ChatProp: Integrating Large Language Models and Physical Chemistry Tools for Enhanced Material Property Prediction

Anonymous ACL submission

Abstract

Material property prediction is essential for optimizing physical processes and developing novel materials in physical chemistry and materials science. Large language models (LLMs) have emerged as powerful tools for this task but encounter challenges in physics-related applications due to limited access to specialized external knowledge. To overcome these limitations, we present ChatProp, an intelligent agent that integrates first-principles (FP) calculations with machine learning-driven potential energy surface (PES) models to enhance the accuracy and efficiency of material property prediction. Leveraging LLMs such as GPT-4, ChatProp extracts critical information from textual inputs and generates appropriate responses, thereby eliminating the need for rigid, structured queries. The system forms a robust pipeline for tasks such as data retrieval and property prediction. In evaluations using GPT-4, ChatProp achieves accuracy rates of 96.8% for property prediction. As the first agent to integrate FP and machine learning PES models for material property prediction, ChatProp demonstrates the potential of combining LLMs with databases and machine learning in physical chemistry, showcasing transformative capabilities for future scientific advancements.

1 Introduction

In recent years, the field of generative artificial intelligence (AI) has experienced an unprecedented surge, primarily driven by the development of large language models (LLMs) such as BERT [Kenton and Toutanova, 2019], GPT-4 [OpenAI, 2023], and PaLM [Chowdhery et al., 2023]. These models, built on the transformer architecture [Vaswani, 2017], have revolutionized natural language processing by effectively handling complex language tasks and demonstrating capabilities similar to certain aspects of human cognition, including few-shot and zero-shot learning [Brown, 2020]. This proficiency is achieved through the analysis of extensive text corpora, underscoring the vast potential of LLMs across various domains. However, LLMs encounter significant limitations in performing precise mathematical computations and domain-specific tasks, such as physical processes[Schick et al., 2024; Castro Nascimento and Pimentel, 2023]. To address these shortcomings, a significant advancement in this swiftly changing domain is the emergence of autonomous LLM agents that augment LLMs with specialized external tools or plugins[Lowe et al., 2011; Shen et al., 2024]. They harness the capabilities of LLMs through prompt engineering [Reynolds and McDonell, 2021; Polak and Morgan, 2024; Zheng et al., 2023], fine-tuning [Bakker et al., 2022; Wei et al., 2021; Dunn et al., 2022], or integrating them with other scientific tools [Shen et al., 2024; Wu et al., 2023; M. Bran et al., 2024].

Despite significant progress in applying LLMs across diverse fields such as medicine [Lee et al., 2023; Waisberg et al., 2023] and biology [Nori et al., 2023; Wang et al., 2023], the full potential of this advanced technology within the physical sciences, particularly in material property prediction [Zhong et al., 2024], remains largely untapped. This limitation primarily stems from three key challenges. Firstly, many material entities lack suitable text-compatible input representations, impeding LLMs' ability to fully capture their complex properties. The inherent difficulties LLMs encounter in encoding physical structures limit their understanding and processing capabilities, thereby restricting their effectiveness in accurately predicting material properties [Hu et al., 2020; Ward et al., 2016]. Secondly, the scarcity of high-quality, domain-specific data in physical chemistry further exacerbates this issue. Unlike other scientific disciplines, physical chemistry suffers from a limited number of specialized databases and datasets, making it challenging to train sufficiently large-scale LLMs. This data de-

ficiency hampers the ability to represent and learn from the intricate physical chemical information necessary for precise property prediction [Zheng et al., 2023; Dagdelen et al., 2024]. Thirdly, the level of automation in physical chemistry remains relatively low compared to other fields, primarily due to its highly experimental nature [Zhao et al., 2022]. This lack of automation is particularly pronounced in material property prediction, where manual interventions are often required to interpret and validate results. Consequently, these limitations collectively underscore the necessity for an intelligent agent specifically designed for material property prediction. Such an agent [M. Bran et al., 2024] can integrate specialized computational tools and methodologies, overcoming the constraints of traditional LLMs and enhancing the accuracy and efficiency of property predictions in the physical sciences.

Current computational methods for material property prediction primarily involve machine learning (ML)-based potential energy surface (PES) models and first-principles (FP) calculations. FP methods, which do not rely on empirical force fields, begin with the initial configuration of a system and solve the Schrödinger equation based on atomic interactions and fundamental principles of quantum mechanics [Schleder et al., 2019]. In contrast, ML-based PES models significantly reduce computational costs [Lanzoni et al., 2022]. However, their applicability is limited to certain materials because of their specific model structures and specific training data, underscoring the continued necessity of FP calculations [Marcato et al., 2023]. Many current studies overly rely on pre-trained ML-based PES models, neglecting the critical role of FP methods and thereby constraining their systems' capabilities. For instance, in [Kang and Kim, 2024], the authors employ pre-trained models to predict material properties without considering the importance of FP methods. To address these challenges, we have developed ChatProp, an intelligent agent that synergistically integrates FP software predictions with ML-driven PES models. Contemporaneously with this work, a strategy is introduced to augment an LLM with external tools to accomplish complex tasks in physical chemistry [Boiko et al., 2023], which GPT-4 alone cannot handle. While their focus is on cloud laboratories, our approach encompasses a broader array of tasks and tools. Our agent overcomes the obstacles by seamlessly combining FP and ML-based PES models, facilitating accurate and efficient material property prediction without requiring extensive computational skills from users.

In this work, we present ChatProp, a pioneering intelligent agent developed for material property prediction, specifically tailored to materials science. Unlike other systems, ChatProp integrates both FP calculations and ML-driven PES models, marking the first such approach in the field. We have implemented nine tools, as shown in Figure 1 and detailed in Section 2, that empower ChatProp with comprehensive knowledge of material properties and the capacity to execute tasks using FP calculation software or pre-trained ML models directly. While the current set of tools is not exhaustive, ChatProp is designed to be flexible and can easily incorporate new tools to support emerging applications. Serving both as an assistant to expert scientists and a user-friendly interface for non-experts, ChatProp bridges the gap between complex material property prediction and accessible, accurate physical knowledge.

2 Methodology

By leveraging computational methods, researchers can efficiently explore vast physical spaces, thereby reducing reliance on time-consuming and costly experimental procedures. Inspired by successful applications in other fields [Schick et al., 2024; Yang et al., 2023; Shen et al., 2024], we propose ChatProp, an LLM-powered physical chemistry engine designed to streamline the reasoning process for various material property prediction tasks.

2.1 Design of ChatProp

ChatProp leverages multiple expert-designed tools for physical chemistry and operates by prompting an LLM (GPT-4 in our experiments) with specific instructions about the task and the desired format, as depicted in [Yao et al., 2022]. This process requires the LLM to reason about the current state of the task, consider its relevance to the final goal, and plan the next steps accordingly, thereby demonstrating its level of understanding. After the reasoning in the "Thought" step, the LLM requests a tool (preceded by the keyword "Action") and the input for this tool (preceded by the keyword "Action Input"). The text generation then pauses, and the program attempts to execute the requested function using the provided input. The result is returned to the LLM, preceded by the keyword "Evaluate", and the LLM

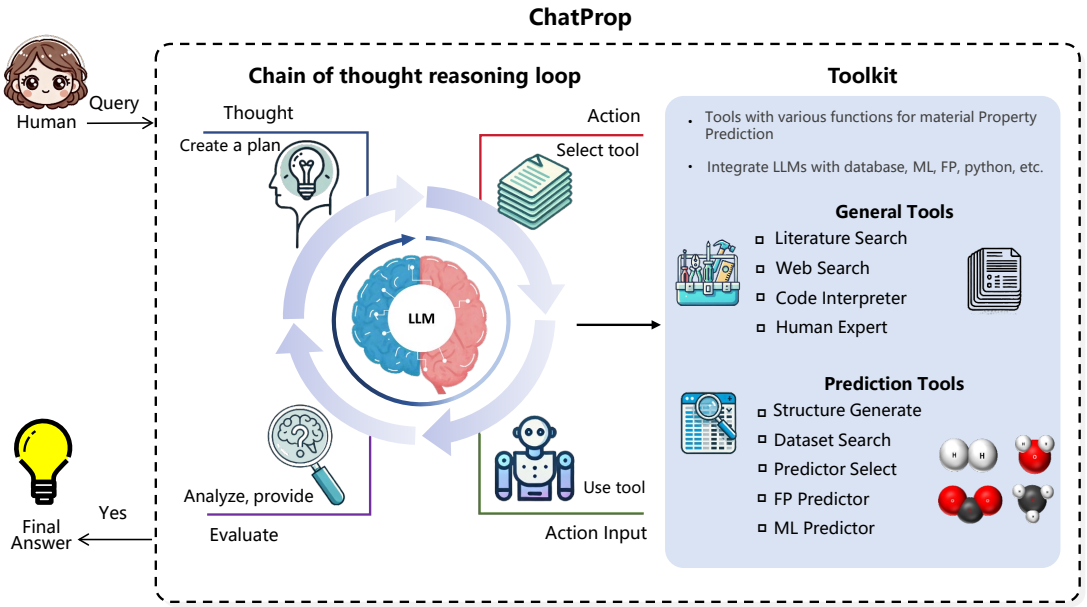

Figure 1: Schematic representation of ChatProp. The LLM proceeds through an automatic, iterative chain-of-thought process, which comprises four core components: Thought, Action, Action Input and Evaluate. A set of tools is created using a variety of physics-related packages and software, enabling the intelligent construction of first-principles (FP) and machine learning (ML) potential energy surfaces (PES) models. These tools and a user input are then given to an LLM. Upon receiving a query from human, the agent formulates a plan and selects a suitable toolkit. Subsequently, the toolkit generates outputs following the proposed plan, and the LLM evaluates the output obtained from tools and makes these results into a final response.

proceeds to the "Thought" step again. It continues iteratively until the final answer is reached.

The effectiveness of autonomous LLM agents is predicated on their capability to accurately extract essential details from textual inputs and offer relevant responses, irrespective of the presence of a rigidly structured query. This concept is clearly illustrated in ChatProp, as demonstrated in Figure 1. A user may pose a query in textual form regarding the properties of a material, to which ChatProp responds by supplying a detailed description related to the material in question. Moreover, the operational scope of this system extends beyond the simple retrieval of information. When a user expresses the need to get properties for specific materials, ChatProp is capable of generating the requested material properties accordingly.

2.2 Toolkit

While our current implementation incorporates a limited assortment of tools, it is important to highlight that this toolkit is highly extensible and can be easily expanded based on specific requirements and resource availability. The tools are categorized into two groups: general tools and prediction tools.

2.2.1 General Tools

Literature Search. The LitSearch tool is designed to extract pertinent information from scientific documents, including PDFs and text files (such as raw HTML), to generate precise and well-founded answers to user queries. This tool utilizes the `paper-qa` Python package [Skarlinski et al., 2024; Lála et al., 2023]. By employing OpenAI Embeddings and FAISS—a vector database—it efficiently embeds and searches through documents. Subsequently, a language model assists in formulating responses based on these embedded vectors.

Web Search. The WebSearch tool enables the language model to access relevant information from the internet. Using SerpAPI, the tool sends queries to search engines and aggregates snippets from the first page of Google search results. A notable feature of this tool is its ability to serve as a fallback option when the model faces a query it cannot handle or is uncertain about which tool to employ.

Code Interpreter. As one of LangChain's standard tools, Python REPL supplies ChatProp with an operational Python shell. This tool allows the LLM to write and execute Python code directly, facilitating the completion of a wide range of intri-

cate tasks, including numerical computations, AI model training, and data analysis.

Human Expert. The Human tool functions as a direct interface for interacting with users, enabling the engine to pose questions and receive responses from the user. The LLM may invoke this tool when it encounters challenges or uncertainties regarding the subsequent steps.

2.2.2 Prediction Tools

Dataset Search. When a user requests information about specific material structures present in the databases, ChatProp can identify and provide the required information from the pre-tabulated data. Upon receiving a user query, ChatProp autonomously selects the most suitable method to retrieve the necessary data. It then generates Python code tailored to extract specific information from the database according to the predefined strategy, typically utilizing the pandas library for data extraction or filtering. The generated code is subsequently executed within ChatProp's designated executor. After processing the results, ChatProp autonomously determines the next steps required to formulate the final answer, which is then presented as the requested response.

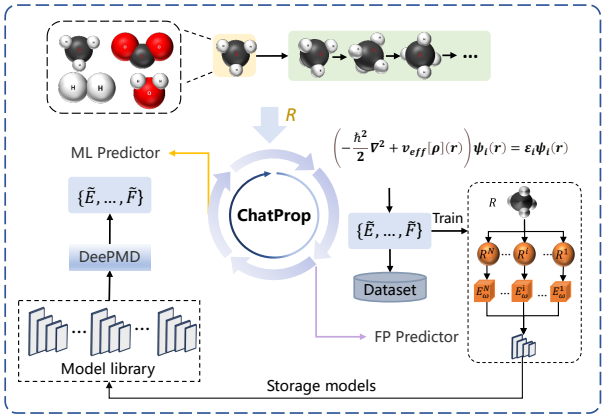

Figure 2: Tasks of ChatProp for predicting material properties. Given an input containing the materials' relevant information, the structure generate tool first obtains the molecular structure information R. Then the Predictor Select tool explores R to decide whether to use FP predictor or ML predictor. The properties obtained from Ab initio calculations will be stored in a dataset and used to train a pre-trained model, enabling fast prediction of properties for similar substances.

Predictor Select. The accuracy of the look-up table search is contingent on the precalculated values available in the specific files. For queries regarding the properties of materials that are not available, computational simulation can serve as an attractive alternative method; however, simulations are time-intensive processes and require an abundance of computational resources. The best resolution to such challenges is to first judge and choose the appropriate calculation method. In this work, ChatProp determines whether to choose FP calculation or ML models based on the following methods [Zhang et al., 2020].

Given a configuration $\mathcal{R}_t$, with t labeling a continuous or discrete series of operations, we define the error indicator ϵ_t as the maximal standard deviation of the atomic force predicted by the model ensemble,

$$\epsilon_t = \max_i \sqrt{\langle \|F_{w,i}(\mathcal{R}_t) - \langle F_{w,i}(\mathcal{R}_t)\rangle\|^2\rangle} \quad (1)$$

where $F_{w,i}(\mathcal{R}_t) = -\nabla_i E_w(\mathcal{R}_t)$ denotes the force on the atom with index i predicted by the model E_w, and ∇_i denotes the derivative with respect to the coordinate of the i-th atom. Both of the notations $\langle \ldots \rangle$ in Eq. 2 denote the expectation with respect to the ensemble of models and are estimated by the average of model predictions. For example, $\langle F_{w,i}(\mathcal{R}_t)\rangle$ is estimated by

$$\langle F_{w,i}(\mathcal{R}_t)\rangle = \frac{1}{N_m}\sum_{\alpha=1}^{N_m} F_{w_\alpha,i}(\mathcal{R}_t) \quad (2)$$

In our approach, the selection of the appropriate computational tool for predicting material properties is governed by a threshold σ_{lo}. This threshold is carefully chosen, not arbitrarily small, but rather set slightly above the accuracy achieved during model training. This ensures that the model is not overly confident in its predictions for configurations that are less reliable, while still leveraging the power of the pre-trained models where appropriate. To facilitate this tool selection process, we design Algorithm 1 to classify molecular configurations into two distinct sets: R_{ml} and R_{fp}. These sets represent the configurations that are suitable for prediction by the ML Predictor tool and the FP Predictor tool, respectively. Specifically, configurations in the R_{ml} set are those for which the predicted atomic forces fall within the acceptable error range defined by $\epsilon_t < \sigma_{\text{lo}}$, meaning that the pre-trained ML models can be confidently used for prediction. On the other hand, configurations in the R_{fp} set have larger prediction errors ($\epsilon_t \geq \sigma_{\text{lo}}$), indicating that more accurate calculations should be performed using FP methods.

Therefore, for a new structure, the algorithm determines whether it belongs to R_{ml} or R_{fp} based on the calculated error indicator ϵ_t. This decision-

Algorithm 1 Prediction Tool Selection Algorithm

Require: Ensemble of Models $\{E_1, E_2, \ldots, E_{N_m}\}$, Set of Configurations $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$, Threshold σ_{lo}
Ensure: Output Sets R_{ml}, R_{fp}

Initialize $R_{\text{ml}} \leftarrow \emptyset$
Initialize $R_{\text{fp}} \leftarrow \emptyset$
for each configuration $\mathcal{R}_t$ in $\{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ **do**
  for each model E_α in $\{E_1, \ldots, E_{N_m}\}$ **do**
    Compute $F_{w,\alpha,i}(\mathcal{R}_t) = -\nabla_i E_w(\mathcal{R}_t)$ for each atom i
  end for
  for each atom i **do**
    $\epsilon_t = \max_i \sqrt{\langle \|F_{w,i}(\mathcal{R}_t) - \langle F_{w,i}(\mathcal{R}_t)\rangle\|^2\rangle}$
    $\langle F_{w,i}(\mathcal{R}_t)\rangle = \frac{1}{N_m}\sum_{\alpha=1}^{N_m} F_{w,\alpha,i}(\mathcal{R}_t)$
  end for
  $\epsilon_t \leftarrow \max_i \sigma_i$
  if $\epsilon_t \geq \sigma_{\text{lo}}$ **then** Add $\mathcal{R}_t$ to
    $R_{\text{fp}} = \{\mathcal{R}_n \mid n \in I_{\text{fp}}, I_{\text{fp}} = \{n \mid \epsilon_t \geq \sigma_{\text{lo}}\}\}$
  elseAdd $\mathcal{R}_t$ to
    $R_{\text{ml}} = \{\mathcal{R}_n \mid n \in I_{\text{ml}}, I_{\text{ml}} = \{n \mid \epsilon_t < \sigma_{\text{lo}}\}\}$
  end if
end for
return R_{ml}, R_{fp}

making process ensures that the most appropriate and computationally efficient tool is used for each material property prediction.

Structure Generate. The Structure Generation tool utilizes the Material Project database to obtain accurate and reliable molecular structures. The Material Project offers a comprehensive repository of material properties and structures, enabling ChatProp to access high-quality data essential for subsequent predictive tasks. After acquiring the molecular structure, it undergoes a normalization process to ensure consistency and enhance computational efficiency. This normalization is performed using the Gram-Schmidt orthogonalization method:

$$u_k = v_k - \sum_{i=1}^{k-1} \frac{\langle v_k, u_i\rangle}{\langle u_i, u_i\rangle} u_i \tag{3}$$

where v_k is the original vector representing the k-th atomic position in the molecular structure, and u_i are the orthonormal basis vectors derived from the preceding steps ($i = 1, 2, \ldots, k-1$).

ML Predictor. As shown in Figure 2, due to its ability to efficiently handle high-dimensional potential energy surfaces and accurately model complex material behaviors through deep learning techniques, the DeePMD [Wen et al., 2021] is employed by the ML Predictor to predict material properties. ML-based PES models provide substantial reductions in computational costs while maintaining comparable accuracy. By training on extensive datasets derived from FP calculations, ML models can learn complex relationships between molecular structures and their properties [Lanzoni et al., 2022]. A common approach involves representing the potential energy $E_{\text{PES}}(\mathbf{R})$ as a function learned by the ML model:

$$E_{\text{PES}}(\mathbf{R}) \approx \text{ML Model}(\mathbf{R}), \tag{4}$$

where $\mathbf{R}$ denotes the atomic coordinates of the molecule.

Techniques such as neural networks, Gaussian processes, and kernel ridge regression have been employed to develop PES models capable of predicting properties like binding energies, reaction rates, and molecular conformations with impressive speed. For instance, the Neural Network Potential (NNP) can be expressed as

$$E_{\text{NNP}}(\mathbf{R}) = \sum_{i=1}^{N} \mathcal{N}(\mathbf{G}_i), \tag{5}$$

where $\mathcal{N}(\mathbf{G}_i)$ is the neural network function applied to the symmetry functions $\mathbf{G}_i$, which represent the local environment of the i-th atom, and N is the total number of atoms in the system.

In this work, we utilize the DeePMD, which employs deep neural networks to accurately represent the PES. It maps the symmetry functions to an energy contribution

$$E_i = \text{Deep Neural Network}(\mathcal{G}_i), \tag{6}$$

where $\mathcal{G}_i$ represents a different symmetry function for the i-th atom.

Then, the DeePMD model expresses the total energy as a sum of atomic contributions as

$$E_{\text{total}} = \sum_{i=1}^{N} E_i, \tag{7}$$

where E_i is the energy contribution from the i-th atom, and N is the total number of atoms in the system.

FP Predictor. The plane-wave density functional theory (PWDFT) platform [Hu et al., 2017a,b, 2021; Feng et al., 2024; Wu et al., 2024] is capable of computing detailed material properties while supporting multi-accelerator and parallel modes, which facilitates the rapid training and deployment of neural network PES models. Therefore, as shown in Figure 2, the FP Predictor utilizes PWDFT as the FP calculation software to obtain material properties. This tool performs FP calculations to determine energies, forces, and other relevant material properties. The process can be summarized as follows:

First-Principles Calculation. FP methods, rooted in quantum mechanics, provide a fundamental framework for predicting material properties without empirical parameters. They are renowned for

their high precision and reliability, making them indispensable for studying complex physical and chemical systems and reactions. The foundation of FP methods lies in solving the Schrödinger equation:

$$\hat{H}\Psi = E\Psi, \tag{8}$$

where $\hat{H}$ is the Hamiltonian operator, Ψ is the wavefunction of the system, and E is the energy eigenvalue. In the context of density functional theory (DFT), the energy of a system is expressed as a functional of the electron density $\rho(\mathbf{r})$:

$$E[\rho] = T[\rho] + \int V_{\text{ext}}(\mathbf{r})\rho(\mathbf{r})\, d\mathbf{r} + \frac{1}{2}\int\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\, d\mathbf{r}\, d\mathbf{r}' + E_{\text{xc}}[\rho], \tag{9}$$

where $T[\rho]$ is the kinetic energy functional, $V_{\text{ext}}(\mathbf{r})$ is the external potential, and $E_{\text{xc}}[\rho]$ represents the exchange-correlation energy functional.

The Kohn-Sham equations, which are central to DFT, are given by

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V_{\text{eff}}(\mathbf{r})\right)\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}), \tag{10}$$

where $\psi_i(\mathbf{r})$ are the Kohn-Sham orbitals, ϵ_i are the orbital energies, and $V_{\text{eff}}(\mathbf{r})$ is the effective potential,

$$V_{\text{eff}}(\mathbf{r}) = V_{\text{ext}}(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\, d\mathbf{r}' + \frac{\delta E_{\text{xc}}[\rho]}{\delta\rho(\mathbf{r})}. \tag{11}$$

Here, $\delta E_{\text{xc}}[\rho]/\delta\rho(\mathbf{r})$ denotes the functional derivative of the exchange-correlation energy with respect to the electron density $\rho(\mathbf{r})$.

Additionally, AIMD integrates these calculations into the equations of motion:

$$m_i\frac{d^2\mathbf{R}_i}{dt^2} = -\nabla_{\mathbf{R}_i}E[\rho], \tag{12}$$

where m_i is the mass of the i-th atom, $\mathbf{R}_i$ is the position vector of the i-th atom, and $\nabla_{\mathbf{R}_i}E[\rho]$ is the gradient of the energy with respect to the position of the i-th atom.

The FP Predictor performs the PWDFT to obtain the total energy E_{total}, forces $\mathbf{F}_i$, and other relevant properties for a given molecular structure.

Data Storage. The computed properties are stored in the dataset with corresponding molecular identifiers, facilitating quick retrieval for future predictions.

Training DeePMD. The stored data is used to train the DeePMD model. The training process involves optimizing the network parameters θ to minimize the loss function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N}\left(E_{\text{total}}^{\text{pred}} - E_{\text{total}}^{\text{true}}\right)^2 + \lambda\|\theta\|^2, \tag{13}$$

where $E_{\text{total}}^{\text{pred}}$ is the predicted total energy, $E_{\text{total}}^{\text{true}}$ is the true total energy from FP calculations, λ is a regularization parameter, and $\|\theta\|^2$ is the L2 norm of the network parameters to prevent overfitting.

Model Utilization. Once trained, the DeePMD model can predict the properties of similar molecules directly, bypassing the need for repetitive FP calculations and thereby enhancing computational efficiency.

3 Experiments

In this section, we delineate the experimental settings, procedures, and outcomes employed to evaluate the performance of ChatProp. The experiments are structured to assess the accuracy and efficiency of ChatProp in managing both data retrieval and property prediction tasks.

3.1 Experimental Settings

3.1.1 Model and Task Division

The primary model utilized in our experiments is GPT-4, integrated into the ChatProp framework. ChatProp employs a suite of specialized tools to execute material property predictions, which are categorized into two primary tasks. **Dataset Search Task**: This task involves retrieving material property data from existing databases; **Prediction Task**: This task entails predicting material properties using computational models when data retrieval is unsuccessful.

3.1.2 Experimental Metrics

Accuracies are evaluated using three categories: True, indicating that the task was completed successfully with correct and reliable results; Token False, signifying that the model's response exceeded the maximum token allowance, resulting in an incomplete output; and Logic False, denoting that ChatProp encountered a logical error during task execution, leading to incorrect or anomalous responses. These categories provide a comprehensive framework for assessing the performance and reliability of ChatProp across various scenarios. Furthermore, the average time is utilized to measure the duration of each task within each set of experiments.

3.2 Experimental Process and Results

To ensure the reliability and robustness of our evaluation, each experimental task has been conducted three times, resulting in a total of six experiments (three runs for each task type). Each run comprises 100 sample questions for both the Dataset Search

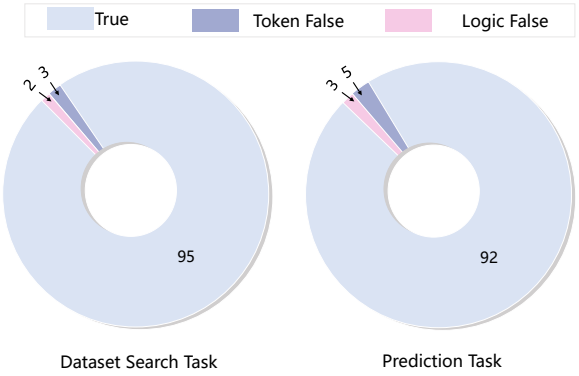

Figure 3: Depiction of average accuracies for tasks utilizing the GPT-4 model—search and prediction. Accuracies are evaluated based on three categories: True, Token False, and Logic False. Token False indicates that the Large Language Model (LLM) has exceeded the maximum token allowance, while Logic False indicates that ChatProp's logic has resulted in an incorrect response or anomaly. The numbers within the bars represent the counts of each category.

and Prediction tasks. Conducting multiple runs allows us to account for inherent variability in the model's performance and to compute average accuracies that more accurately reflect ChatProp's true capabilities. Figure 3 presents the accuracy measurements for the two tasks using ChatProp with GPT-4. Accuracy is evaluated over three runs, each comprising 100 sample questions for both the Dataset Search and Prediction tasks. The bar graph displays the number of questions in each accuracy category: True, Token False, and Logic False.

As shown in Table 1, for the Dataset Search Task, out of 100 questions, 95 are answered correctly (True), 3 exceed the token limit, and 2 contain logic errors. Excluding the instances where the token limit is exceeded, the Dataset Search Task achieves an accuracy of approximately 97.9%. Similarly, for the Prediction Task, 92 out of 100 questions are answered correctly (True), 5 exceed the token limit, and 3 contain logic errors, resulting in an accuracy of approximately 96.8% when excluding instances where the token limit is exceeded. The slightly lower accuracy observed in the Prediction Task compared to the Dataset Search Task is attributed to the inherent complexity of predictive modeling, which involves multiple computational steps and the integration of various tools.

Owing to the high accuracy rates, both tasks demonstrate ChatProp's effectiveness in providing reliable answers. These tasks are particularly significant because they address questions that cannot be effectively answered by directly querying LLMs. LLMs often fall short in delivering precise information due to their lack of detailed material-specific data, especially for properties that are challenging to ascertain through internet searches alone.

Run	Task	True	Token False	Logic False
1	Dataset Search	95	3	2
1	Prediction Task	92	5	3
2	Dataset Search	94	4	2
2	Prediction Task	93	4	3
3	Dataset Search	96	2	2
3	Prediction Task	91	6	3

Table 1: Accuracy measurements for dataset search and prediction tasks across three runs. The table presents the number of True, Token False, and Logic False outcomes for each run of the Dataset Search and Prediction tasks. Three experimental runs were conducted, each consisting of 100 sample questions per task.

Building upon our initial experiments, which demonstrate ChatProp's overall task completion capabilities, we have designed an ablation study to specifically evaluate its performance in completing prediction tasks and also compared it with existing chemical and physical agent strategies for predicting material properties. Additionally, the experimental results indicate that ChatProp outperforms existing physical chemistry agent strategies in predicting material properties [Kang and Kim, 2024]. This study compares three distinct approaches: using only ML models, using only FP software, and using both ML and FP methods as implemented in ChatProp. The rationale behind this comparison is to assess the effectiveness and efficiency of ChatProp in leveraging both computational strategies to enhance material property predictions.

The ablation experiments have been conducted by testing 100 tasks under each approach. The results, summarized in Table 2, reveal that using only pre-trained models results in a completion rate of 41.3%. The completion rate of the ML Group's tasks is much lower than that of the other two groups, indicating the serious shortcomings of existing methods. This is because that the effectiveness of the ML models is contingent upon the quality and diversity of the training data. In contrast to ML Group, the FP Group can achieve a higher accuracy but the time it spends on each task is much longer than that of the ML Group. It is noteworthy that relying solely on FP software achieves an 87.5% completion rate, albeit with each task taking more than twice the average time required by Chat-

Prop. Therefore, the FP Predictor tool is necessary. Further, when combining both ML and FP methods, ChatProp maintains a 96.8% completion rate while significantly reducing the average task completion time compared to the FP-only approach. These findings highlight the advantage of ChatProp in effectively integrating both computational methods to improve prediction accuracy and efficiency.

Method	Accuracy (%)	Average Time (min)
ML Group	41.3	0.856
FP Group	87.5	50.937
ChatProp	96.8	23.685

Table 2: Comparison of task completion and average time across different approaches. The table presents the average accuracies and average time per task for three groups: ML Group, which adopts only ML models; FP Group, which utilizes only first-principles software; and ChatProp. ChatProp achieves the highest accuracy with an average task completion time less than twice that of the FP-only approach.

To further illustrate ChatProp's capabilities, we conduct a case study addressing the following question: *"How do the total energy and the force for the centroid of* $(H_2O)_8$ *compare with those of* $(NH_3)_4$*?"*. In response, ChatProp first utilizes the Dataset Search tool to retrieve the total energy and force for the centroid of $(H_2O)_8$ from the database. However, it cannot locate corresponding properties for $(NH_3)_4$. Consequently, ChatProp invokes the Structure Generator tool to obtain the molecular structure of $(NH_3)_4$. Utilizing this structure, it employs the ML Predictor to calculate the total energy and the FP Predictor to determine the force for the centroid of $(NH_3)_4$. This sequential process enables ChatProp to provide a comprehensive comparison between the two molecular clusters, demonstrating its capability to effectively integrate data retrieval and predictive modeling. The detailed workflow of this process is illustrated in Figure 4.

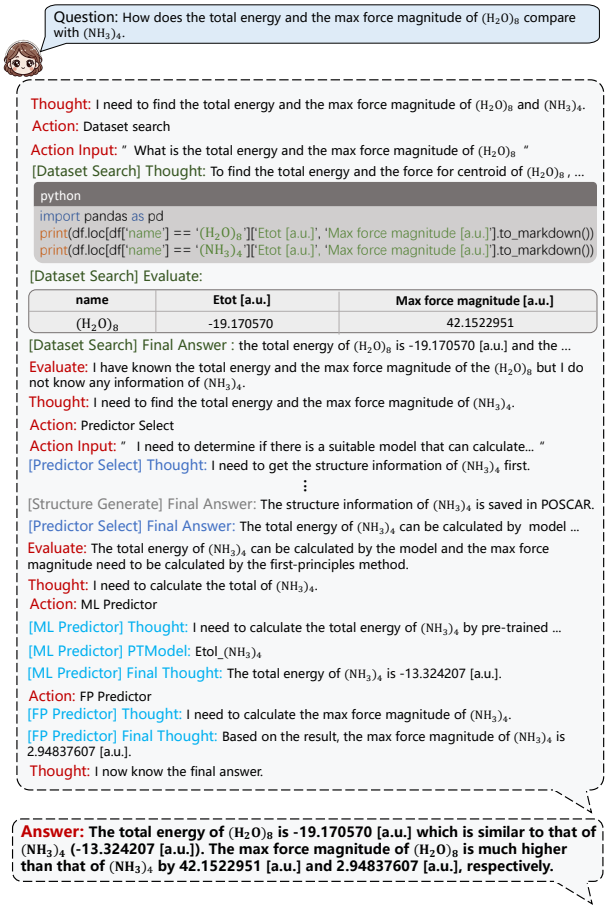

Figure 4: Example of a predictor for the question "How do the total energy and the force for the centroid of $(H_2O)_8$ compare with those of $(NH_3)_4$?" ChatProp accomplishes the task by employing the Dataset Search tool, the Structure Generator tool, ML Predictor tool, and FP Predictor tool.

4 Conclusion and Discussion

The investigation into the role of generative AI in natural science, specifically through the lens of ChatProp leveraging the strengths of FP calculations and ML-based PES models, unveils substantial potential for predicting material properties. Through the Dataset Search and Prediction Task frameworks, ChatProp demonstrates high accuracy rates of 97.9% and 96.8%, respectively, across three experimental runs involving 100 sample questions each. Furthermore, an ablation experiment is designed, and the comparison of experimental results further highlights the advantages of using this integration strategy. These results underscore ChatProp's reliability and effectiveness in providing precise material property predictions, particularly for complex queries that exceed the direct capabilities of standard LLMs.

5 Limitations

Despite its impressive performance, ChatProp's reliance on external computational tools such as DeePMD and PWDFT necessitates substantial computational resources, which may limit its scalability for extremely large or highly complex molecular systems. Future work focuses on addressing these limitations by optimizing the integration of computational tools to enhance scalability and efficiency.

References

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. 2022. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Cayque Monteiro Castro Nascimento and André Silva Pimentel. 2023. Do large language models understand chemistry? a conversation with ChatGPT. *Journal of Chemical Information and Modeling*, 63(6):1649–1655.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Alexander Dunn, John Dagdelen, Nicholas Walker, Sanghoon Lee, Andrew S Rosen, Gerbrand Ceder, Kristin Persson, and Anubhav Jain. 2022. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint arXiv:2212.05238*.

Junwei Feng, Lingyun Wan, Jielan Li, Shizhe Jiao, Xinhui Cui, Wei Hu, and Jinlong Yang. 2024. Massively parallel implementation of iterative eigensolvers in large-scale plane-wave density functional theory. *Computer Physics Communications*, 299:109135.

Tiantian Hu, Hui Song, Tao Jiang, and Shaobo Li. 2020. Learning representations of inorganic materials from generative adversarial networks. *Symmetry*, 12(11):1889.

Wei Hu, Lin Lin, Amartya S Banerjee, Eugene Vecharynski, and Chao Yang. 2017a. Adaptively compressed exchange operator for large-scale hybrid density functional calculations with applications to the adsorption of water on silicene. *Journal of Chemical Theory and Computation*, 13(3):1188–1198.

Wei Hu, Lin Lin, and Chao Yang. 2017b. Projected commutator DIIs method for accelerating hybrid functional electronic structure calculations. *Journal of Chemical Theory and Computation*, 13(11):5458–5467.

Wei Hu, Xinming Qin, Qingcai Jiang, Junshi Chen, Hong An, Weile Jia, Fang Li, Xin Liu, Dexun Chen, Fangfang Liu, yuwen Zhao, and Jinlong Yang. 2021. High performance computing of DGDFT for tens of thousands of atoms using millions of cores on Sunway TaihuLight. *Science Bulletin*, 66(2):111–119.

Yeonghun Kang and Jihan Kim. 2024. Chatmof: an artificial intelligence system for predicting and generating metal-organic frameworks using large language models. *Nature Communications*, 15(1):4705.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota.

Daniele Lanzoni, Fabrizio Rovaris, and Francesco Montalenti. 2022. Machine learning potential for interacting dislocations in the presence of free surfaces. *Scientific Reports*, 12(1):3760.

Peter Lee, Sebastien Bubeck, and Joseph Petro. 2023. Benefits, limits, and risks of GPT-4 as an AI chatbot for medicine. *New England Journal of Medicine*, 388(13):1233–1239.

Daniel M Lowe, Peter T Corbett, Peter Murray-Rust, and Robert C Glen. 2011. Chemical name to structure: OPSIN, an open source solution. *Journal of Chemical Information and Modeling*.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G. Rodriques, and Andrew D. White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11.

Agnese Marcato, Daniele Marchisio, and Gianluca Boccardo. 2023. Reconciling deep learning and first-principle modelling for the investigation of transport phenomena in chemical engineering. *The Canadian Journal of Chemical Engineering*, 101(6):3013–3018.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of GPT-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Maciej P Polak and Dane Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, 15(1):1569.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems*, pages 1–7.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Gabriel R Schleder, Antonio CM Padilha, Carlos Mera Acosta, Marcio Costa, and Adalberto Fazzio. 2019. From DFT to machine learning: Recent approaches to materials science–a review. *Journal of Physics: Materials*, 2(3):032001.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. HuggingGPT: Solving ai tasks with ChatGPT and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprent arXiv:2409.13740*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Ethan Waisberg, Joshua Ong, Mouayad Masalkhi, Sharif Amit Kamran, Nasif Zaman, Prithul Sarker, Andrew G Lee, and Alireza Tavakkoli. 2023. GPT-4: a new era of artificial intelligence in medicine. *Irish Journal of Medical Science (1971-)*, 192(6):3197–3200.

Yuqing Wang, Yun Zhao, and Linda Petzold. 2023. Are large language models ready for healthcare? a comparative study on clinical language understanding. In *Machine Learning for Healthcare Conference*, pages 804–823. PMLR.

Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. 2016. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2(1):1–7.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Tongqi Wen, Rui Wang, Lingyu Zhu, Linfeng Zhang, Han Wang, David J Srolovitz, and Zhaoxuan Wu. 2021. Specialising neural network potentials for accurate properties and application to the mechanical response of titanium. *npj Computational Materials*, 7(1):206.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Wentiao Wu, Zhengbang Zhou, Qingcai Jiang, Junwei Feng, Xinming Qin, Huanhuan Ma, Zhenwei Cao, Junshi Chen, Sheng Chen, Xinyong Meng, et al. 2024. Enabling 13k-atom excited-state gw calculations via low-rank approximations and hpc on the new sunway supercomputer. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14. IEEE.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting ChatGPT for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and E Weinan. 2020. DP-GEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253:107206.

Zhi-Wen Zhao, Marcos Del Cueto, and Alessandro Troisi. 2022. Limitations of machine learning models when predicting compounds with completely new chemistries: possible improvements applied to the discovery of new non-fullerene acceptors. *Digital Discovery*, 1(3):266–276.

Zhiling Zheng, Oufan Zhang, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. 2023. ChatGPT chemistry assistant for text mining and the prediction of MOF synthesis. *Journal of the American Chemical Society*, 145(32):18048–18062.

Zhiqiang Zhong, Kuangyu Zhou, and Davide Mottin. 2024. Benchmarking large language models for molecule prediction tasks. *arXiv preprint arXiv:2403.05075*.