Int_Tables_number   1
_chemical_formula_structural   NaCl
_chemical_formula_sum   ’Na1 Cl1’
_cell_volume   42.95553287
_cell_formula_units_Z   1
loop_
 _symmetry_equiv_pos_site_id
 _symmetry_equiv_pos_as_xyz
  1  ’x, y, z’
loop_
 _atom_type_symbol
 _atom_type_oxidation_number
  Na+  1.0
  Cl-  -1.0
loop_
 _atom_site_type_symbol
 _atom_site_label
 _atom_site_symmetry_multiplicity
 _atom_site_fract_x
 _atom_site_fract_y
 _atom_site_fract_z
 _atom_site_occupancy
  Na+  Na0  1  0.00000000  0.00000000  0.00000000  1
  Cl-  Cl1  1  0.50000000  0.50000000  0.50000000  1
```

Description

NaCl is Tetraauricupride structured and crystallizes in the cubic $P\overline{m}3m$ space group. Na^{1+} is bonded in a body-centered cubic geometry to eight equivalent Cl^{1-} atoms. All Na-Cl bond lengths are 3.03 Å. Cl^{1-} is bonded in a body-centered cubic geometry to eight equivalent Na^{1+} atoms.

B Statistics of All Properties in LLM4Mat-Bench

Table 7: Statistics of all datasets in LLM4Mat-Bench. It is important to note that we retain the naming convention of each property from the original data source with the intent to provide the distribution of properties in each dataset.

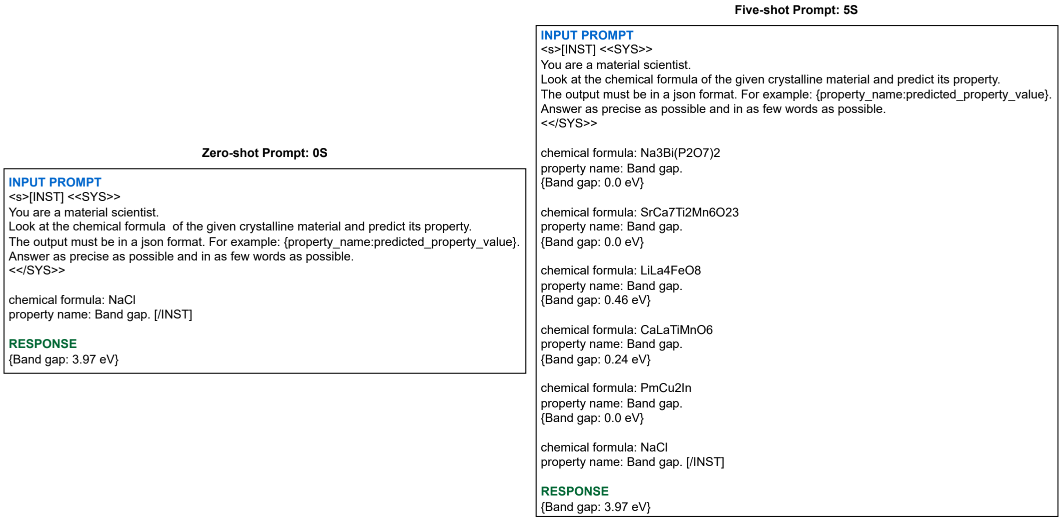
Property	Task type	# Samples/Data source									
		JARVIS-DFT	Materials Project	SNUMAT	hMOF	GNoME	JARVIS-QETB	Cantor HEA	QMOF	OQMD	OMDB
Bandgap	Regression	-	145,302	-	-	288,209	-	-	16,340	1,007,324	12,500
Bandgap (OPT)	Regression	75,965	-	-	-	-	-	-	-	-	-
Bandgap (MBJ)	Regression	19,800	-	-	-	-	-	-	-	-	-
Bandgap GGA	Regression	-	-	10,481	-	-	-	-	-	-	-
Bandgap HSE	Regression	-	-	10,481	-	-	-	-	-	-	-
Bandgap GGA Optical	Regression	-	-	10,481	-	-	-	-	-	-	-
Bandgap HSE Optical	Regression	-	-	10,481	-	-	-	-	-	-	-
Indirect Bandgap	Regression	-	-	-	-	-	829,576	-	-	-	-
Formation Energy Per Atom (FEPA)	Regression	75,965	145,262	-	-	384,871	829,576	84,024	-	1,008,266	-
Energy Per Atom (EPA)	Regression	-	145,262	-	-	-	829,576	84,024	-	-	-
Decomposition Energy Per Atom (DEPA)	Regression	-	-	-	-	384,871	-	-	-	-	-
Energy Above Hull (Ehull)	Regression	75,965	145,262	-	-	-	-	84,024	-	-	-
Total Energy	Regression	75,965	-	-	-	384,871	829,576	-	16,340	-	-
Efermi	Regression	-	145,262	-	-	-	-	-	-	-	-
Exfoliation Energy	Regression	812	-	-	-	-	-	-	-	-	-
Bulk Modulus (Kv)	Regression	23,823	-	-	-	-	-	-	-	-	-
Shear Modulus (Gv)	Regression	23,823	-	-	-	-	-	-	-	-	-
SLME	Regression	9,765	-	-	-	-	-	-	-	-	-
Spillage	Regression	11,377	-	-	-	-	-	-	-	-	-
ϵ_x (OPT)	Regression	52,158	-	-	-	-	-	-	-	-	-
ϵ (DFPT)	Regression	4,704	-	-	-	-	-	-	-	-	-
Max. piezoelectri c strain coeff (dij)	Regression	3,347	-	-	-	-	-	-	-	-	-
Max. piezo. stress coeff (eij)	Regression	4,797	-	-	-	-	-	-	-	-	-
Max. EFG	Regression	11,871	-	-	-	-	-	-	-	-	-
Avg. m_e	Regression	17,643	-	-	-	-	-	-	-	-	-
Is Stable	Classification	-	145,262	-	-	-	-	-	-	-	-
Is Gap Direct	Classification	-	145,262	-	-	-	-	-	-	-	-
n-Seedbeck	Regression	23,211	-	-	-	-	-	-	-	-	-
n-PF	Regression	23,211	-	-	-	-	-	-	-	-	-
p-Seedbeck	Regression	23,211	-	-	-	-	-	-	-	-	-
p-PF	Regression	23,211	-	-	-	-	-	-	-	-	-
Density	Regression	-	145,262	-	-	384,871	-	-	-	-	-
Density Atomic	Regression	-	145,262	-	-	-	-	-	-	-	-
Volume	Regression	-	145,262	-	-	384,871	-	-	-	-	-
Volume Per Atom (VPA)	Regression	-	-	-	-	-	-	84,024	-	-	-
Is Direct	Classification	-	-	10,481	-	-	-	-	-	-	-
Is Direct HSE	Classification	-	-	10,481	-	-	-	-	-	-	-
SOC	Classification	-	-	10,481	-	-	-	-	-	-	-
LCD	Regression	-	-	-	133,524	-	-	-	16,340	-	-
PLD	Regression	-	-	-	133,524	-	-	-	16,340	-	-
Max CO2	Regression	-	-	-	133,524	-	-	-	-	-	-
Min CO2	Regression	-	-	-	133,524	-	-	-	-	-	-
Void Fraction	Regression	-	-	-	133,524	-	-	-	-	-	-
Surface Area m2g	Regression	-	-	-	133,524	-	-	-	-	-	-
Surface Area m2cm3	Regression	-	-	-	133,524	-	-	-	-	-	-

C Chat-like Model Inference Details

Table 8: Hyperparameters used during inference. Temp. represents temperature.

Model Type	Model Name	Input Length	Output Length	Batch Size	Temp.	Top_K
Llama Family	Llama 2-7b-chat-hf	4000	256	256	0.8	10
	Llama 3-8b-Instruct	8000	256	256	0.8	10
	Llama 3.1-8b-Instruct	98000	256	128	0.8	10
	Llama 3.2-3b-Instruct	47000	256	128	0.8	10
Gemma Family	Gemma 7b-it	4000	256	256	0.8	10
	Gemma 1.1-7b-it	4000	256	256	0.8	10
	Gemma 2-9b-it	3000	256	256	0.8	10
Mistral Family	Mistral 7b-Instruct-v0.1	20000	256	256	0.8	10
	Mistral 7b-Instruct-v0.2	20000	256	256	0.8	10
	Mistral 7b-Instruct-v0.3	20000	256	256	0.8	10

D Prompt Templates



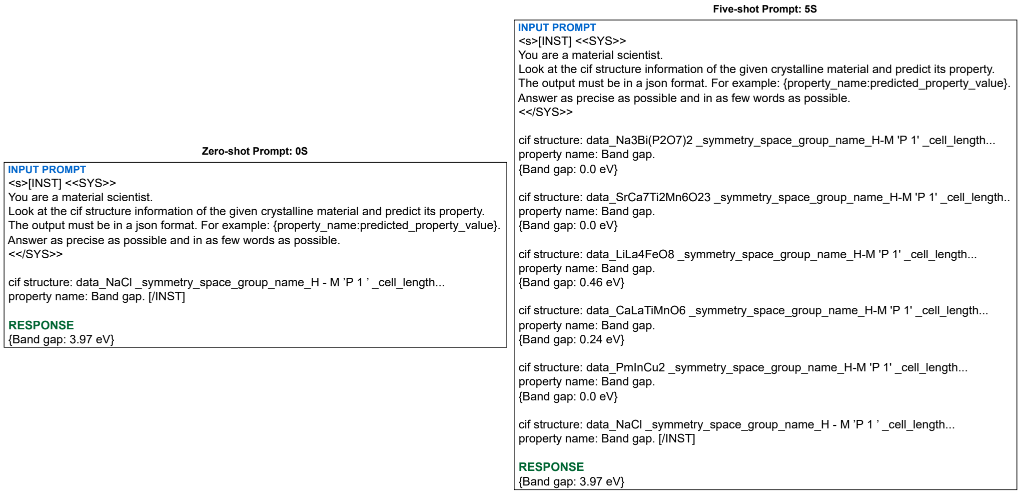
Zero-shot Prompt: 0S

```
INPUT PROMPT
<s>[INST] <<SYS>>
You are a material scientist.
Look at the chemical formula  of the given crystalline material and predict its property.
The output must be in a json format. For example: {property_name:predicted_property_value}.
Answer as precise as possible and in as few words as possible.
<</SYS>>

chemical formula: NaCl
property name: Band gap. [/INST]

RESPONSE
{Band gap: 3.97 eV}
```

Five-shot Prompt: 5S

```
INPUT PROMPT
<s>[INST] <<SYS>>
You are a material scientist.
Look at the chemical formula of the given crystalline material and predict its property.
The output must be in a json format. For example: {property_name:predicted_property_value}.
Answer as precise as possible and in as few words as possible.
<</SYS>>

chemical formula: Na3Bi(P2O7)2
property name: Band gap.
{Band gap: 0.0 eV}

chemical formula: SrCa7Ti2Mn6O23
property name: Band gap.
{Band gap: 0.0 eV}

chemical formula: LiLa4FeO8
property name: Band gap.
{Band gap: 0.46 eV}

chemical formula: CaLaTiMnO6
property name: Band gap.
{Band gap: 0.24 eV}

chemical formula: PmCu2In
property name: Band gap.
{Band gap: 0.0 eV}

chemical formula: NaCl
property name: Band gap. [/INST]

RESPONSE
{Band gap: 3.97 eV}
```

Figure 4: Prompt templates when the input is a chemical formula.



Zero-shot Prompt: 0S

```
INPUT PROMPT
<s>[INST] <<SYS>>
You are a material scientist.
Look at the cif structure information of the given crystalline material and predict its property.
The output must be in a json format. For example: {property_name:predicted_property_value}.
Answer as precise as possible and in as few words as possible.
<</SYS>>

cif structure: data_NaCl _symmetry_space_group_name_H - M 'P 1 ' _cell_length...
property name: Band gap. [/INST]

RESPONSE
{Band gap: 3.97 eV}
```

Five-shot Prompt: 5S

```
INPUT PROMPT
<s>[INST] <<SYS>>
You are a material scientist.
Look at the cif structure information of the given crystalline material and predict its property.
The output must be in a json format. For example: {property_name:predicted_property_value}.
Answer as precise as possible and in as few words as possible.
<</SYS>>

cif structure: data_Na3Bi(P2O7)2 _symmetry_space_group_name_H-M 'P 1' _cell_length...
property name: Band gap.
{Band gap: 0.0 eV}

cif structure: data_SrCa7Ti2Mn6O23 _symmetry_space_group_name_H-M 'P 1' _cell_length..
property name: Band gap.
{Band gap: 0.0 eV}

cif structure: data_LiLa4FeO8 _symmetry_space_group_name_H-M 'P 1' _cell_length...
property name: Band gap.
{Band gap: 0.46 eV}

cif structure: data_CaLaTiMnO6 _symmetry_space_group_name_H-M 'P 1' _cell_length...
property name: Band gap.
{Band gap: 0.24 eV}

cif structure: data_PmInCu2 _symmetry_space_group_name_H-M 'P 1' _cell_length...
property name: Band gap.
{Band gap: 0.0 eV}

cif structure: data_NaCl _symmetry_space_group_name_H - M 'P 1 ' _cell_length...
property name: Band gap. [/INST]

RESPONSE
{Band gap: 3.97 eV}
```

Figure 5: Prompt templates when the input is a CIF file.

Zero-shot Prompt: 0S

```
INPUT PROMPT
<s>[INST] <<SYS>>
You are a material scientist.
Look at the structure description of the given crystalline material and predict its property.
The output must be in a json format. For example: {property_name:predicted_property_value}.
Answer as precise as possible and in as few words as possible.
<</SYS>>

structure description: NaCl is Tetraauricupride structured and crystallizes in the cubic P m3m...
property name: Band gap. [/INST]

RESPONSE
{Band gap: 3.97 eV}
```

Five-shot Prompt: 5S

```
INPUT PROMPT
<s>[INST] <<SYS>>
You are a material scientist.
Look at the structure description of the given crystalline material and predict its property.
The output must be in a json format. For example: {property_name:predicted_property_value}.
Answer as precise as possible and in as few words as possible.
<</SYS>>

structure description: Na3Bi(P2O7)2 crystallizes in the triclinic P-1 space group...
property name: Band gap.
{Band gap: 0.0 eV}

structure description: SrCa7Ti2Mn6O23 crystallizes in the triclinic P1 space group...
property name: Band gap.
{Band gap: 0.0 eV}

structure description: LiLa4FeO8 is (La,Ba)CuO4-derived structured and crystallizes in the...
property name: Band gap.
{Band gap: 0.46 eV}

structure description: CaLaTiMnO6 is Orthorhombic Perovskite-derived structured and crysta...
property name: Band gap.
{Band gap: 0.24 eV}

structure description: PmCu2In is Heusler structured and crystallizes in the trigonal R-3m...
property name: Band gap.
{Band gap: 0.0 eV}

structure description: NaCl is Tetraauricupride structured and crystallizes in the cubic P m3m...
property name: Band gap. [/INST]

RESPONSE
{Band gap: 3.97 eV}
```

Figure 6: Prompt templates when the input is a crystal structure description.

E Result Details for Each Dataset

Table 9: Results for MP dataset. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better) while that of classification tasks (Is Stable and Is Gab Direct) is evaluated in terms of AUC score. FEPA: Formation Energy Per Atom, EPA: Energy Per Atom.

Input	Model	FEPA 145.2K	Bandgap 145.3K	EPA 145.2K	Ehull 145.2K	Efermi 145.2K	Density 145.2K	Density Atomic 145.2K	Volume 145.2K	Is Stable 145.2K	Is Gab Direct 145.2K
CIF	CGCNN (baseline)	8.151	3.255	7.224	3.874	3.689	8.773	5.888	1.703	0.882	0.810
Comp.	Llama 2-7b-chat:0S	0.008	0.623	0.009	0.001	0.003	0.967	0.754	0.747	0.500	0.482
	Llama 2-7b-chat:5S	0.33	1.217	0.239	0.132	0.706	0.899	0.724	0.771	0.502	0.512
	MatBERT-109M	**8.151**	**2.971**	**9.32**	**2.583**	**3.527**	**7.626**	**5.26**	**3.099**	**0.764**	**0.681**
	LLM-Prop-35M	7.482	2.345	7.437	2.006	3.159	6.682	3.523	2.521	0.746	0.636
CIF	Llama 2-7b-chat:0S	0.032	0.135	0.022	0.001	0.015	0.97	0.549	1.41	0.503	0.499
	Llama 2-7b-chat:5S	Inval.	1.111	0.289	Inval.	0.685	0.98	0.99	0.926	0.498	0.506
	MatBERT-109M	11.017	3.423	13.244	**3.808**	4.435	10.426	**6.686**	6.58	**0.790**	**0.710**
	LLM-Prop-35M	**14.322**	**3.758**	**17.354**	2.182	**4.515**	**13.834**	4.913	7.556	0.776	0.700
Descr.	Llama 2-7b-chat:0S	0.019	0.633	0.023	0.001	0.008	1.31	0.693	0.807	0.500	0.500
	Llama 2-7b-chat:5S	0.394	1.061	0.297	0.247	0.684	0.916	0.782	0.704	0.500	0.504
	MatBERT-109M	11.935	3.524	13.851	**4.085**	4.323	9.9	**6.899**	6.693	**0.794**	**0.713**
	LLM-Prop-35M	**15.913**	**3.931**	**18.412**	2.74	**4.598**	**14.388**	4.063	**8.888**	**0.794**	0.690

Table 10: Results for JARVIS-DFT. he performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better). FEPA: Formation Energy Per Atom, Tot. En.: Total Energy, Exf. En.: Exfoliation Energy.

Input	Model	FEPA 75.9K	Bandgap (OPT) 75.9K	Tot. En. 75.9K	Ehull 75.9K	Bandgap (MBJ) 19.8K	Kv 23.8K	Gv 23.8K	SLME 9.7K	Spillage 11.3K	ϵ_x (OPT) 18.2K
CIF	CGCNN (baseline)	13.615	4.797	22.906	1.573	4.497	3.715	2.337	1.862	1.271	2.425
Comp.	Llama 2-7b-chat:0S	0.021	0.011	0.02	0.005	0.92	0.428	0.374	0.148	Inval.	0.18
	Llama 2-7b-chat:5S	0.886	0.011	0.02	1.292	0.979	0.88	0.992	0.456	0.85	1.148
	MatBERT-109M	**6.808**	**4.083**	**9.21**	**2.786**	**3.755**	**2.906**	**1.928**	**1.801**	**1.243**	**2.017**
	LLM-Prop-35M	4.765	2.621	5.936	2.073	2.922	2.162	1.654	1.575	1.14	1.734
CIF	Llama 2-7b-chat:0S	0.023	0.011	0.02	0.002	0.193	0.278	0.358	0.186	0.702	0.781
	Llama 2-7b-chat:5S	0.859	Inval.	Inval.	1.173	1.054	0.874	0.91	0.486	0.916	1.253
	MatBERT-109M	10.211	**5.483**	15.673	**4.862**	**5.344**	**4.283**	**2.6**	**2.208**	**1.444**	**2.408**
	LLM-Prop-35M	**12.996**	3.331	**22.058**	2.648	4.93	4.121	2.409	2.175	1.37	2.135
Descr.	Llama 2-7b-chat:0S	0.007	0.011	0.02	0.004	0.94	0.498	0.382	0.07	0.135	0.647
	Llama 2-7b-chat:5S	0.845	0.011	0.02	1.273	1.033	0.87	0.969	0.461	0.857	1.201
	MatBERT-109M	10.211	**5.33**	15.141	**4.691**	**5.01**	**4.252**	**2.623**	**2.178**	**1.452**	**2.384**
	LLM-Prop-35M	**12.614**	3.427	**23.509**	4.532	4.983	4.128	2.419	2.061	1.307	2.334

		ϵ (DFPT) 4.7K	Max. Piezo. (dij) 3.3K	Max. Piezo. (eij) 4.7K	Max. EFG 11.8K	Exf. En. 0.8K	Avg. m_e 17.6K	n-Seebeck 23.2K	n-PF 23.2K	p-Seebeck 23.2K	p-PF 23.2K
CIF	CGCNN (baseline)	1.12	0.418	1.291	1.787	0.842	1.796	2.23	1.573	3.963	1.59
Comp.	Llama 2-7b-chat:0S	0.012	0.121	0.001	0.141	0.384	0.028	0.874	0.801	0.971	0.874
	Llama 2-7b-chat:5S	1.416	1.289	1.305	0.765	0.512	0.535	1.008	1.04	0.93	0.568
	MatBERT-109M	**1.533**	**1.464**	1.426	**1.658**	**1.124**	**2.093**	**1.908**	**1.318**	**2.752**	**1.356**
	LLM-Prop-35M	1.454	1.447	**1.573**	1.38	1.042	1.658	1.725	1.145	2.233	1.285
CIF	Llama 2-7b-chat:0S	0.033	0.104	0.001	0.246	0.411	0.041	0.429	0.766	0.83	0.826
	Llama 2-7b-chat:5S	Inval.	Inval.	Inval.	0.796	0.51	Inval.	1.039	1.396	Inval.	Inval.
	MatBERT-109M	1.509	1.758	**2.405**	**2.143**	**1.374**	**2.45**	**2.268**	**1.446**	**3.337**	**1.476**
	LLM-Prop-35M	**1.578**	**2.103**	**2.405**	1.936	1.044	1.796	1.955	1.332	2.503	1.399
Descr.	Llama 2-7b-chat:0S	0.08	0.266	0.001	0.138	0.285	0.019	0.769	0.793	0.825	0.829
	Llama 2-7b-chat:5S	**1.649**	1.174	1.152	0.806	0.661	0.523	1.098	1.024	0.948	0.563
	MatBERT-109M	1.534	1.807	**2.556**	**2.081**	**1.36**	**2.597**	**2.241**	**1.432**	**3.26**	**1.565**
	LLM-Prop-35M	1.64	**2.116**	2.315	1.978	1.168	1.858	2.154	1.364	2.61	1.407

Table 11: Results for SNUMAT. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better) while that of classification tasks (Is Direct, Is Direct HSE, and SOC) is evaluated in terms of AUC score.

Input	Model	Bandgap GGA 10.3K	Bandgap HSE 10.3K	Bandgap GGA Optical 10.3K	Bandgap HSE Optical 10.3K	Is Direct 10.3K	Is Direct HSE 10.3K	SOC 10.3K
CIF	CGCNN (baseline)	2.075	2.257	1.727	1.835	0.691	0.675	0.800
Comp.	Llama 2-7b-chat:0S	0.797	0.948	1.156	0.859	0.503	0.484	Inval.
	Llama 2-7b-chat:5S	1.267	1.327	0.862	1.174	0.475	0.468	0.455
	MatBERT-109M	**1.899**	**1.975**	**1.646**	**1.793**	**0.671**	**0.645**	0.820
	LLM-Prop-35M	1.533	1.621	1.392	1.491	0.647	0.624	**0.829**
CIF	Llama 2-7b-chat:0S	0.346	0.454	1.09	0.838	0.479	0.488	0.500
	Llama 2-7b-chat:5S	Inval.	Inval.	Inval.	Inval.	0.494	0.500	0.427
	MatBERT-109M	**2.28**	**2.472**	**1.885**	1.889	**0.677**	0.650	**0.823**
	LLM-Prop-35M	1.23	2.401	1.786	**1.9**	0.661	**0.664**	0.656
Descr.	Llama 2-7b-chat:0S	0.802	0.941	1.013	0.779	0.499	0.509	Inval.
	Llama 2-7b-chat:5S	0.774	1.315	0.901	1.172	0.594	0.623	0.486
	MatBERT-109M	**2.298**	**2.433**	**1.901**	**1.978**	**0.683**	0.645	0.862
	LLM-Prop-35M	2.251	2.142	1.84	1.569	0.681	**0.657**	**0.866**

Table 12: Results for GNoME. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better). FEPA: Formation Energy Per Atom, DEPA: Decomposition Energy Per Atom, Tot. En.: Total Energy.

Input	**Model**	**FEPA** **376.2K**	**Bandgap** **282.7K**	**DEPA** **376.2K**	**Tot. En.** **282.7K**	**Volume** **282.7K**	**Density** **282.7K**
CIF	CGCNN (baseline)	34.57	8.549	2.787	7.443	7.967	56.077
Comp.	Llama 2-7b-chat:0S	0.002	0.177	0.0	0.088	0.455	0.368
	Llama 2-7b-chat:5S	0.194	0.086	0.255	0.765	1.006	0.865
	MatBERT-109M	**30.248**	**4.692**	**2.787**	8.57	13.157	15.145
	LLM-Prop-35M	25.472	3.735	1.858	**21.624**	**16.556**	**25.615**
CIF	Llama 2-7b-chat:0S	0.003	0.045	0.0	0.706	**43.331**	0.794
	Llama 2-7b-chat:5S	Inval.	0.087	Inval.	Inval.	1.029	0.878
	MatBERT-109M	24.199	**9.16**	**3.716**	15.309	16.691	16.467
	LLM-Prop-35M	**28.469**	3.926	3.344	**17.837**	17.082	**25.615**
Descr.	Llama 2-7b-chat:0S	0.002	0.114	0.0	0.661	0.654	0.805
	Llama 2-7b-chat:5S	0.192	0.086	0.106	0.75	1.006	0.891
	MatBERT-109M	**30.248**	**5.829**	**3.716**	**18.205**	**17.824**	16.599
	LLM-Prop-35M	28.469	5.27	**3.716**	17.02	17.02	**25.936**

Table 13: Results for hMOF. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better).

Input	**Model**	**Max CO2** **132.7K**	**Min CO2** **132.7K**	**LCD** **132.7K**	**PLD** **132.7K**	**Void Fraction** **132.7K**	**Surface Area m2g** **132.7K**	**Surface Area m2cm3** **132.7K**
CIF	CGCNN (baseline)	1.719	1.617	1.989	1.757	2.912	3.765	2.039
Comp.	Llama 2-7b-chat:0S	0.011	0.002	0.009	0.008	0.5	0.454	0.233
	Llama 2-7b-chat:5S	0.679	0.058	0.949	1.026	0.945	0.567	0.366
	MatBERT-109M	1.335	**1.41**	**1.435**	1.378	1.57	1.517	**1.367**
	LLM-Prop-35M	**1.41**	1.392	1.432	**1.468**	**1.672**	**1.657**	1.321
CIF	Llama 2-7b-chat:0S	0.017	0.003	0.016	0.011	0.549	0.54	0.359
	Llama 2-7b-chat:5S	Inval.	Inval.	0.951	1.067	Inval.	Inval.	Inval.
	MatBERT-109M	1.421	**1.428**	1.544	**1.482**	1.641	1.622	**1.461**
	LLM-Prop-35M	**1.564**	1.41	**1.753**	1.435	**1.9**	**1.926**	1.374
Descr.	Llama 2-7b-chat:0S	0.129	0.014	0.026	0.006	0.382	0.497	0.299
	Llama 2-7b-chat:5S	0.684	0.058	0.955	1.006	0.931	0.571	0.37
	MatBERT-109M	1.438	1.466	1.602	1.511	1.719	1.697	1.475
	LLM-Prop-35M	**1.659**	**1.486**	1.623	**1.789**	1.736	**2.144**	**1.508**

Table 14: Results for Cantor HEA. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better). FEPA: Formation Energy Per Atom, EPA:Energy Per Atom, VPA:Volume Per Atom.

Input	**Model**	**FEPA** **84.0K**	**EPA** **84.0K**	**Ehull** **84.0K**	**VPA** **84.0K**
CIF	CGCNN (baseline)	9.036	49.521	9.697	2.869
Comp.	Llama 2-7b-chat:0S	0.005	0.098	0.003	0.031
	Llama 2-7b-chat:5S	0.896	0.658	0.928	0.986
	MatBERT-109M	**3.286**	16.17	**5.134**	2.489
	LLM-Prop-35M	**3.286**	**22.638**	**5.134**	**2.543**
CIF	Llama 2-7b-chat:0S	0.001	0.084	0.0	0.004
	Llama 2-7b-chat:5S	Inval.	Inval.	Inval.	Inval.
	MatBERT-109M	7.229	17.607	9.187	5.809
	LLM-Prop-35M	**8.341**	**36.015**	**11.636**	**6.919**
Descr.	Llama 2-7b-chat:0S	0.001	0.101	0.164	0.011
	Llama 2-7b-chat:5S	0.797	0.615	0.938	0.93
	MatBERT-109M	7.229	17.607	9.187	5.881
	LLM-Prop-35M	**8.341**	**36.015**	**11.636**	**7.713**

Table 15: Results for QMOF. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better). Tot. En.: Total Energy.

Input	**Model**	**Bandgap**	**Tot. En.**	**LCD**	**PLD**
		7.6K	**7.6K**	**7.6K**	**7.6K**
CIF	CGCNN (baseline)	2.431	1.489	4.068	4.317
Comp.	Llama 2-7b-chat:0S	0.901	0.26	0.045	0.009
	Llama 2-7b-chat:5S	0.648	0.754	1.241	1.086
	MatBERT-109M	**1.823**	**1.695**	**2.329**	**2.349**
	LLM-Prop-35M	1.759	1.621	2.293	2.157
CIF	Llama 2-7b-chat:0S	0.201	0.244	0.02	0.011
	Llama 2-7b-chat:5S	Inval.	Inval.	Inval.	Inval.
	MatBERT-109M	1.994	**4.378**	2.908	2.818
	LLM-Prop-35M	**2.166**	4.323	**2.947**	**2.87**
Descr.	Llama 2-7b-chat:0S	0.358	0.217	0.025	0.006
	Llama 2-7b-chat:5S	0.777	0.713	1.125	1.17
	MatBERT-109M	**2.166**	4.133	**2.981**	**2.941**
	LLM-Prop-35M	2.091	**4.312**	2.831	2.829

Table 16: Results for JARVIS-QETB. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better). FEPA: Formation Energy Per Atom, EPA:Energy Per Atom, Tot. En.: Total Energy, Ind. Bandgap: Indirect Bandgap.

Input	**Model**	**FEPA**	**EPA**	**Tot. En.**	**Ind. Bandgap**
		623.9K	**623.9K**	**623.9K**	**623.9K**
CIF	CGCNN (baseline)	1.964	228.201	11.218	5.534
Comp.	Llama 2-7b-chat:0S	0.003	0.369	0.172	0.21
	Llama 2-7b-chat:5S	0.812	1.037	1.032	1.306
	MatBERT-109M	1.431	37.979	8.19	0.21
	LLM-Prop-35M	**2.846**	**211.757**	**21.309**	**1.861**
CIF	Llama 2-7b-chat:0S	0.003	0.412	0.656	0.04
	Llama 2-7b-chat:5S	0.8	1.024	1.076	1.71
	MatBERT-109M	24.72	135.156	26.094	**4.779**
	LLM-Prop-35M	23.346	**318.291**	**48.192**	1.845
Descr.	Llama 2-7b-chat:0S	0.003	0.408	0.484	0.16
	Llama 2-7b-chat:5S	0.85	1.015	1.035	1.021
	MatBERT-109M	**26.265**	122.884	29.409	**7.788**
	LLM-Prop-35M	22.513	312.218	35.43	1.845

Table 17: Results for OQMD. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better). FEPA: Formation Energy Per Atom.

Input	Model	FEPA	Bandgap
		963.5K	**963.5K**
CIF	CGCNN (baseline)	22.291	6.701
Comp.	Llama 2-7b-chat:0S	0.019	0.192
	Llama 2-7b-chat:5S	1.013	1.306
	MatBERT-109M	7.662	**3.883**
	LLM-Prop-35M	**9.195**	2.845
CIF	Llama 2-7b-chat:0S	0.009	0.047
	Llama 2-7b-chat:5S	1.051	1.731
	MatBERT-109M	13.879	**7.163**
	LLM-Prop-35M	**18.861**	3.22
Descr.	Llama 2-7b-chat:0S	0.025	0.187
	Llama 2-7b-chat:5S	0.991	1.468
	MatBERT-109M	15.012	**7.041**
	LLM-Prop-35M	**16.346**	3.644

Table 18: Results for OMDB. The performance on regression tasks is evaluated in terms of MAD:MAE ratio (the higher the better).

Input	Model	Bandgap
		12.1K
CIF	CGCNN (baseline)	2.751
Comp.	Llama 2-7b-chat:0S	0.886
	Llama 2-7b-chat:5S	1.009
	MatBERT-109M	**1.554**
	LLM-Prop-35M	1.507
CIF	Llama 2-7b-chat:0S	0.159
	Llama 2-7b-chat:5S	0.930
	MatBERT-109M	**1.777**
	LLM-Prop-35M	**1.777**
Descr.	Llama 2-7b-chat:0S	0.155
	Llama 2-7b-chat:5S	1.002
	MatBERT-109M	**1.847**
	LLM-Prop-35M	1.656

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claim is the lack of standardized evaluation benchmarks and datasets for advancing the application of LLMs in predicting material properties. To address this, we introduced LLM4Mat-Bench, a benchmark dataset for evaluating LLM performance in predicting the properties of atomic and molecular crystals, as well as MOFs.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This work does not contain any theoretical assumptions and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We discussed the experimental details required to reproduce the results in Section 3.1 and we will publicly release the proposed benchmark and the code used to perform the experiments in our camera-ready version.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We commit to publicly release the proposed benchmark dataset and the code used to perform the experiments in our camera-ready version.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We discussed the experimental details required to reproduce the results in Section 3.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results details are discussed in Section 3.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided the details regarding the computer resources required to reproduce the results in Section 3.1.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed and conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The positive impacts of our work on advancing new materials discovery are discussed in Sections 1 and 4. We did not identify any negative societal impacts of our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The collected datasets do not pose any risks since they are from publicly trusted databases and are commonly used in materials sience community.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The collected data is from publicly available materials databases, and the licensing remains with their respective owners.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provided the details of the datasets included in our benchmark in Section 2.1.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.