Supplementary Information

1. Open-Source Code and Usage Details

We present the full open-source code for ChatMat. This code is made publicly available to facilitate the reproduction of the research and to support further development in this area.

1.1 Code Structure and Remote Server Setup

The open-source version of the ChatMat platform has been released at https://github.com/Xiaobu-USTC/ChatMat. Access to the proprietary GPT-4o Application Programming Interface (API) can be obtained through OpenAI. Specifically, it includes:

- Server Connection Code: As the experiments were conducted on a remote server equipped with 8 NVIDIA V100 GPUs, the code includes sections for connecting to remote servers. Users can make modifications according to their own needs and circumstances.
- README File: A README file is provided within the code repository, explaining the steps for easy deployment and execution of the project. This file includes instructions for setting up the necessary software environment, as well as how to run the code on both local and remote machines.
- Requirements File: All the dependencies for the project are explicitly listed in the *requirements.txt* file. This ensures that users can quickly install the required libraries and packages to run the code seamlessly.

1.2 OpenAI GPT-4o Configuration

Since this project is based on GPT-4o, it is essential to configure the OpenAI API details before running the code. Instructions for configuring OpenAI's API, including key setup and other necessary parameters, are also outlined in the README file.

The code is fully open-sourced to allow readers to reproduce the results of this work. By following the provided instructions, users will be able to deploy and run the project without issues.We hope that this open access will encourage further exploration and application of ChatMat and related tools in the scientific community.

2. Detailed Parameters for the PWDFT

The configuration parameters used for the density functional theory (DFT) calculations in the plane-wave density functional theory (PWDFT) software are provided in Table S1:

Table S1. Parameters for the PWDFT

Class	Parameter	Value
Base input parameters for PWDFT	Mixing_Variable	potential
	Mixing_Type	anderson
	Mixing_StepLength	0.8
	Mixing_MaxDim	9
Exchange-Correlation Functional	VDW_Type	DFT-D2
	Pseudo_Type	ONCV
	XC_Type	XC_GGA_XC_PBE
SCF iteration parameters for PWDFT	Ecut_Wavefunction	40.0
	Temperature	300.0
	SCF_Inner_Tolerance	1e-4
	SCF_Outer_Tolerance	1e-6
	SCF_Outer_MaxIter	30
AIMD settings for PWDFT	Ion_Move	verlet
	Ion_Max_Iter	100
	Ion_Temperature	300
	MD_SCF_Outer_MaxIter	999
	MD_SCF_Phi_MaxIter	999

This detailed configuration ensures that the computational setup used in this study is fully reproducible.

Our PWDFT is a first-principle molecular dynamics simulation of 100 steps (step size 1 femtosecond) at a plane wave base at 300K. Among them, we use Optimized

Norm-Conserving Vanderbilt (ONCV) pseudopotential and Perdew-Burke-Ernzerhof (PBE) functional, and add Van der Waals (VDW) correction, truncation can be 40 Hartree. At the same time, we use the inner and outer layer scf double iteration method to accelerate the calculation, in which the inner layer iteration convergence standard is 1e-4, and the outer layer iteration convergence standard is 1e-6.

3. Design of σ_{lo}

The deviation of the ensemble model serves as a reliable indicator of the true prediction error, effectively distinguishing between the trustworthy interpolation regime and the unreliable extrapolation regime. In our method, the selection of the appropriate computational tool for predicting material properties is determined by a threshold σ_{lo}. Empirical tests indicate that the system's performance is robust when σ_{lo} varies between 1.0 and 1.2 times the training Root Mean Squared Error (RMSE). Setting σ_{lo} too low ($\leq$1.0× RMSE) leads to excessive and unnecessary DFT calls (low efficiency), while setting it too high (>1.5× RMSE) risks accepting inaccurate predictions. Specifically, in our experiments, we set σ_{lo} to be 1.1 times higher than the converged training RMSE, which provides an optimal balance between computational efficiency and prediction accuracy.

4. Evaluation Rubric and Annotated Examples

To ensure transparency and reproducibility, we provide the detailed rubric shown in Table S2, along with annotated examples for each classification.

Table S2. Evaluation Rubric

Classification	Definition	Detailed Criteria
True	The response is scientifically accurate, logically consistent, and verifiable.	1. Tool Execution: Successfully invokes the correct tool. 2. File Verification: Generated files (e.g., POSCAR, DEN.cub) exist and contain valid chemical data. 3. Accuracy: Numerical values are within the acceptable approximate

		range. 4. Clarity: Textual output is unambiguous.
Factual False	The response contains factual errors, hallucinations, or logical contradictions.	1. Hallucination: Fabricating data without calculation. 2. Calculation Error: Values exceed error thresholds. 3. Logical Contradiction: Text contradicts data. 4. Wrong Species: Retrieving data for the wrong molecule. 5. Execution Failure: The agent fails to complete the task due to tool errors, timeouts, or exceptions, resulting in no answer.
Formatting False	The response is factually derived but fails to meet linguistic or syntactic standards.	1. Ambiguity: Grammatical structure prevents clear mapping between subjects and values. 2. Syntax Error: Malformed JSON/Code. 3. Parsing Failure: Output cannot be extracted by the automated parser.

Examples for each classification:

a) Example 1: Classification [True]

- Query: " Would you mind providing the charge density distribution of C_2H_4?"
- Response: " The charge density distribution of C_2H_4 has been successfully retrieved and saved to a DEN.cub file."
- Verification: The DEN.cub file exists in the directory and contains valid volumetric charge density data and atomic coordinates for the C2H4 molecule.

b) Example 2: Classification [Factual False]

- Query: "Could you list the details of the atomic Forces section for $Ca(OH)_2$?"

- Response: " I am unable to provide the details of the atomic forces for $Ca(OH)_2$ due to a persistent error with the DFT operator and ML performer."
- Reason: Although the agent truthfully reported the error and did not hallucinate, it failed to retrieve or calculate the requested chemical property. Since the ground truth data was not provided, this counts as a failure in Factuality.

c) Example 3: Classification [Formatting False]

- Query: " Where does the Fermi level sit for $KMnO_4$?"
- Response: " I am unable to determine the Fermi level of $KMnO_4$ due to invalid JSON format."
- Reason: The JSON format of the final output is incomplete, resulting in the incorrect extraction of the answer.

4 Dataset and Model Library

The database, which contains the generated data for material property retrieval, has been fully constructed and is now publicly available on GitHub. This database is designed to facilitate material property searches, providing essential data for various material-related research tasks. All datasets used in this research, including molecular properties and structural information, are included and accessible alongside the accompanying code.

In addition to the database, the machine learning potential energy surface (ML-PES) models, constructed for each material in the dataset, are stored in the Model Library. These ML-PES models are crucial for simulating material behaviors and understanding interactions at the atomic level. The models, which have been successfully generated and saved, allow researchers to predict material properties with high efficiency.

The public release of both the dataset and the model library enhances the convenience of material property prediction. Researchers can now prioritize using the data from

the dataset and the models from the Model Library before resorting to first-principles calculations. This approach not only simplifies the process but also greatly reduces computational effort and resource consumption, making material property predictions faster and more efficient.

Both the dataset and the machine learning models are stored in a structured and easy-to-use format. The relevant code for querying, utilizing, and expanding the database or models is also provided. These resources are freely available for use, modification, and extension by the broader scientific community. All materials, models, and the code can be accessed via the following GitHub repository: https://github.com/Xiaobu-USTC/ChatMat.