Supplementary Information for "Evaluating LLMs in Scientific Discovery"

Zhangde Song[1, †, ‡], Jieyu Lu[1, †], Yuanqi Du[2, †], Botao Yu[3, †], Thomas M. Pruyn[4, †], Yue Huang[5, †], Kehan Guo[5, †], Xiuzhe Luo[6, †], Yuanhao Qu[7, †], Yi Qu[8, ‡], Yinkai Wang[9, ‡], Haorui Wang[10, ‡], Jeff Guo[11, ‡], Jingru Gan[12, ‡], Parshin Shojaee[13, ‡], Di Luo[14, 15, ‡], Andres M Bran[11], Gen Li[16], Qiyuan Zhao[1], Shao-Xiong Lennon Luo[17], Yuxuan Zhang[18, 19, 20], Xiang Zou[4], Wanru Zhao[21], Yifan F. Zhang[22], Wucheng Zhang[23], Shunan Zheng[24], Saiyang Zhang[25], Sartaaj Takrim Khan[4], Mahyar Rajabi-Kochi[4], Samantha Paradi-Maropakis[26], Tony Baltoiu[27], Fengyu Xie[28], Tianyang Chen[29], Kexin Huang[7], Weiliang Luo[30, 31], Meijing Fang[32], Xin Yang[30], Lixue Cheng[33], Jiajun He[34], Soha Hassoun[9], Xiangliang Zhang[5], Wei Wang[12], Chandan K. Reddy[13], Chao Zhang[10], Zhiling Zheng[35], Mengdi Wang[22], Le Cong[7], Carla P. Gomes[2], Chang-Yu Hsieh[32], Aditya Nandy[36], Philippe Schwaller[11], Heather J. Kulik[30, 31], Haojun Jia[1, *], Huan Sun[3, *], Seyed Mohamad Moosavi[4, 18, *], and Chenru Duan[1, ‡, *]

[1]Deep Principle, Hangzhou, China
[2]Department of Computer Science, Cornell University, Ithaca, NY, USA
[3]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA
[4]Department of Chemical Engineering & Applied Chemistry, University of Toronto, Toronto, ON, Canada
[5]Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, USA
[6]QuEra Computing Inc., Boston, MA, USA
[7]Department of Pathology, Department of Genetics, Cancer Biology Program, Stanford University School of Medicine, Stanford, CA, USA
[8]Harvard Law School, Cambridge, MA, USA
[9]Department of Computer Science, Tufts University, Medford, MA, USA
[10]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA
[11]Laboratory of Artificial Chemical Intelligence, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland
[12]Department of Computer Science, University of California, Los Angeles, Los Angeles, CA, USA
[13]Department of Computer Science, Virginia Tech, Arlington, VA, USA
[14]Department of Physics, Tsinghua University, Beijing, China
[15]Institute for Advanced Study, Tsinghua University, Beijing, China
[16]Department of Chemistry, Princeton University, Princeton, NJ, USA
[17]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
[18]Vector Institute for Artificial Intelligence, Toronto, ON, Canada
[19]Department of Physics, University of Toronto, Toronto, ON, Canada
[20]Institute of Physics, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland
[21]Department of Computer Science and Technology, University of Cambridge, Cambridge, United Kingdom
[22]Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, USA
[23]Department of Physics, Princeton University, Princeton, NJ, USA
[24]Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX, USA
[25]Department of Physics, The University of Texas at Austin, Austin, TX, USA
[26]Department of Biomedical Engineering, University of Toronto, Toronto, ON, Canada
[27]Department of Mechanical Engineering, McGill University, Montreal, QC, Canada
[28]College of Artificial Intelligence and Data Science, Suzhou Institute of Advanced Research, University of Science and Technology of China, Suzhou, Jiangsu, China
[29]School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), Shenzhen, Guangdong, China
[30]Department of Chemistry, Massachusetts Institute of Technology, Cambridge, MA, USA
[31]Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA
[32]College of Pharmaceutical Sciences, Zhejiang University, Hangzhou, Zhejiang, China
[33]Department of Chemistry, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China
[34]Department of Engineering, University of Cambridge, Cambridge, United Kingdom
[35]Department of Chemistry, Washington University in St. Louis, St. Louis, MO, USA
[36]Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, USA
[†]These authors contribute equally
[‡]Project contributor
[*]Correspondence to: haojunjia@deepprinciple.com, sun.397@osu.edu, mohamad.moosavi@utoronto.ca, duanchenru@gmail.com

Abbreviation

The following is the list of abbreviations utilized in the main paper and Supplementary Information.

- LLM: Large Language Model
- SDE: Scientific Discovery Evaluation
- Q&A: Question and Answer
- API: Application Programming Interface
- RL: Reinforcement Learning
- MSE: Mean Squared Error
- NMSE: Normalized Mean Squared Error
- AUC: Area Under the Curve
- $\text{AUC}_{\text{top-}k}$: Area Under the Curve of Top-k Metric
- XML: Extensible Markup Language
- AIME: American Invitational Mathematics Examination
- MMMU: Multidiscipline Multimodal Benchmark for Universality
- GPQA: Graduate-level Google-Proof Scientific Q&A
- SWE-bench: Software Engineering Benchmark
- τ-bench: Tool–Agent–User Interaction Benchmark
- HLE: Humanity's Last Exam
- NMR: Nuclear Magnetic Resonance
- IR: Infrared Spectroscopy
- MS: Mass Spectrometry
- TMC: Transition Metal Complex
- MOF: Metal Organic Framework
- PXRD: Powder X-Ray Diffraction
- VASP: Vienna Ab-initio Simulation Package
- LAMMPS: Large-scale Atomic/Molecular Massively Parallel Simulator
- GFN2-xTB: Geometry-optimized eXtended Tight Binding Method
- SC: Synthetic Complexity (small molecule synthesizability metric)
- SA: Synthetic Accessibility (small molecule synthesizability metric)
- USPTO: United States Patent and Trademark Office
- MCTS: Monte Carlo Tree Search
- SMILES: Simplified Molecular Input Line Entry System
- HOMO-LUMO gap: Highest Occupied Molecular Orbital – Lowest Unoccupied Molecular Orbital gap
- E_d: Energy above the convex hull
- SUN: Stable, Unique, Novel (crystal structure metric)
- CHGNet: Crystal Hamiltonian Graph Neural Network
- CDVAE: Crystal Diffusion Variational Autoencoder
- DiffCSP: Diffusion Model for Crystal Structure Prediction
- GA: Genetic Algorithm
- GWAS: Genome-Wide Association Study
- CRISPR: Clustered Regularly Interspaced Short Palindromic Repeats

- IFNG: Interferon-gamma
- AAV: Adeno-Associated Virus
- GFP: Green Fluorescent Protein
- ID: In-Domain
- OOD: Out-of-Domain
- RDKit: Cheminformatics Software Toolkit
- ZINC: Small Molecule Database
- molSimplify: Transition Metal Complex Toolkit
- PySR: Python Symbolic Regression Package
- StructureMatcher: Pymatgen Structural Comparator
- MatBench: Materials Benchmark Dataset
- MatBench-bandgap: MatBench Bandgap Prediction Dataset

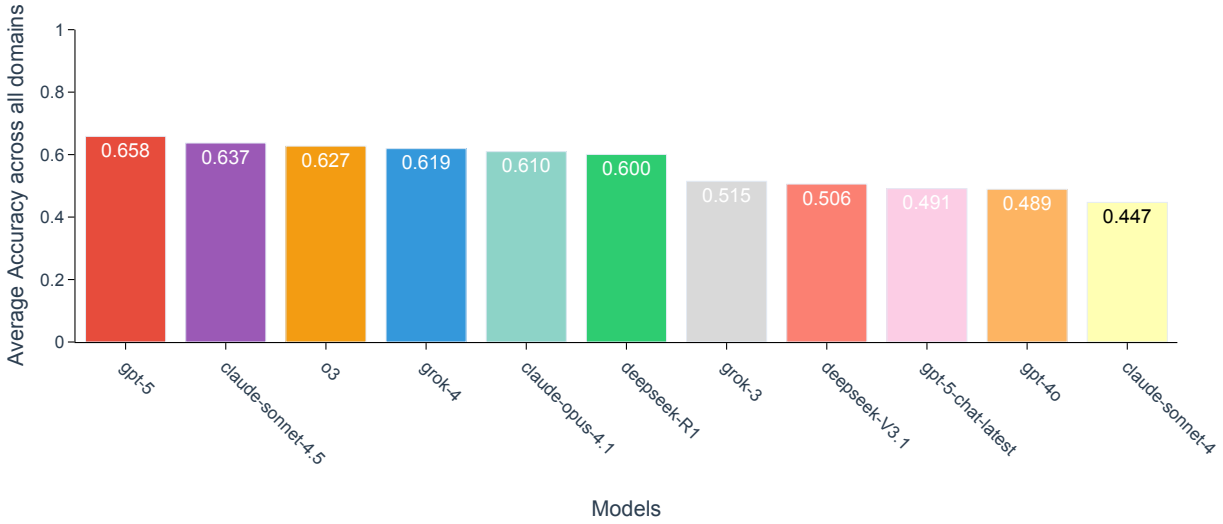

Figure 1. **Average model accuracy across all 43 research scenarios.** The models are ranked by the average accuracy.

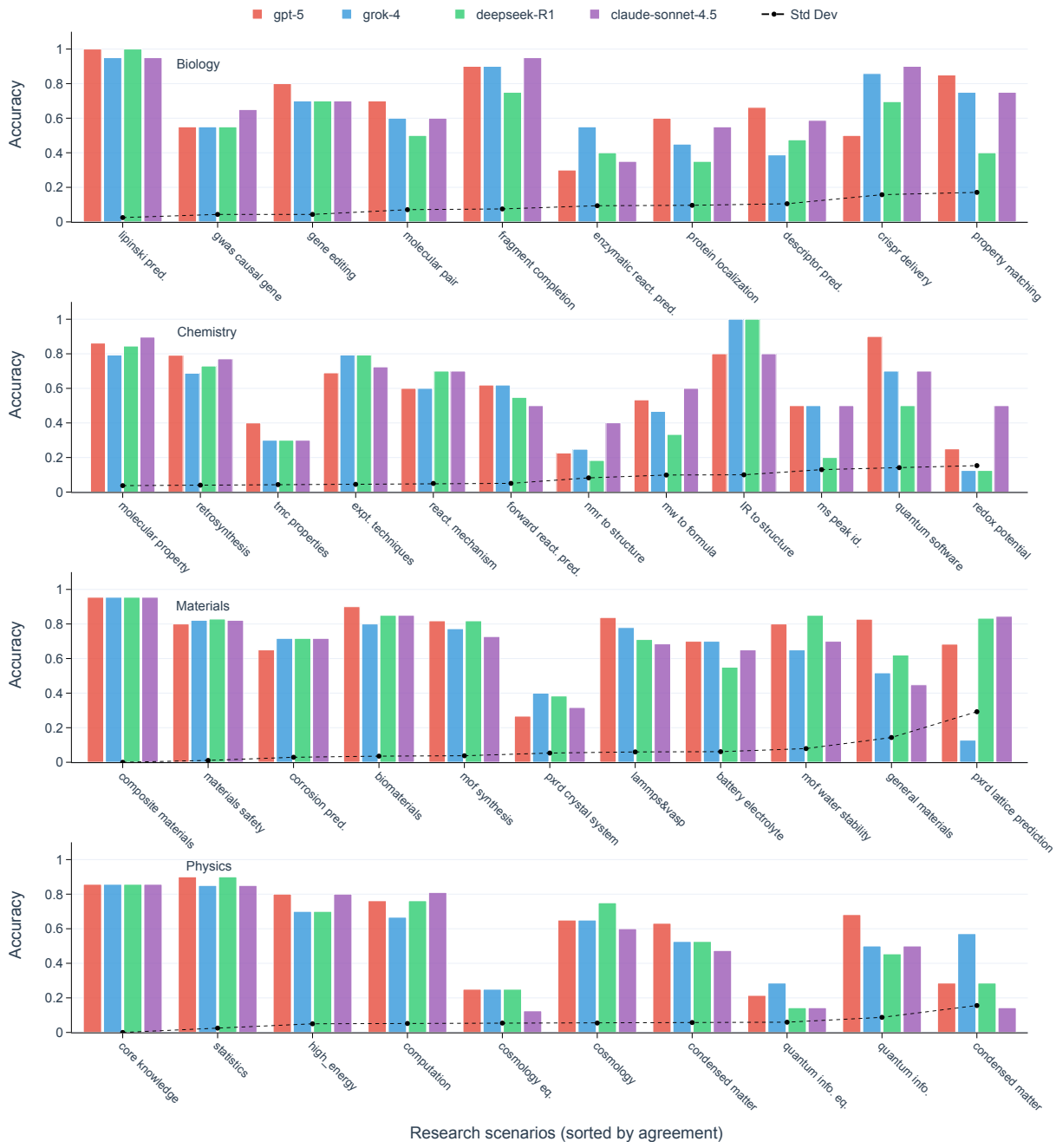

Figure 2. **Per-scenario accuracy for top-performing models at four domains.** `gpt-5` is colored in red, `grok-4` in blue, `deepseek-R1` in green, and `claude-sonnet-4.5` in purple. Research scenarios are ranked with increasing standard deviations of the four model accuracies for each domain, which are shown as the black dashed lines.

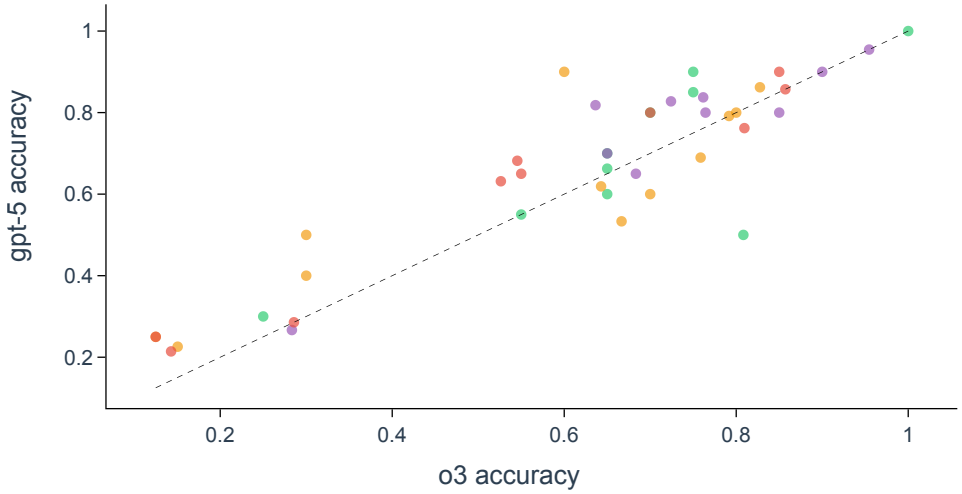

Figure 3. **Per-scenario accuracy for `gpt-5` and `o3`.** Scenarios in biology are colored in green, chemistry in orange, materials in purple, and physics in red. Parity is shown with a black dashed line.

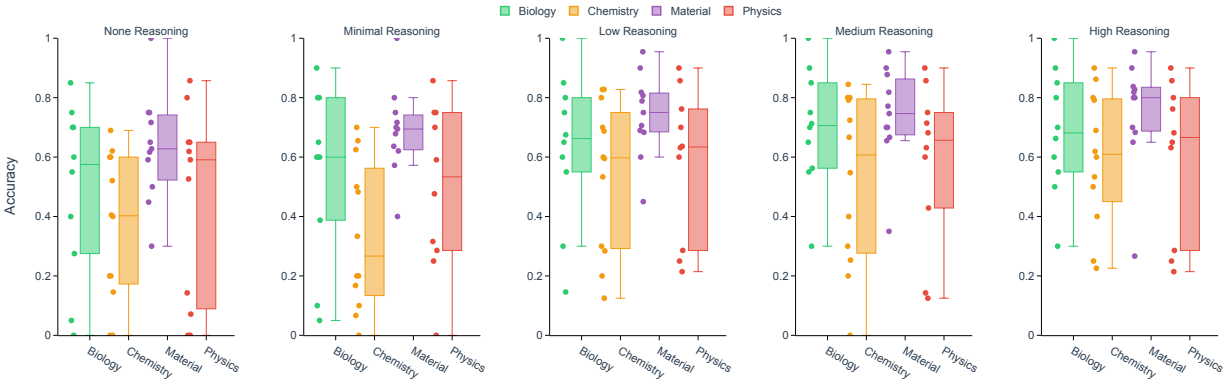

Figure 4. **Accuracy of `gpt-5` at various reasoning levels.** Scenarios in biology are colored in green, chemistry in orange, materials in purple, and physics in red. A box plot is shown for the distribution where all points are explicitly added.

Figure 5. **Pairwise per-scenario accuracy for top-performing models.** Scenarios in biology are colored in green, chemistry in orange, materials in purple, and physics in red. Pearson's r is shown for each pairwise plot, where the overall Pearson's r for each domain is displayed on the legend.

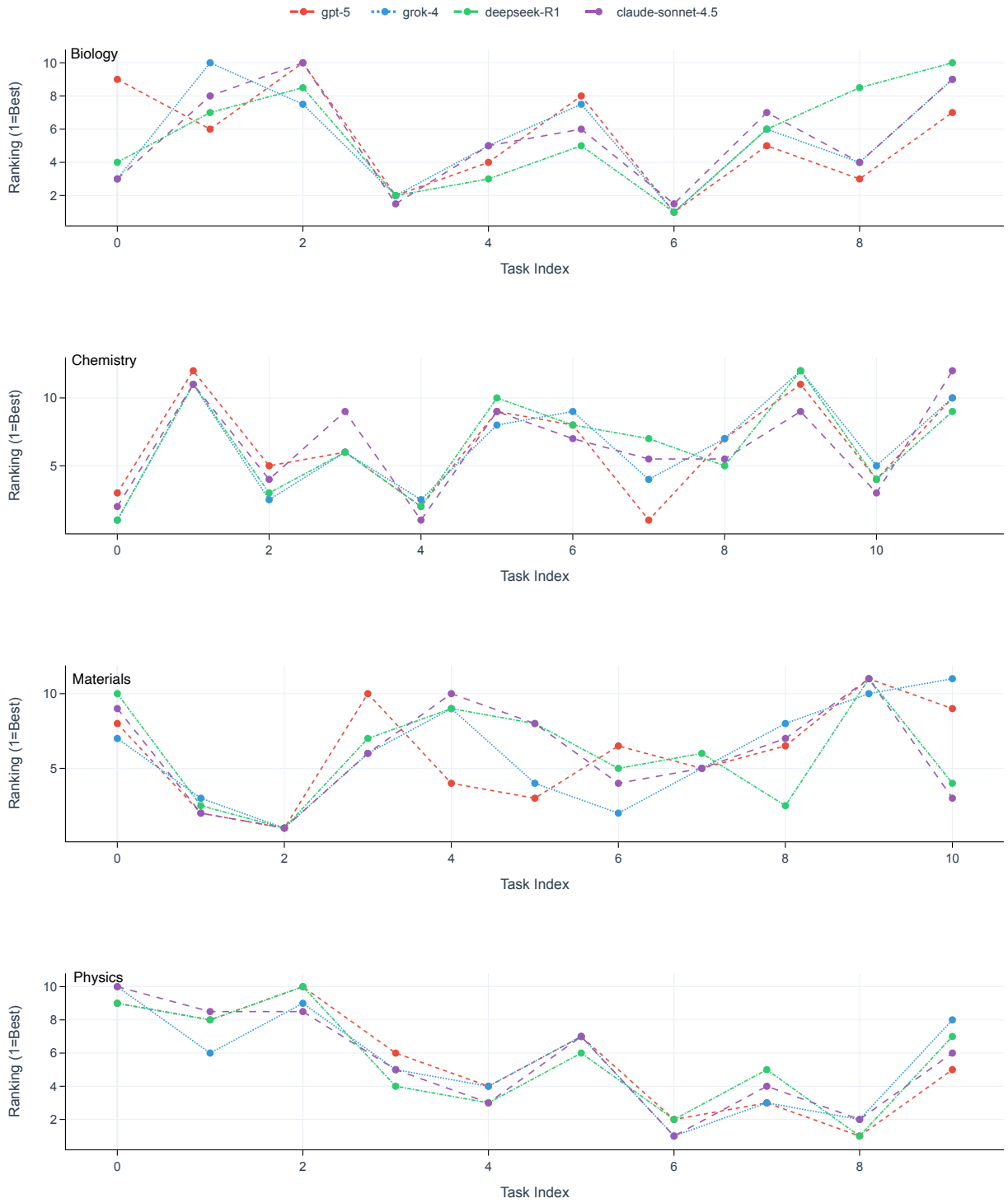

Figure 6. **Ranking vs. research scenario for top-performing models.** `gpt-5` is colored in red, `grok-4` in blue, `deepseek-R1` in green, and `claude-sonnet-4.5` in purple.

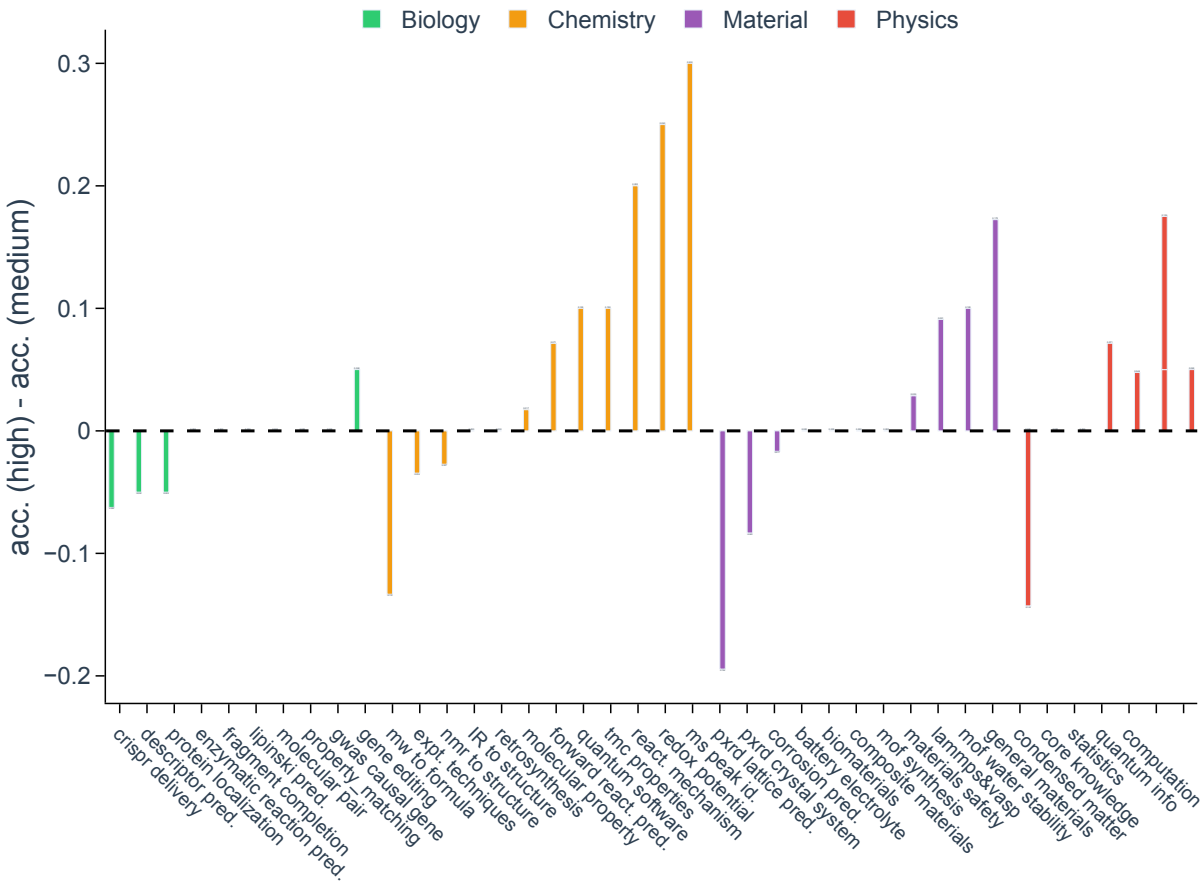

Figure 7. **Performance difference between** `gpt-5` **with high and medium reasoning efforts.** Scenarios in biology are colored in green, chemistry in orange, materials in purple, and physics in red. A dashed line is shown for no accuracy difference.

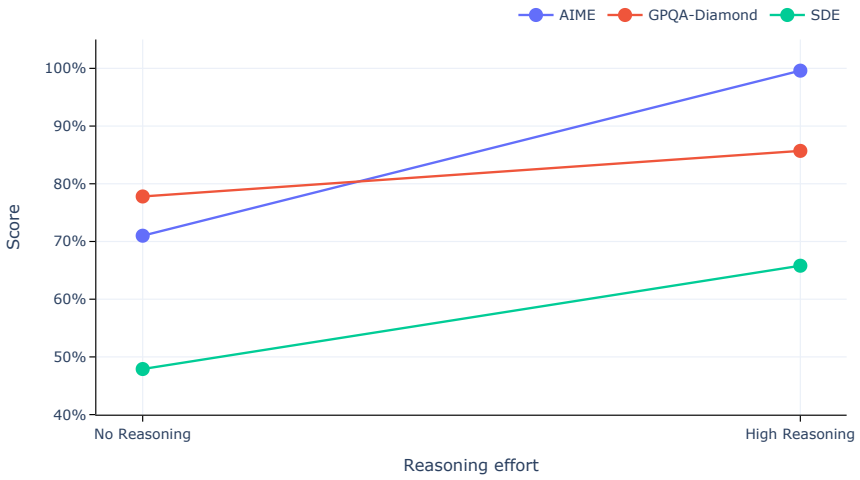

Figure 8. **Reasoning versus no-reasoning performance on three benchmarks for** `gpt-5`**.** AIME 2025 is colored in blue, GPQA-Diamond in red, and SDE question-level evaluation in green.

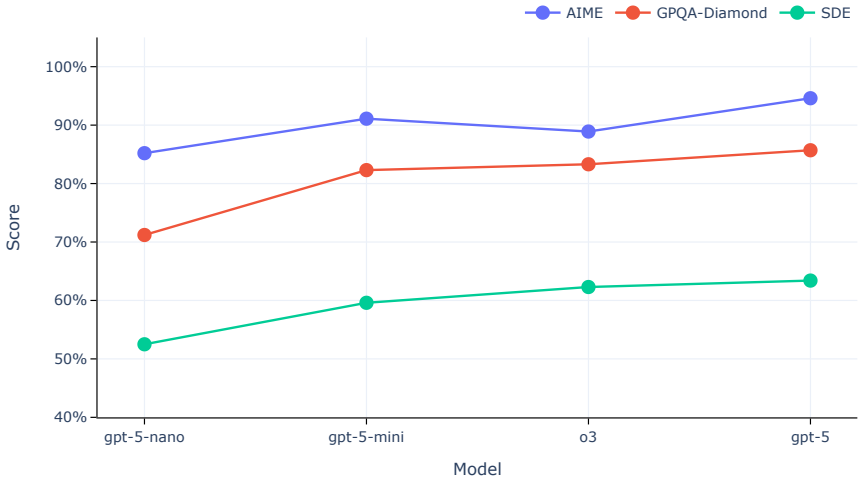

Figure 9. **Model size effects on three benchmarks illustrated by** `gpt-5-nano`**,** `gpt-5-mini`**,** `gpt-5`**, and** `gpt-o3`**.** AIME 2025 is colored in blue, GPQA-Diamond in red, and SDE question-level evaluation in green.

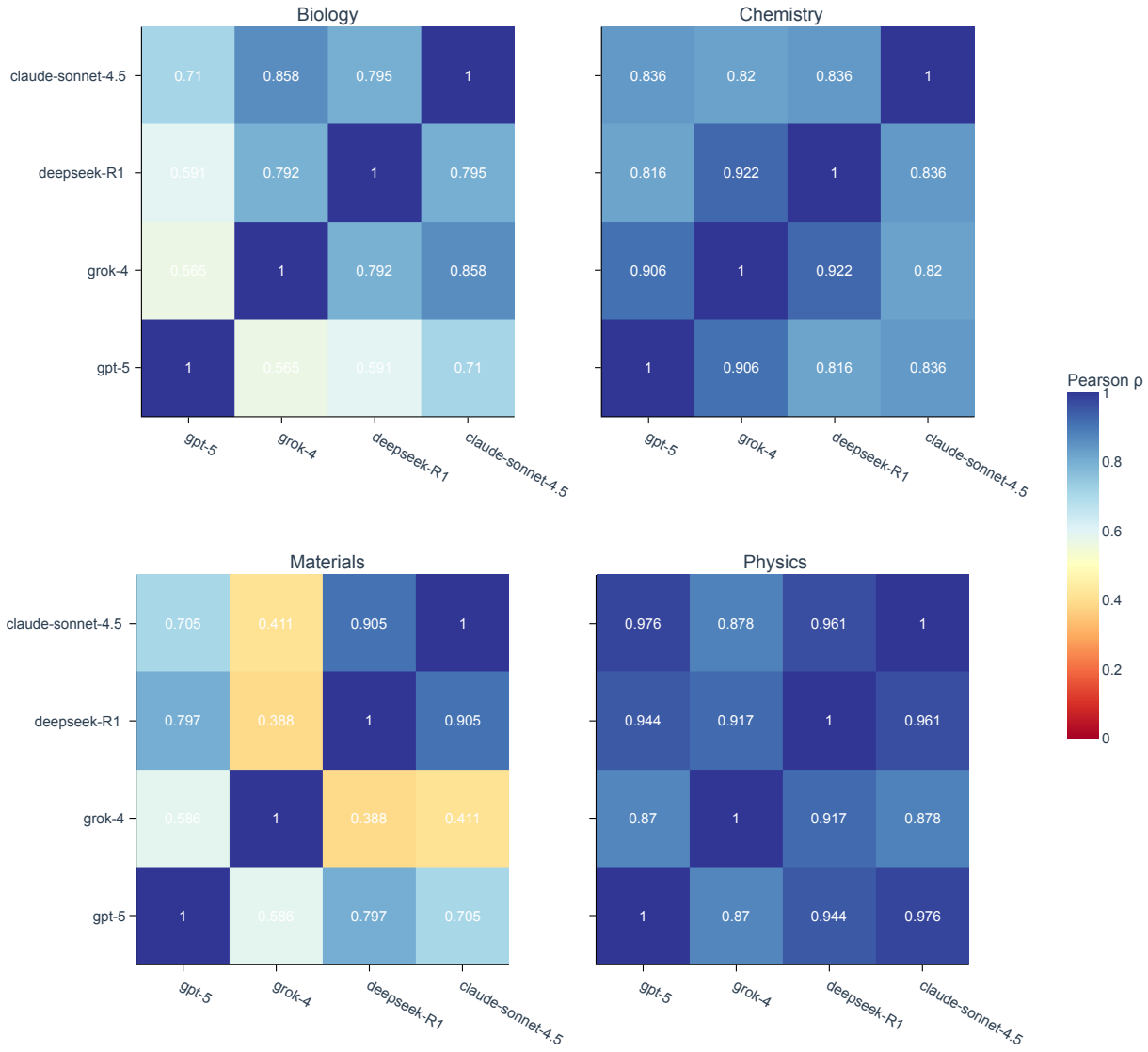

Figure 10. **Pearson correlation heatmaps for top-performing models.** The results are shown by domains.

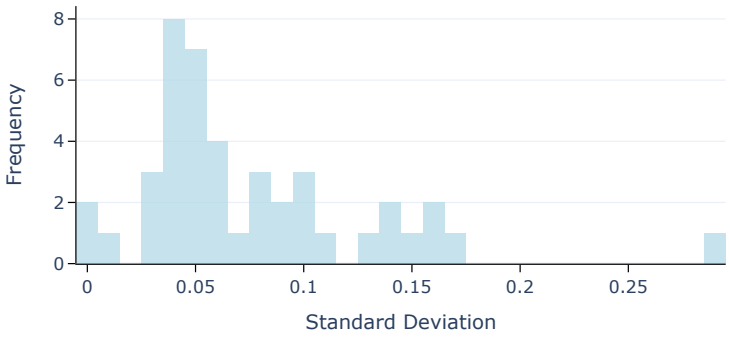

Figure 11. **Distribution of standard deviation for four top-performing models on 46 tasks.** The four top-performing models are `gpt-5`, `grok-4`, `deepseek-R1`, and `claude-sonnet-4.5`.

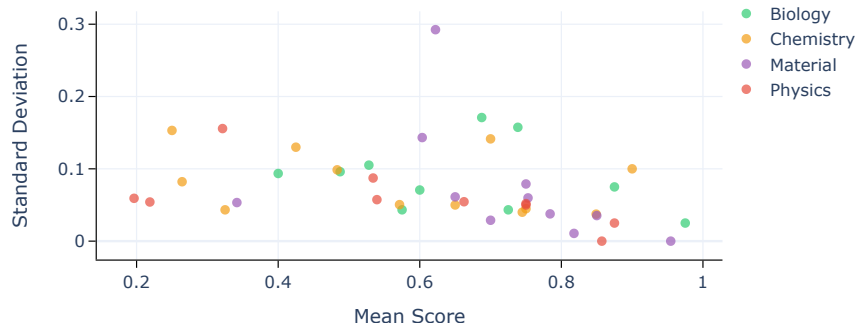

Figure 12. Standard deviation vs. mean value on accuracy for four top-performing models on 46 tasks. The four top-performing models are `gpt-5`, `grok-4`, `deepseek-R1`, and `claude-sonnet-4.5`. Scenarios in biology are colored in green, chemistry in orange, materials in purple, and physics in red.

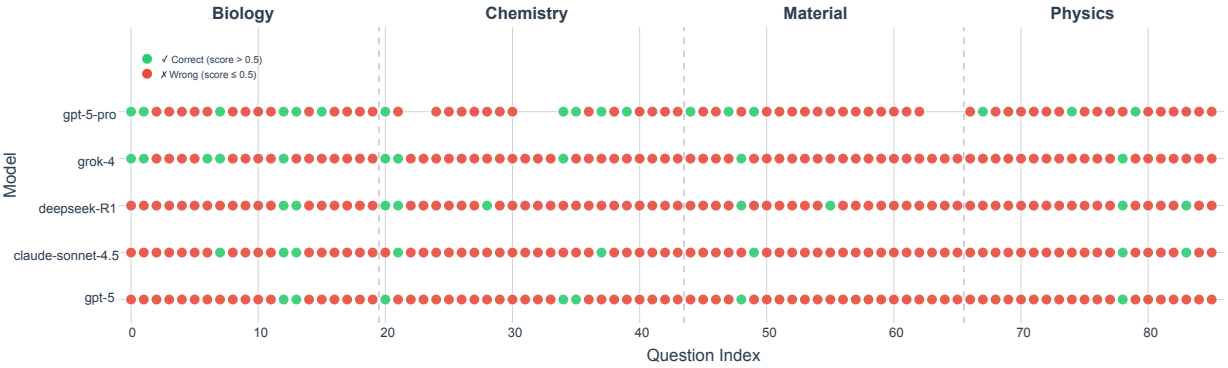

Figure 13. Question-level performance correlation among five models on SDE-HARD. Each question is marked by its correctness, green for correct and red for incorrect.

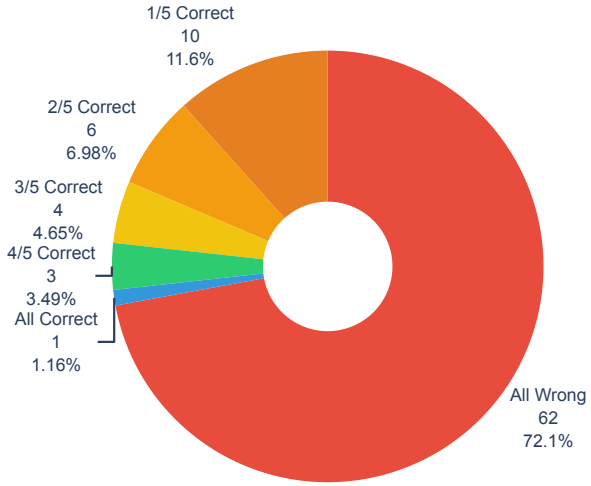

Figure 14. Doughnut plot for analysis of model consensus on SDE-HARD The five models are `gpt-5-pro`, `gpt-5`, `grok-4`, `deepseek-R1`, and `claude-sonnet-4.5`.

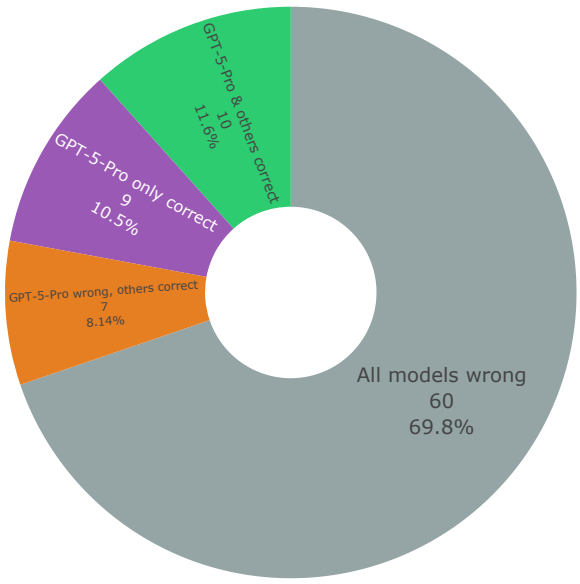

Figure 15. **Doughnut plot for analysis of** `gpt-5-pro` **performance over other four top-performing models**. The four models are `gpt-5`, `grok-4`, `deepseek-R1`, and `claude-sonnet-4.5`.

1 Question curation details

1.1 Question curation for chemistry tasks

This section describes the task-specific curation procedures for the chemistry domain in the SDE benchmark. The chemistry subset comprises 276 questions spanning twelve distinct task types, designed to reflect recurring reasoning patterns in both wet-lab and dry-lab chemical research. Question construction followed a hybrid strategy combining semi-automated generation from public resources with manual expert curation for scenarios lacking structured records. Representative example questions illustrating these task formats are provided in Representative example questions.

Forward reaction prediction (42) Questions for forward reaction prediction were sourced from two parts. A subset was adapted from existing benchmark-style questions (e.g., GPQA), while others were sampled from the USPTO reaction dataset covering diverse reaction classes. Reaction entries were filtered to retain single-step transformations with unambiguous reactant–product mappings. Structured reaction records were then converted into natural-language multiple-choice questions using standardized templates. Additional questions were manually curated based on real-world wet-lab scenarios to capture practical reaction reasoning beyond dataset distributions.

Retrosynthesis (48) Retrosynthesis questions were derived from both benchmark-style sources and template-based sampling from USPTO (USPTO_TPL[80]). For the latter, we used known reaction templates as a reference for what constitutes a chemically reasonable retrosynthetic step. Given a target molecule, we identified which types of bond disconnections are consistent with known reaction pattern. We then formulated multiple-choice questions in which chemically reasonable disconnections served as correct answers, while implausible or poorly motivated disconnections—drawn from unrelated reaction patterns were used as distractors.

Molecular property estimation (58) Molecular property questions were constructed using a mixture of benchmark-derived examples and molecules sampled from the ZINC database[81]. Molecules were represented using SMILES or IUPAC names. Reference properties—including logP, topological polar surface area (TPSA), number of rotatable bonds, ring counts, molecular weight, and Tanimoto similarity—were computed using RDKit[68]. Questions were framed to require comparative or ordering-based reasoning (e.g., ranking molecules by a given property).

NMR-based structure elucidation (31) NMR structure elucidation questions were manually curated from real experimental scenarios and published supplementary information. Each question provides molecular formulae and multi-modal spectroscopic data (^{1}H NMR, ^{13}C NMR, and occasionally MS), requiring models to infer the complete molecular structure. Reference answers are represented as SMILES strings. During evaluation, predicted and reference structures are canonicalized using RDKit and scored via Tanimoto similarity between Morgan fingerprints, allowing partial credit for near-correct structures.

Redox potential estimation (8) Redox potential questions were curated from published electrochemical studies. Given the three-dimensional structure of a photocatalyst or metal complex, models are asked to predict the reduction potential relative to a specified reference electrode. Answers are numeric and evaluated using a task-specific tolerance band (0.25) around the reference value, reflecting experimental uncertainty.

Experimental techniques (29)/Quantum chemistry software usage (10)/Reaction mechanism reasoning (10)/MS peak identification (10)/IR-based structure elucidation (5)/Transition-metal complex property prediction (10)/Mass-to-formula conversion (15) Questions were entirely manually curated by domain experts. These questions were inspired by real world wet-lab/dry-lab experimental procedures or literature. All questions are formulated as either short answers or multiple-choice problems, and answers were evaluated with exact match accuracy.

1.2 Question curation for biology tasks

This section describes the task-specific curation procedures for the biology domain in the SDE benchmark. The biology subset contains 200 questions spanning 10 task types, designed to capture recurring reasoning patterns in biochemistry, cell biology, genetics and molecular biology. Most questions are derived from established biological databases (e.g., ChEMBL, Therapeutics Data Commons, UniProt, GWAS Catalog), supplemented by expert-curated scenarios where structured records are unavailable. Representative example questions are provided in Representative example questions.

Molecular descriptor prediction (20) Descriptor prediction questions assess a model's ability to infer fundamental physicochemical properties directly from molecular structure. Drug-like molecules were sampled with reference labels obtained from CycPeptMPDB and ChEMBL records. Reported values for hydrogen bond donors (HBD), hydrogen bond acceptors (HBA), molecular weight (MW), and logP were cross-checked for consistency with RDKit calculations. Questions are formulated as structured short-answer tasks requiring numerical prediction of multiple descriptors from a given SMILES representation. During evaluation, exact matches are required for integer-valued descriptors (HBD, HBA), while MW and logP are scored using a predefined numerical tolerance to account for minor computational variability.

Lipinski rule assessment (20) Lipinski assessment questions evaluate integrative reasoning over multiple molecular descriptors. Molecules and labels were sampled from CycPeptMPDB and ChEMBL. Given a SMILES string, models are required to determine whether the molecule satisfies Lipinski's Rule of Five by implicitly estimating HBD, HBA, MW, and logP and verifying all threshold conditions. Questions are framed as binary multiple-choice classification tasks and evaluated via exact match accuracy.

Enzymatic reaction prediction (20) Enzymatic reaction prediction questions probe biochemical transformation reasoning under enzyme-specific constraints. Reaction records were sourced from the ECREACT 1.0 dataset, supplemented with authoritative enzyme annotations from IUBMB resources. Each question provides reactant structures, an enzyme EC number, and curated supporting information describing enzyme specificity and cofactors. Models are asked to predict the transformed product in canonical SMILES form, accounting for reaction mechanism and physiological protonation state. Evaluation is performed by exact match comparison between canonicalized predicted and reference SMILES.

Fragment-based molecular completion (20) Fragment completion questions evaluate structure–property reasoning under partial structural information. Masked molecular fragments and associated target property values (e.g., LD50) were sampled from the Therapeutics Data Commons. Models are required to identify which candidate molecule, when completing the masked fragment, best matches the specified property value. Questions are formulated as multiple-choice selection tasks, requiring both structural compatibility and property inference.

Matched molecular pair comparison (20) Matched molecular pair questions test comparative property reasoning between closely related molecules. Property values (LD50 or Vdss) were obtained from the Therapeutics Data Commons. Given two structurally similar molecules, models must predict whether the comparison molecule exhibits a higher or lower property value relative to a reference molecule. This task emphasizes sensitivity to localized structural modifications. Questions are framed as binary classification tasks and evaluated via exact match accuracy.

Property-based molecular matching (20) Property-based matching questions assess similarity-based reasoning across structurally diverse candidates. Given a target molecule with a known property value and a set of candidate molecules, models are asked to identify the candidate whose property value is numerically closest to the target. Property values (LD50 or Vdss) were obtained from the Therapeutics Data Commons. Questions are evaluated via exact match over multiple-choice selections.

Protein subcellular localization prediction (20) Protein localization questions evaluate sequence-based biological inference. Protein sequences and reference localization annotations were derived from UniProtKB/Swiss-Prot, focusing on well-annotated proteins with experimentally supported localization labels. Given an amino acid sequence, models are asked to predict the most likely subcellular compartment (e.g., nucleus, cytoplasm, mitochondrion, secreted), based on known localization signals and sequence features. Questions are formulated as multiple-choice classification tasks and evaluated via exact match accuracy.

GWAS causal gene identification (20) GWAS causal gene questions probe integrative genetic reasoning across phenotype, gene function, and locus context. GWAS loci and associated phenotypes were curated from the NHGRI-EBI GWAS Catalog. For each question, a phenotype and a list of genes within the associated genomic locus are provided. Models are required to identify the gene most likely to be causally related to the phenotype, based on known biological functions, pathways, and disease associations. Answers are evaluated via exact match against expert-curated reference genes.

CRISPR delivery strategy selection (20) CRISPR delivery questions evaluate experimental design reasoning in gene-editing workflows. Scenarios were manually curated by domain experts based on common experimental constraints reported in the CRISPR literature and community best practices. Given a biological context (e.g., editing

primary T cells), models are asked to select the most appropriate and second-most appropriate delivery methods from a predefined list (e.g., viral vectors, electroporation, lipid nanoparticles). Evaluation is performed using a weighted scoring scheme that assigns partial credit based on the relative suitability of the selected delivery methods.

Gene-editing protocol reasoning (20) These questions assess practical reasoning about CRISPR screening and gene-editing protocols. Scenarios were inspired by widely used experimental workflows reported in the functional genomics and CRISPR screening literature. Models are asked to select the most scientifically sound option in situations such as maintaining sgRNA library diversity or designing screening logistics. Questions are framed as multiple-choice tasks and evaluated via exact match accuracy.

1.3 Question curation for physics & math tasks

This section describes the task-specific curation procedures for the physics domain in the SDE benchmark. The physics subset contains 163 questions with 8 task categories, covering astrophysics, quantum information, condensed matter physics, high-energy physics, probability and statistics, computational physics, and core physics knowledge. Unlike domains dominated by large curated datasets, many physics problems involve symbolic manipulation, mathematical equivalence, and conceptual abstraction that resist straightforward data-driven generation, and are primarily manually curated by domain experts, reflecting the fact that many physics reasoning tasks—particularly those involving analytical derivations or conceptual arguments—lack large-scale structured public benchmarks.

Tasks include both multiple-choice and short-answer formats. For short-answer physics questions involving mathematical expressions, symbolic relationships, or derived formulas, answers are evaluated using a symbolic equivalence verification framework. Predicted and reference answers are parsed as mathematical expressions and verified for equivalence up to algebraic transformations with *math_verify* package [82]. This approach ensures that mathematically correct but syntactically different expressions are treated as equivalent, while incorrect derivations are penalized. When symbolic verification is not applicable, a normalized exact-match fallback is used. Representative example questions are provided in Representative example questions.

Astrophysics and cosmology (28) Astrophysics and cosmology questions probe conceptual reasoning and domain knowledge related to galaxy classification, large-scale structure, and cosmological models. Questions were manually written by domain experts and include both multiple-choice conceptual questions and short-answer reasoning tasks.

Quantum information science (36) Quantum information questions assess reasoning about quantum states, entanglement measures, quantum channels, and information-theoretic quantities. Questions are manually curated to reflect reasoning steps commonly encountered in quantum information research rather than rote formula recall. This task category includes both multiple-choice questions and short-answer problems.

Condensed matter physics (26) Condensed matter questions cover statistical mechanics, solid-state physics and quantum many-body systems. Tasks include conceptual multiple-choice questions as well as short-answer derivations.

High-energy physics (20) High-energy physics questions focus on conceptual understanding of quantum field theory and particle physics. These questions are formulated as multiple-choice problems that require recognizing theoretical correspondences rather than recalling isolated facts. All questions in this category are multiple-choice problems.

Probability and statistics (25) Probability and statistics questions assess mathematical reasoning relevant to statistical physics and data analysis. The task set includes both multiple-choice questions and short-answer derivation problems.

Computational physics (21) Computational physics questions evaluate understanding of numerical methods and algorithmic error analysis. Questions are formulated as multiple-choice problems.

Core physics knowledge (7) Core physics questions test foundational operator algebra and elementary physical principles that frequently serve as sub-steps in more complex reasoning tasks. These questions are framed as multiple-choice problems and evaluated via exact match accuracy.

1.4 Question curation for materials tasks

This section describes the task-specific curation procedures for the materials domain in the SDE benchmark. The materials subset contains 540 questions, spanning thirteen distinct task types. These tasks are organized around three key use-cases for LLMs in materials discovery applications: Question-Answering (Q&A), Materials Property Prediction, and Text-Mining.

Q&A-style questions were manually curated by experts and are meant to simulate the types of questions researchers encounter during routine materials discovery workflows. These questions are designed to assess the breadth and coverage of an LLM's foundational materials science knowledge. Property Prediction questions test an LLM's ability to reasonably predict material properties given various textual descriptions and representations of materials as inputs. Lastly, Text-Mining questions evaluate an LLM's ability not only to extract structured materials data from unstructured text, but also to draw conclusions from text with ambiguous or implicitly stated meanings, commonly referred to as argument mining.

General materials science Q&A (29) Basic materials science questions were sourced from expert domain knowledge and first-principle theories and equations. Questions were framed to represent a variety of lab experimental design and result scenarios, with distractor responses formulated based on common misconceptions.

Composite materials Q&A (22) Questions relating to composite materials knowledge were compiled from fundamental mechanics theories and domain knowledge. Lab-scenario and more broad composite material knowledge was constructed into natural-language multiple-choice questions utilizing standard templates, with implausible mechanics of materials and modelling responses as distractors.

Biomaterials Q&A (20) Biomaterials questions were constructed by adapting exam-style scenario-based questions, derived from educational and experimental experience, and supplemented by relevant textbooks and literature where applicable [83–85]. 12 questions were case-study style, in which the correct biomaterial phenomenon to explain the observations, or the correct conclusion based on the biomaterial phenomenon observed, must be selected. 5 questions were design style, in which the correct design choice to meet certain criteria must be selected. 3 questions were basic knowledge recall style questions. All questions were formatted as multiple choice, with questions and answers in natural language, often using acronyms or common names for compounds or phenomena.

Metal-organic framework synthesis Q&A (20) These questions were manually authored by a domain expert and formatted as multiple-choice questions. They cover a range of materials synthesis, characterization, and analysis topics for metal-organic framweworks (MOFs), with incorrect options designed to reflect common misconceptions or plausible alternatives.

Battery electrolyte Q&A (20) These questions were manually authored by a domain expert and formatted as multiple-choice questions focusing on liquid electrolytes for lithium metal batteries. The questions span fundamental concepts and multi-step problems requiring more advanced reasoning, covering Li-ion solvation, solid–electrolyte interphase behavior, electrolyte composition, and performance metrics such as Coulombic efficiency. Incorrect options reflect plausible alternatives or common misunderstandings within the field.

LAMMPS and VASP Q&A (33) These Q&A pairs were manually authored by a domain expert and formatted as multiple-choice questions focusing on atomistic simulation workflows using LAMMPS and VASP. The questions span fundamental software capabilities, simulation setup, and methodological considerations, as well as multi-step reasoning related to interpreting simulation behavior and selecting appropriate computational approaches. Incorrect options reflect plausible alternatives or common misunderstandings in the use of molecular dynamics and electronic-structure tools.

Corrosion prediction of metals against organic solvents (60) Questions for corrosion prediction were constructed using data from the corrosion-survey-database published by NACE International (The Corrosion Society). The dataset comprises experimentally reported corrosion outcomes of organic solvents against a wide range of metallic materials. Organic compounds were represented using SMILES strings, and each question was formulated as a binary classification task indicating whether a given compound exhibits corrosive behavior toward any of 87 metals. Ground-truth labels were assigned based on documented corrosion observations, with positive labels indicating the presence of corrosion and negative labels indicating no observed corrosion. Questions were framed in a true/false format to assess a model's ability to reason about chemical structure–corrosion relationships across diverse organic solvents and metal types.

Polymer glass-transition prediction (24) Questions for polymer glass-transition temperature (Tg) prediction were constructed using data from the NeurIPS Open Polymer Prediction 2025 Kaggle competition[86]. The dataset comprises experimentally reported Tg values for a diverse set of polymer systems. Polymers were represented using structured molecular descriptors provided in the dataset, and each question was formulated as a property prediction task requiring estimation of Tg from the given polymer representation. Ground-truth labels were assigned directly from the reported experimental measurements. Questions were framed to assess a model's ability to reason about polymer structure–property relationships relevant to thermomechanical behavior.

Lattice prediction from PXRD (60) We evaluated lattice parameter predictions using a dataset of ionic compounds from the Materials Project[87] and metal-organic frameworks (MOFs) from the CoRE-2019 database[88]. Each model input was represented by a simulated powder X-ray diffraction (PXRD) pattern, its corresponding Miller indices, and the identified crystal system. To derive these features, we utilized Pymatgen's[89] SpacegroupAnalyzer - a symmetry-determination tool built upon the Spglib[90] library - extracting the crystal system of each structure. The PXRD patterns and associated Miller indices were either simulated using Pymatgen's XRDCalculator[91] under a $CuK\alpha$ radiation source, or already computed in previous works[92]. Ground truth lattice parameters were extracted directly from the source crystallographic information files (CIFs) to serve as the target for our evaluations.

Crystal system prediction from PXRD (60) In this task, we aimed to classify a material's crystal system directly from its observed PXRD peaks. The dataset composition and the underlying methodologies for symmetry determination and PXRD simulation remain consistent with the workflow described in the PXRD Lattice Prediction section. For this classification model, the input features consist of the peak positions (defined by 2θ) and their corresponding relative intensities. We extracted these features using a peak-finding algorithm from the SciPy[93] library, applying a minimum intensity threshold of 5 to ensure the inclusion of only significant diffraction signals.

PXRD denoising (30) This task focuses on denoising a corrupted PXRD pattern to recover its original, "clean" state. The target data consists of the simulated PXRD patterns for ionic compounds, as defined in the PXRD Lattice Prediction section. To prepare the inputs, the original patterns were smoothed using a Gaussian function and discretized into 1D arrays of 200 data points, where each element represents a specific diffraction intensity. As these arrays are standardized to a constant size and uniform 2θ range, explicit 2θ values were omitted to avoid redundancy. Finally, to simulate experimental artifacts, we introduced random Gaussian noise ($\mu = 0, \sigma = 1$) to the normalized intensities. This methodology can be fully described in this work[92].

Materials safety prediction (140) Safety-related questions were constructed using data sampled from the PubChem database, focusing on two hazard categories: inhalation toxicity (80 questions) and flammability (60 questions). Molecules were represented using SMILES strings, and each question was framed as a binary classification task based on Globally Harmonized System (GHS) hazard annotations. Ground-truth labels were assigned according to whether a given molecule was reported to carry the corresponding GHS hazard code, with positive labels indicating the presence of the hazard and negative labels indicating its absence. Questions were presented in a true/false format (e.g., "Does this molecule classify as a flammable liquid according to GHS classification?") to evaluate a model's ability to infer safety-relevant properties directly from molecular structure.

Metal-organic framework water stability text-mining (20) Questions were expertly curated using the experimentally measured water-stability data from the MOF-ChemUnity database [15]. Each question is based on unstructured textual descriptions of reported stability behavior, requiring extraction of relevant evidence and, where necessary, inference of stability outcomes from implicitly stated or qualitative information.

2 Representative example questions

This section provides representative example questions for selected tasks described in previous sections, illustrating the diversity of question formats, reasoning requirements, and evaluation protocols used in SDE.

2.1 Chemistry domain

Retrosynthesis. You are an expert in organic chemistry. The following are multiple choice questions about Retrosynthesis. Let's think step by step. Please wrap the final answer in XML tags: <answer>X</answer> (where X is A, B, C, or D)
Question:
For the retrosynthetic analysis of Nc1cc(Cl)c(Oc2ccc(Cl)c3ccccc23)c(Cl)c1, which transformation would be the poorly strategic choice?
A. ([NH2;D1;+0:1]-[c:2])»(C-C(-C)(-C)-O-C(=O)-[NH;D2;+0:1]-[c:2])
B. ([NH2;D1;+0:1]-[c:2]1:[c:3]:[c:4]:[c:5](-[#8:6]-[c:7]):[c:8]:[c:9]:1)»(O=[N+;H0;D3:1](-[O-])-[c:2]1:[c:3]:[c:4]:[c:5](-[#8:6]-[c:7]):[c:8]:[c:9]:1)
C. ([NH2;D1;+0:1]-[c:2])»(C-C(=O)-[NH;D2;+0:1]-[c:2])
D. ([C:2]-[CH2;D2;+0:1]-[NH;D2;+0:4]-[C:3])»(C-S(=O)(=O)-O-[CH2;D2;+0:1]-[C:2]).([C:3]-[NH2;D1;+0:4])
Answer:

Molecular property estimation. You are an expert in chemistry. The following are multiple choice questions about Molecular Property. Let's think step by step. Please wrap the final answer in XML tags: <answer>X</answer> (where X is A, B, C, or D)
Question:
Based on number of rotatable bonds, which arrangement represents these molecules in ascending order?
1. CC1CC(C(=O)O1)NC(=O)C2=CC=CC=C2
2. CCOC(=O)C1=NOC(=C1C(=O)OC)C
3. CCC1=CC2=C(S1)C(=O)N3CCOC3=N2
4. methyl 1,3-dimethyl-7,8,9,10,11,12-hexahydrocycloocta[a]indolizine-6-carboxylate
A. Molecule 2 < Molecule 1 < Molecule 3 = Molecule 4
B. Molecule 1 < Molecule 2 < Molecule 4 = Molecule 3
C. Molecule 1 < Molecule 4 < Molecule 2 < Molecule 3
D. Molecule 3 = Molecule 4 < Molecule 1 < Molecule 2
Answer:

Experimental techniques. You are an expert in experimental chemistry. The following are multiple choice questions about Laboratory Techniques. Let's think step by step. Please wrap the final answer in XML tags: <answer>X</answer> (where X is A, B, C, or D)
Question:
During a methylation reaction using MeI (1.2 equiv.) under ice bath conditions (0°C), you observe poor regioselectivity between competing nucleophilic sites. Which approach would most likely improve the selectivity?
A. Reduce MeI to 1 equivalents to minimize over-alkylation
B. Remove the ice bath and run at room temperature
C. Use an ice-salt bath to reach -15°C
D. Switch to a dry ice-acetone bath (-78°C)
Answer:

NMR-based structure elucidation. You are a chemist assistant with expertise in molecular structure elucidation. Given the following spectroscopic data for an unknown compound: Molecular Formula: C13H12OSe. 1H NMR (CDCl3, 400 MHz): δ 7.50 (d, J = 8.8 Hz, 2H); 7.33-7.31 (m, 2H); 7.21-7.16 (m, 3H); 6.84 (d, J = 8.4 Hz, 2H); 3.79 (s, 3H). 13C NMR (CDCl3 100 MHz); δ (ppm): 159.7, 136.5, 133.2, 130.9, 129.1, 126.4, 119.9, 115.1, 55.2. MS (relative intensity) m/z: 264 (65), 262 (34), 184 (100), 153 (32), 65 (14). Determine the complete molecular structure.
Requirements:
- Provide the structure in SMILES notation
- The answer should be exact and canonical
- Include only the SMILES string in your answer, wrapped in <SMILES></SMILES> tags
Example answer format:
<SMILES>CCN(CC)CC</SMILES>
Answer:

2.2 Biology Domain

Lipinski rule assessment. You are an expert biochemist specializing in drug-like property analysis. Your task is to analyze a molecule's SMILES representation and determine whether it satisfies Lipinski's Rule of Five.
Given Information:
- Input SMILES: CC(C)C[C@@H]1NC(=O)[C@@H](CC(C)C)NC(=O)[C@@H](C)NC(=O)[C@@H]2CCCN2C(=O)[C@H](Cc2ccccc2)NC(=O)[C@@H](CC(C)C)N(C)C(=O)[C@@H]2CCCN2C1=O
Lipinski's Rule of Five Criteria:
A molecule satisfies the rule if ALL of the following conditions are met:
- Hydrogen Bond Donors (HBD) $\leq$ 5
- Hydrogen Bond Acceptors (HBA) $\leq$ 10
- Molecular Weight (MW) < 500 Da
- LogP < 5
Task: Calculate the four properties and determine if the molecule satisfies all criteria.
Options:
A: Satisfies Lipinski's Rule of Five (meets all criteria)
B: Violates Lipinski's Rule of Five (fails one or more criteria)
Requirements:
- Calculate all four molecular properties from the SMILES
- Check if ALL criteria are satisfied - Provide only the letter of your answer
- Do not include explanations or reasoning
- Wrap your answer in <ANSWER></ANSWER> tags
Output Format: <ANSWER>[Single letter: A or B]</ANSWER>

GWAS causal gene identification. You are an expert in genetics and GWAS analysis. Your task is to identify the most likely causal gene within a genomic locus for a given GWAS phenotype. Given Information:
GWAS Phenotype: Multi-trait sex score Genes in Locus: APOC1, APOC2, APOC4, APOC4-APOC2, APOE, BCAM, BCL3, BLOC1S3, CBLC, CEACAM16, CEACAM19, CEACAM20, CKM, CLASRP, CLPTM1, ENSG00000267173, ERCC2, EXOC3L2, GEMIN7, IGSF23, KLC3, MARK4, NECTIN2, NKPD1, PPP1R13L, PPP1R37, PVR, RELB, TOMM40, TRAPPC6A, ZNF180, ZNF229, ZNF285, ZNF296
Requirements:
Analyze the biological relationship between the phenotype and each gene in the locus Consider known gene functions, pathways, and disease associations Select the gene most likely to be causally related to the phenotype Choose only from the genes provided in the list Provide only the gene symbol as your answer
Output Format: <ANSWER>[gene symbol from the provided list]</ANSWER>

CRISPR delivery strategy selection. You are an expert in CRISPR gene editing technologies. Your task is to identify the two most relevant CRISPR delivery methods based on the user's requirements.
Given Information:
- User Requirements: I hope to edit primary T-cell
Available Delivery Methods:
a. Plasmid Transfection
b. Lentivirus/Retrovirus
c. RNP/mRNA electroporation
d. RNP/mRNA microinjection
e. mRNA LNP
f. AAV
Requirements:
- Analyze the user's needs and experimental context
- Select the most appropriate delivery method as Answer1
- Select the second most appropriate delivery method as Answer2
- Provide only single letters (a-f) as answers
- Wrap your answer in <ANSWER></ANSWER> tags
Output Format: <ANSWER>[Answer 1: single letter: a, b, c, d, e, or f], [Answer 2: single letter: a, b, c, d, e, or f]</ANSWER>

2.3 Physics domain

Astrophysics and cosmology. You are an expert in Astrophysics and Cosmology. The following are multiple choice questions about astrophysics and cosmology. Let's think step by step. Please wrap the final answer in XML tags: <answer>X</answer> (where X can be A, B, C, D, or combinations like AB, ABC, etc.)
Question:
If we are living in a universe with primordial black holes as the only dark matter candidate (i.e. $\Omega_{DM} = \Omega_{PBH}$), and all of them are centered around some mass M_{PBH}, with number density given by n_{PBH}. What is the dominant change (Poisson fluctuation) induced by PBHs to the linear power spectrum $P(k)$ related to the number density of PBHs n_{PBH} (only show the proportional relationship)? Derive this proportionality from the cosmic over-density field
Answer:

Quantum information science. You are an expert in Quantum Information Theory. Please solve the following question step by step.

Question. Consider a tripartite state $\boxed{\rho_{ABC}}$ consisting of d-dimensional qudits. Let A, B, C contain N_A, N_B, and N_C qubits, respectively. Suppose there exist a channel R acting on the Hilbert space and outputing to the Hilbert space of AB such that

$$\|R[\rho_{BC}] - \rho_{ABC}\|_1 = \epsilon.$$

One can show that the conditional mutual information

$$\boxed{I(A : C \mid B) \leq c\sqrt{\epsilon}}.$$

How does the prefactor c depend on d, N_A, N_B, and N_C? You can throw away all the constants. Provide your reasoning first, then conclude with your final answer.
IMPORTANT: You MUST put your final mathematical answer in this exact format:

Final Answer: $\boxed{\text{your mathematical expression}}$

Do not use any other format for your final answer.
Answer:

Probability and statistics. You are an expert in Probability and Statistics. The following are multiple choice questions about probability and statistics. Let's think step by step. Please wrap the final answer in XML tags: <answer>X</answer> (where X can be A, B, C, D, or combinations like AB, ABC, etc.) Question:
Which of the concentration inequalities below are based on moment generating function?
A: Chebyshev
B: Chernoff
C: Azuma-Hoeffding
D: Bernstein
Answer:

2.4 Materials Domain

MOF synthesis Q&A. You are an expert in MOF synthesis. Select the correct answer from the choices below.

Wrap your final answer in <answer> tags, do not include any additional text or explanations in your response. Only answer with the letter corresponding to the correct option, for example: <answer>A</answer>

A research team is synthesizing an aluminum-based MOF using a 4,4',4"-Benzene-1,3,5-triyl-tris-(benzoate) linker, denoted as BTB, and explores different parameters to maximize both yield and crystal size. They find that at temperatures below 120°C, yield is < 15%, while at 140°C, yield is 80%. A BTB:Al ratio of 3:4 increases yield by 10% over 2:1. Using a modulator of formic acid:water at 1:1 doubles the crystal size without changing yield, while a 4:1 ratio decreases crystal size by 30%. Reaction times > 72h do not affect yield or size. Which of the following recipes maximizes both yield and crystal size?

A. 2:1 BTB:Al, pure formic acid, 140°C, 72h
B. 3:4 BTB:Al, formic acid:water 1:1, 140°C, 72h
C. 3:4 BTB:Al, formic acid:water 4:1, 140°C, 72h
D. 4:1 BTB:Al, pure formic acid, 130°C, 72h
E. 3:4 BTB:Al, pure formic acid, 120°C, 96h
Answer:

Crystal system prediction from PXRD You are an expert crystallographer. Your task is to classify the crystal system of the following material based on its PXRD data.

Material type: Ionic Compound
Peak positions: 25.235, 41.792, 49.462, 60.609, 66.758, 76.329, 76.393
Peak intensities: 100.0, 48.161, 31.473, 15.599, 6.261, 10.34, 7.737

Answer:

Metal-organic framework water stability text-mining You are an expert in metal-organic frameworks. You will be given a passage of text from a paper, and must use it to decide the MOF's water stability. Note, the passage may not contain enough information, or contain relevant information, needed to answer the question. Wrap the final answer in <answer>X</answer> tags (X = A, B, or C).

Based off this passage, identify if the MOF called "MOF-805" is stable or unstable in water. Note, there may not be enough information to provide an answer:

"The cycle performance results show that, for MOF-805, MOF-806, MOF-808, and Basolite A100, A300, and C300, the uptake constantly drops in every cycle. The surface area of these MOFs was redetermined after the water cycle tests, showing a significant decrease. This observation suggests that the loss of water uptake capacity is related to the loss of porosity."

A: Stable
B: Unstable
C: Not enough information
Answer:

3 Detailed discussion on research project experiments

3.1 Retrosynthesis pathway design

The workflow follows an `Initialization` and `Mutation` phase which we detail below. The original implementation is from Wang et al.[71]

Initialization. Each experiment began by computing the Tanimoto similarity between the target molecule to a list of reference molecules using Morgan fingerprints with radius 2. These reference molecules are from the training and validation sets of Chen et al.[69], which in turn are extracted from the United States Patent and Trademark Office (USPTO)[94]. Three reference synthesis routes are retrieved with probability proportional to the Tanimoto similarity.

The `Initialization` phase tasks the LLM with proposing an initial synthesis route to the target molecule, given the reference routes as context. The proposed routes are assessed based on *validity* which is comprised of the following:

1. **Molecule validity**: all molecules are RDKit parsable and the final precursors are commercially available (here, the eMolecules commercially available building blocks stock from Chen et al.[69] is used).
2. **Reaction validity**: all LLM-proposed reaction templates have an exact match in the reaction database (here, the reaction database is derived from `USPTO-Full` extracted from Yu et al.[95]).
3. **Route validity**: The LLM proposes a route following the desired data format. If the first step of the proposed route fails, the LLM is prompted to propose another route.

In the case that an LLM-proposed reaction template does not exist in the reaction database, a difference fingerprint (RDKit's CreateDifferenceFingerprintForReaction) is created for the proposed reaction and similar reactions are retrieved from the reference USPTO database. These retrieved reactions are applied and the first one that results in valid reactants is selected. If no valid reaction is found, then a reaction template is selected based on Tanimoto similarity to the target molecules present in the USPTO database. Following the original work, 10 total initial valid routes are generated.

Mutation. The `Mutation` phase refines the population of initial routes. One parent route is selected and the non-purchasable intermediate molecules are identified (e.g. molecules that are a result of applying a reaction template but are not commercially available so must be decomposed further). Using the same Tanimoto similarity comparison to the USPTO reference database, routes of similar targets are retrieved. The LLM is then tasked to propose a modified route given these reference routes and feedback on current problems in the parent route (which is one of the initial routes generated in the initialization phase). If the LLM successfully proposes valid steps in the synthesis route, these steps are appended to the parent route and the full route (so far). This "full" route may still have problems which can then be iterated on in the next mutation trial. All routes in the population are scored with a combination of the synthetic complexity (SC)[96] and synthetic accessibility (SA)[97] scores for the non-purchasable intermediates. The 10-best scoring routes are kept before beginning a new mutation round. The entire search process terminates when either a valid synthesis pathway is found or the LLM budget is exhausted (in this work, we allow 100 LLM queries), whichever occurs first. The evaluation runs the LLM framework on prescribed sets of target molecules and reports the number of molecules that return a valid synthesis route, within 100 LLM queries (i.e. the solve rate).

Results. Using the `Pistachio Hard`[98] benchmark set of 100 molecule targets, we evaluated the solve rate using multiple LLMs and compared to common methods spanning MCTS[99] and Retro*[69] search Table 1. Overall, the LLMs' performance is competitive with baselines with an explicit search algorithm. The results show the feasibility of using the LLM itself for retrosynthetic planning, replacing the search algorithm. We note however, that the results were only run for one replicate due to computational cost (newer LLMs are notably more expensive than its predecessors). Interestingly, GPT-4o outperformed all newer models which empirically struggled more to generate valid routes. Specific failure modes include outputting routes that do not conform to the desired route data format and/or violating the molecule or reaction validity checks.

Table 1. Retrosynthesis solve rate comparison on `Pistachio Hard` (100 targets). Following the original work, `temperature = 0.7`, if permitted. For `gpt-5` and `gpt-5-chat`, only `temperature = 1.0` is permitted. For `gpt-4o` and `gpt-5-chat`, `max_tokens = 16384`. Otherwise, `max_tokens = 32768`

Method	**Pistachio Hard Solve Rate (%)**
Graph2Edits[100] (MCTS)	26.0
RootAligned[101] (MCTS)	83.0
LocalRetro[102] (MCTS)	52.0
Graph2Edits (Retro*)	71.0
RootAligned (Retro*)	78.0
LocalRetro (Retro*)	63.0
`gpt-4o`	60.0
`gpt-5-chat`	49.0
`gpt-5`	53.0
`claude-sonnet-4.5`	53.0
`deepseek-R1`	42.0

gpt-4o Outperforms Newer LLMs. Table 1 shows that `gpt-4o` consistently outperforms newer LLMs, particularly its successors, `gpt-5` and `gpt-5-chat`, on the `Pistachio Hard` set of targets. We investigate the failure modes of all compared LLMs by analyzing the frequency in which an LLM-proposed reaction step encounters specific errors (Table 2). We make the following observations for the most apparent differences amongst the LLMs: (1) `gpt-5` and `claude-sonnet-4-5` propose routes that have more steps, before either solving the molecule or encountering an error (*Steps Per Route Attempt*). (2) A notable failure mode for `gpt-5`, `gpt-5-chat`, and `deepseek-reasoner` is generating invalid SMILES. Note that this does not necessarily mean that these models are worse than `gpt-4o` and `claude-sonnet-4-5` at generating valid SMILES, but rather, worse at generating valid SMILES in the context of the synthesis route data structure. The two types of invalid SMILES errors occur in the *Updated Set*: After applying a reaction template, the LLM adds invalid reactant SMILES to the molecule set and in the *Mol Set*: The LLM tries to decompose a SMILES that is invalid (this could have been inherited from the previous step). (3) Reasoning can improve SMILES validity in the route context. The most relevant comparison is `gpt-5` and `gpt-5-chat` where the former is a reasoning model and has notably lower invalid SMILES rates. We additionally highlight that `claude-sonnet-4-5` has a particularly low invalid SMILES rate, but has a notably higher *Updated Set Mismatch* error. This error means that after applying a reaction template, the LLM incorrectly generated the *updated molecule set*, which should equal the original set plus the new reactants, minus the decomposed product.

Table 2. LLM failure modes on `Pistachio Hard` (100 targets).

Metric	`gpt-4o`	`gpt-5`	`gpt-5-chat`	`claude-sonnet-4-5`	`deepseek-reasoner`
Absolute Count Metrics					
Total Solved Routes	60	53	49	53	42
Total Route Attempts	4630	4701	3729	3392	4824
Total Steps	12674	17283	10815	11825	15343
Steps Per Route Attempt	2.74	3.68	2.90	3.49	3.18
Percentage (%) of Total Steps Metrics					
Valid Steps	36.77	28.78	34.83	31.97	32.46
Invalid SMILES Errors					
In the Updated Set	16.47	21.53	41.85	4.96	24.27
In the Mol Set	6.75	13.98	19.57	3.32	13.22
Invalid Reaction Errors					
Reaction not in Database	54.23	58.99	59.57	48.68	56.16
Reaction Exists but Template Fails	0.43	0.55	0.42	0.82	0.68
Set Errors					
Updated Set Mismatch	8.57	11.68	5.18	18.53	10.70
Selected Mol to Decompose not in Set	36.63	28.06	31.79	31.09	30.52

3.2 Molecule optimization

We evaluated the molecule optimization task targeting two objectives, *jnk3* and *gsk3*β. Each experiment began with an initial population of 120 molecules sampled from the ZINC dataset[81]. In each generation, two parent molecules were drawn from the current population with probability proportional to their fitness. For LLM-based methods, mutation and crossover were implemented by prompting the model with one or two parent molecules and asking it to propose a new molecule, either by mutating a single parent or recombining both. This procedure was repeated until 70 offspring were generated. All offspring were evaluated by the oracle, and the union of parents and offspring was re-ranked by fitness; the top-120 molecules were retained as the population for the next generation. We capped the total number of oracle calls at 10,000 and applied early stopping: if the mean fitness of the top-100 molecules failed to improve by at least 10^{-3} over 5 consecutive generations, the run was terminated. Methods were compared using the area under the curve of the top-k average objective versus the number of oracle calls ($\text{AUC}_{\text{top-}k}$) with $k = 10$, which jointly captures optimization quality and sample efficiency.

The quantitative results for both objectives are reported in Table 3. For non-LLM baselines, Graph GA is consistently weaker than learning-based approaches, especially on *jnk3*, while REINVENT provides a strong and stable reference across tasks. This highlights the advantage of specialized molecular generative frameworks for property-driven optimization.

Among LLM-based methods, GPT-4o performs competitively on both objectives, matching REINVENT on *jnk3* and remaining close on *gsk3*β. Claude Sonnet 4.5 achieves the best overall performance across both tasks, with the highest AUC Top 10 on *jnk3* and a particularly strong result on *gsk3*β, suggesting that both optimization efficiency and final top-k quality benefit from strong generative priors and effective exploration. DeepSeek-R1 remains competitive but shows a larger gap on *gsk3*β in terms of Avg Top 10, indicating that maintaining high-quality top candidates may be more sensitive to model-specific generation behavior for this objective.

In contrast, GPT-5 and GPT-5-chat underperform on *jnk3*, showing noticeable drops in both AUC Top 10 and Avg Top 10. On *gsk3*β, they improve substantially (e.g., GPT-5 Avg Top 10 0.942), but still lag behind the best-performing models in sample efficiency. Consistent with our empirical observations, GPT-5 tends to propose molecules with a higher duplication rate, which reduces effective exploration of chemical space and can disproportionately hurt AUC-based metrics, even when the final top-k set is reasonably strong. This comparison is also affected by an unavoidable decoding mismatch: GPT-5 and GPT-5-chat can only be run with `temperature = 1.0`, whereas all other LLMs are evaluated with `temperature = 0.8`. As a result, the observed gaps may reflect a combination of model behavior and sampling differences, rather than the reasoning-oriented design of GPT-5 alone.

Table 3. Performance comparison of different models on molecule optimization tasks (jnk3, gsk3β, sitagliptin_mpo). All LLM models were tested with temperature = 0.8 and max_tokens = 8192 except for GPT-5 and GPT-5-chat.

Method	**jnk3**		**gsk3**β		**sitagliptin_mpo**	
	AUC Top 10	**Avg Top 10**	**AUC Top 10**	**Avg Top 10**	**AUC Top 10**	**Avg Top 10**
Graph GA	0.548	0.890	0.779	0.945	0.425	0.475
REINVENT	0.794	0.912	0.868	0.978	0.434	0.508
`gpt-4o`	0.796	0.932	0.857	0.972	0.572	0.633
`gpt-5-chat`	0.717	0.803	0.832	0.887	0.453	0.478
`gpt-5`	0.695	0.752	0.822	0.942	0.553	0.765
`claude-sonnet-4.5`	0.865	0.935	0.981	1.00	0.422	0.456
`deepseek-R1`	0.749	0.932	0.849	0.893	0.575	0.608

3.3 Transition metal complex optimization

In each iteration, a prompt is meticulously crafted, comprising generic instructions alongside specific information, constraints, and objectives, and is then presented to an LLM to be evaluated. The reference data as the input into the model through carefully constructed prompts containing: (1) a pool of 50 ligands represented by their SMILES strings, IDs, charges, and connecting atom information; (2) 20 randomly sampled initial TMCs from a space of 1.37M possible Pd(II) square planar complexes, along with their pre-calculated properties (HOMO-LUMO gap and polarisability); and (3) natural language descriptions of the design objectives (e.g., maximizing HOMO-LUMO gap). The LLM subsequently proposes a new set of TMCs, which undergo a rigorous validation process including charge constraints (-1, 0, or +1), structure generation using molSimplify[103], geometry optimization with GFN2-xTB, and connectivity validation to ensure no unintended bond rearrangements occur. Valid TMCs and their calculated properties are integrated into the prompt for the subsequent iteration, effectively completing the scientific discovery loop. Specifically, we start with 20 initial TMCs, with the LLM proposing 10 new TMCs at each iteration until a maximum iteration of 20 is reached. All TMCs explored during the optimization process, regardless of their fitness values, are included in the prompt for the next iteration. This mimics the human learning experience where one can learn from both good and bad examples. To prevent the LLM from overemphasizing specific records, the historical TMC data within the prompt is randomly shuffled before each iteration. To minimize bias from the initial TMC sampling, five random seeds are used to sample the initial known TMCs.

In the first task of proposing TMCs with maximized polarisability, `gpt-5`, `deepseek-R1`, and `claude-sonnet-4.5` successfully finds the optimal solution in the space of 1.37M TMCs at all five random seeds (Fig. 16). On the contrary, `gpt-5-chat-latest` fails to do so in all five random seeds, showing the significance of reasoning capability in TMC optimization. Across three reasoning models, `claude-sonnet-4.5` demonstrates quicker convergence through iteration compared to `gpt-5` and `deepseek-R1` consistently. A similar trend is observed when models are asked to expand the Pareto frontiers by proposing TMCs (Fig. 19). There, `gpt-5-chat-latest` still finds the most limited Pareto frontiers, hardly identifying TMCs with polarsability > 400 a.u. and HOMO-LUMO gap > 4 eV. Out of the three reasoning models, `deepseek-R1` gives the most expanded and balanced Pareto frontiers. It should be noted that the influence of random seed is significant, which leads to different sets of TMCs found as Pareto frontiers. `claude-sonnet-4.5` is impacted the most by random seeds, and only explore TMCs locally at seed 68, 86, and 1234. `gpt-5`, despite not yielding the best combined Pareto frontiers across all models, follows the instruction clearly and attempts a balanced exploration at all random seeds.

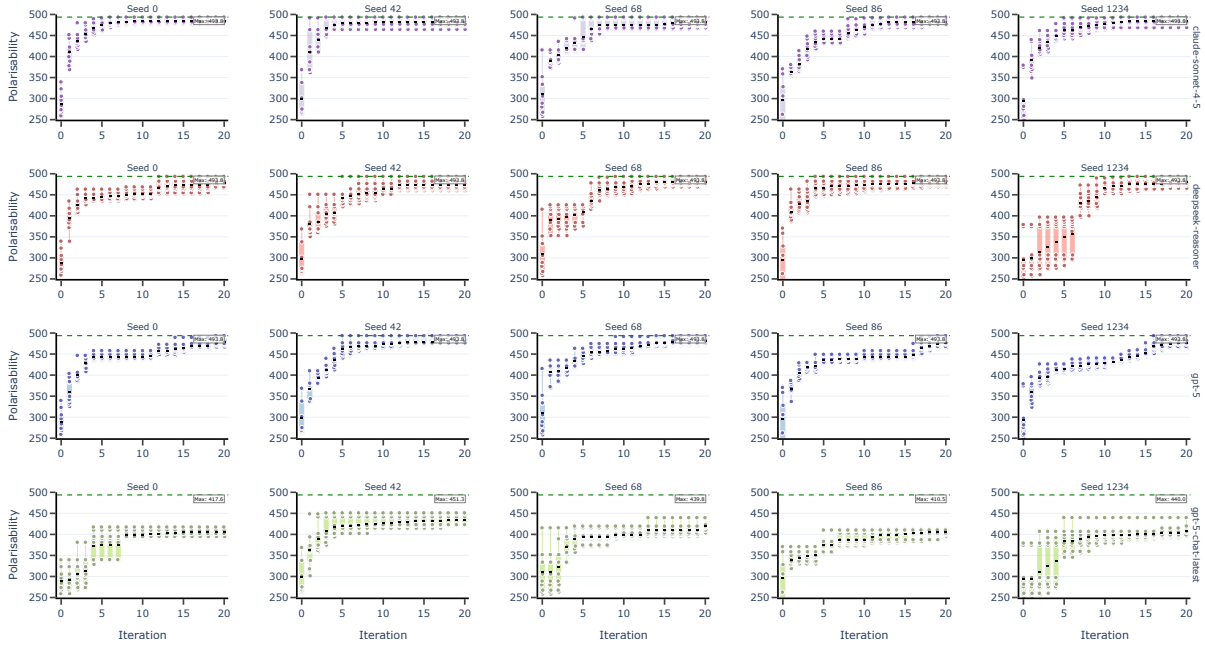

Figure 16. Distribution of top-10 unique TMCs by iterations on maximizing polarisability. Each model is evaluated on five different random seeds and shown in different columns. Results from different models are displayed at different rows. `gpt-5-chat-latest` is colored in green, `gpt-5` in blue, `deepseek-R1` in red, and `claude-sonnet-4.5` in purple.

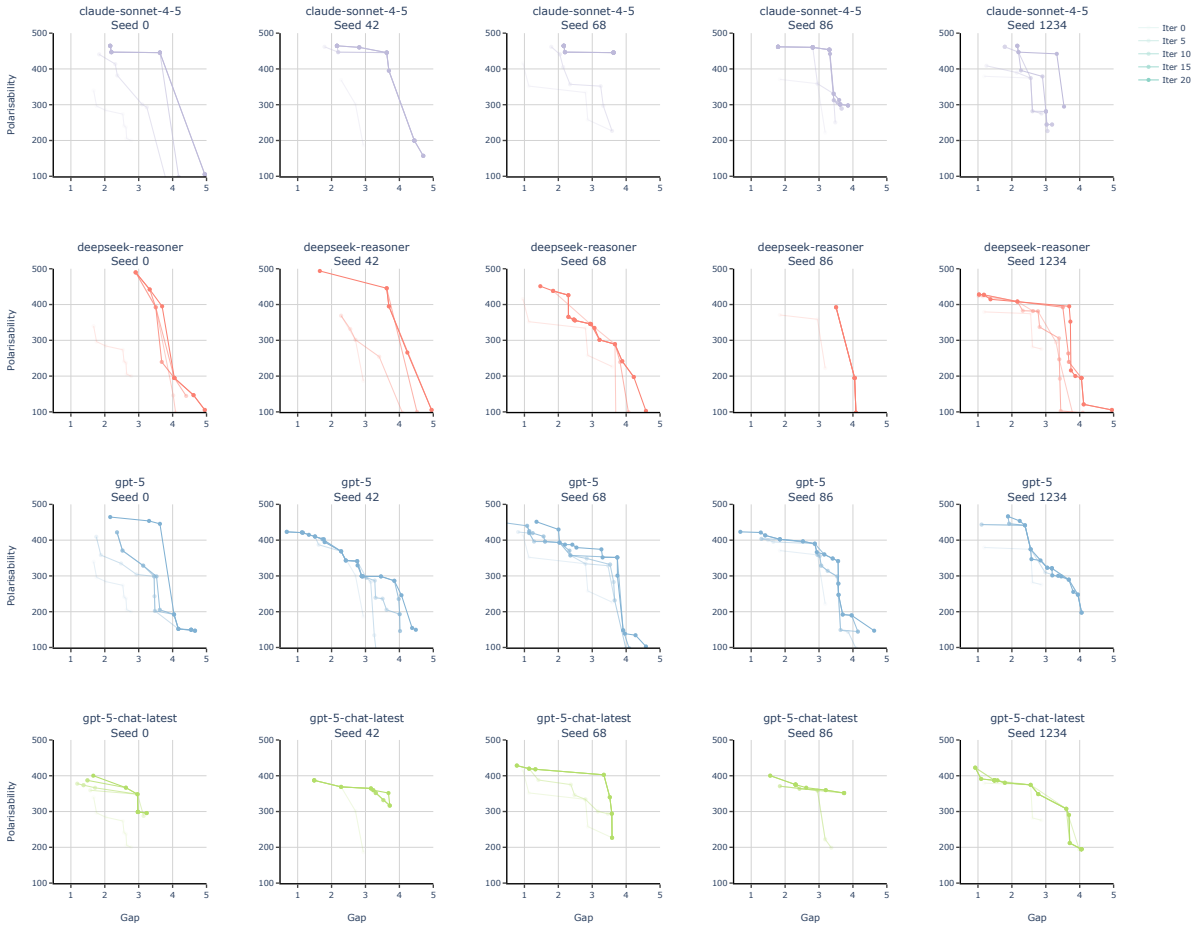

Figure 17. **Pareto frontiers of proposed TMCs by iterations.** Each model is evaluated on five different random seeds and shown in different columns. Results from different models are displayed at different rows. `gpt-5-chat-latest` is colored in green, `gpt-5` in blue, `deepseek-R1` in red, and `claude-sonnet-4.5` in purple, with reduced transparency as iteration number increases.

3.4 Crystal structure discovery

Each experiment began with an initial population of 100 groups of parents ($100 \times 2 = 200$ parent structures), randomly seeded from the reference pool, which is composed with 5,000 known stable structures from MatBench-bandgapt [104] dataset with lowest deformation energy evaluated by CHGNet [105]. The mutation and crossover operations for LLMs were implemented by prompting the LLMs with two sampled parent structures based on their fitness values (minimizing E_{d}) and querying them to propose 5 new structures either through mutation of one structure or crossover of both structures. After generating new offspring in each generation, we evaluated the new offspring and merged their evaluations with the parent evaluations from the previous iteration. The merged pool of parents and children were then ranked by their fitness values (minimizing E_{d}), and the top-100×2 candidates were kept in the population as the pool for the next iteration. We evaluate generated structures through metrics that assess validity, diversity, novelty, and stability. Structural validity checks three-dimensional periodicity, positive lattice volume, and valid atomic positions. Composition validity verifies positive element counts and reasonable number of elements (≤ 10). Structural diversity is computed by deduplicating the generated set using pymatgen's StructureMatcher algorithm, then calculating the ratio of unique structures to total generated. Composition diversity measures the fraction of distinct chemical compositions. For novelty assessment, we compare generated structures against the initial reference pool. Composition novelty identifies structures whose reduced formulas are absent from the reference set. Structural novelty is determined by grouping reference structures by formula, then for each generated structure with a matching formula, using StructureMatcher to check if it matches any reference structure with the same composition; unmatched structures are considered structurally novel. Stability evaluation uses CHGNet to relax structures and compute formation energy, then calculates energy above the convex hull (E_{d}) via a pre-computed patched phase diagram database. We report metastability rates at three thresholds: $E_{\mathrm{d}} < 0.0$ eV/atom (thermodynamically stable), $E_{\mathrm{d}} < 0.03$ eV/atom (highly metastable), and $E_{\mathrm{d}} < 0.10$ eV/atom (M3GNet metastability criterion). The integrated SUN (Structures Unique and Novel) score combines stability and novelty: (1) filter to structures with $E_{\mathrm{d}} < 0.0$ eV/atom; (2) identify unique structures within this stable subset using pymatgen's Structure.matches with scaling enabled; (3) check novelty against the reference pool; (4) compute SUN score as the number of structures simultaneously stable, unique, and novel, divided by the total number of generated structures.

We evaluated multiple LLMs on the crystal structure generation task, comparing their performance against established baseline methods CDVAE [106] and DiffCSP [107]. Table 4 summarizes the results across key metrics when considering both parent and child structures to form the next generation. Overall, LLM-based search achieves superior validity and metastability rates compared to traditional generative models CDVAE and DiffCSP. S.U.N. rate for LLM-based search is evaluated against the entire MatBench-bandgap dataset. GPT-5 achieved the highest overall performance in generating novel structures that are both compositionally novel and thermodynamically metastable. DeepSeek Reasoner and Grok-4 showed competitive metastability rates (88.90% and 87.13% for $E_d < 0.1$ eV/atom, respectively). Comparison among the GPT-5 family suggests that model scale and training objectives significantly impact crystal structure generation quality. Claude Sonnet 4.5 achieves comparable metastability rates to GPT-5-mini but a lower S.U.N. rate (38.99%), indicating less exploration and more exploitation in the search space. In addition, reasoning models like DeepSeek Reasoner require more tokens for internal reasoning chains. With the 8,000 token limit used in these experiments, such models might overflow the context window when generating complex crystal structures with detailed chemical reasoning. Experiments with extended context windows can further improve reasoning model performance on this task.

Table 4. Performance comparison of different models on MatLLMSearch crystal structure generation task. All LLM models were tested with `temperature: 1.0` and `max_tokens: 8000`.

Method	Structural Validity (%)	Comp Validity (%)	Metastability ($E_d < 0.1$ eV/atom, %)	Metastability ($E_d < 0.0$ eV/atom, %)	S.U.N. Rate (%)
CDVAE	100	86.70	28.8	–	–
DiffCSP	100	83.25	–	5.06	3.34
`gpt-5-mini`	100	100	74.60	50.05	46.24
`gpt-5-chat`	100	100	64.36	46.93	44.37
`gpt-5`	100	100	88.33	63.22	55.31
`claude-sonnet-4.5`	100	100	78.71	50.21	38.99
`deepseek-R1`	100	100	88.90	61.22	48.25
`grok-4`	100	100	87.13	60.29	49.80

3.5 Protein sequence optimization

Each experiment began with an initial population of 200 sequences, seeded with one experimentally defined wild-type sequence and 199 additional variants generated by single-site random mutations of the wild type sequence. For GB1 and TrpB, this pool contains all measured 4-site variants, so initialization is not limited to single mutants. The mutation and crossover for LLMs were implemented by prompting the LLMs with two sampled parent sequences based on their fitness values and querying it to propose a new sequence either through mutation of one sequence or crossover of both sequences. If a proposed sequence was not present in the benchmark's provided pool of scored variants (e.g., for GB1 and TrpB), it was treated as invalid and discarded, and the model was prompted again until a valid sequence with a defined fitness value was obtained. When we turned fitness values into probabilities to sample parents, we first normalized the fitness values to ensure non-negativity and then divided by the sum to obtain probabilities. We also applied a small shift to ensure the pool was not dominated by one candidate. After generating 100 new and valid offspring in each generation, the new pool including the offspring and the parents were re-ranked by the fitness values and the top-200 candidates were kept in the population as the pool for next iteration. This process was repeated for 8 iterations. We reported the fitness values of the top 1 candidates, normalized with their $(\min, \max)$ score range specific to each dataset. For GB1[108] and TrpB[109], the range was computed directly from the experimentally measured fitness values in their respective benchmark datasets. For the ML-based oracles (GFP[110] and AAV[111]), the $(\min, \max)$ values were taken from the corresponding oracle's training data to maintain consistency with its predictive scale[112]. For the synthetic Potts-model landscape (Syn-3bfo), the range was fixed to the most frequent region $(\min, \max) = (-3, 3)$. We validated and compared the performance of LLMs against a simple evolutionary algorithm with identical initial populations and hyperparameters. The mutation operator in the baseline was a single-site mutation with a fixed probability of 0.3. Once triggered, a mutation site was chosen at random and flipped to another random amino acid. The crossover operator in the baseline was implemented by selecting two parents based on their fitness values and a crossover site at random, then swapping the prefix and suffix of the two sequences.

Figure 18 summarizes Top-1 performance for LLM-guided search, where each bar reflects the best score achieved per model and then averaged across the five tasks. `deepseek-R1` attains the highest average Top-1 score of 0.8713, improving about 16.0% over the baseline. `gpt-5-chat` and `gpt-5` follow closely at 0.8582 and 0.8561, indicating leading positions in the protein sequence optimization problem.

Beyond the aggregated ranking, the gains are concentrated on the more challenging landscape: on Syn-3bfo, all LLM-guided variants substantially exceed the baseline, with improvements ranging from 59.5% to 103.8%, suggesting that LLM-driven mutation and crossover better navigate fitness landscapes with strong interaction effects between mutation sites, leading to rugged regions where simple local operators struggle. In contrast, results on GB1 and GFP show limited and sometimes mixed gains. For GB1, the effective search space is small and the baseline already performs strongly, leaving little headroom and making the outcome sensitive to whether proposals preserve the few critical sites. For GFP, performance is near saturation for several methods, so differences tend to be smaller and can vary by model. Finally, the mid-tier models show reduced robustness across tasks, most notably `gpt-5-mini`, which performs competitively on some benchmarks yet drops sharply on TrpB, whereas `gpt-5-chat` exhibits the most consistent performance across tasks. Overall, these results suggest that LLM-based search is most valuable for harder search regimes, while robustness across diverse fitness landscapes remains an important consideration.

The convergence curves in Figure 18 show a common pattern across methods: rapid gains in the first few iterations followed by slower, diminishing improvements as the search concentrates around strong candidates. `gpt-5` makes the largest early jump within the first three iterations, suggesting it proposes high-quality mutations or crossovers with fewer oracle evaluations. `deepSeek-R1` improves more gradually but continues to climb in later iterations, which points to stronger stability during refinement rather than relying on a single early leap. `gpt-5-chat` displays a clear improvement around iterations 4 to 5, consistent with moving from initial exploration into a more effective search regime, and it ultimately approaches the top-performing methods. By contrast, `gpt-5-mini` and `claude-sonnet-4.5` increase more slowly and level off earlier at lower scores, indicating less effective candidate proposals under the same evaluation budget. Overall, the curves suggest that differences arise from both final performance and optimization dynamics, including how quickly a model finds good regions and how reliably it improves thereafter[75].

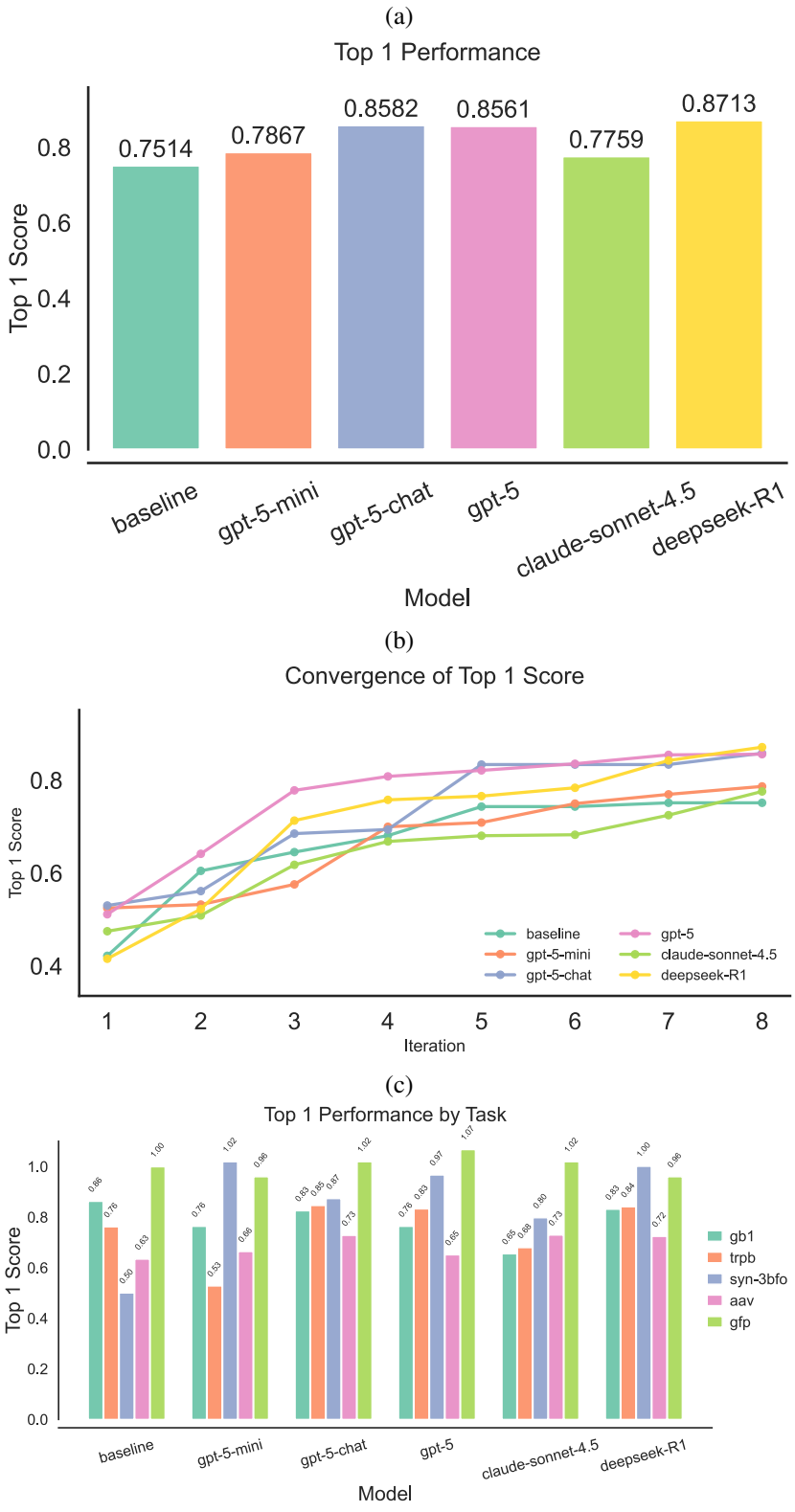

Figure 18. (a) Bar chart of Top-1 performance across models. Bars report the Top-1 score for each model, where higher is better. `deepSeek-R1` achieves the highest Top-1 score at 0.8713, followed by `gpt-5-chat` at 0.8582 and `gpt-5` at 0.8561, while the baseline attains 0.7514. (b) Convergence of Top-1 score over evolutionary iterations for each model. Curves report the mean Top-1 score at each iteration averaged across the five protein optimization tasks, where higher indicates better solutions. (c) Top-1 performance by task and model. For each model cluster on the x-axis, five bars report the final Top-1 score achieved on GB1, TrpB, Syn-3bfo, AAV, and GFP.

3.6 Gene editing

We assess model performance using data from past genetic perturbation experiments. We simulate the perturbation of a gene g by retrieving the relevant observation of the perturbation-induced phenotype f(g) from this dataset. In every experimental round we perturb 128 genes at a single please, representing a reasonably sized small-scale biological screen. Five rounds of experiment are performed with all historical observation fed into the prompts for the next round. We use Interferon-γ (IFNG) dataset, which measures the changes in the production of a key cytokine involved in immune signaling in primary human T-cells.

Across all models, `gpt-5` and `claude-sonnet-4.5` perform similarly well, reaching the same number of total hits with the only difference that `gpt-5` sometimes generates invalid perturbations. Meanwhile, `deepseek-R1` generates fewer hits compared to the two best models. `gpt-5-chat`, despite having much fewer total hits, reaches the best final efficiency as many of genes generated there are invalid and thus would not be tested explicitly in labs.

Table 5. Comparison of 4 LLMs on IFNG gene discovery task across 5 rounds (128 genes per round).

Model	Total hits	Mean hit rate (HR)	Final efficiency	Best round HR	Unique genes tested
`claude-sonnet-4.5`	**83**	**14.65%**	**14.09%**	**30.33%**	**640**
`gpt-5`	**83**	13.84%	13.88%	26.02%	598
`deepseek-R1`	74	12.67%	12.65%	21.67%	585
`gpt-5-chat`	50	8.94%	14.71%	15.83%	340

3.7 Symbolic regression

The symbolic regression task aims to discover closed-form mathematical expressions that govern the underlying dynamics of observed systems directly from data. Given a dataset of input–output pairs $\{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ denotes system states (or derived features) and $y_i \in \mathbb{R}$ denotes target values, the objective is to recover an interpretable symbolic expression $\hat{f}(x)$ that accurately approximates the true governing equation and generalizes beyond the training regime. Unlike purely numerical regression, symbolic regression or equation discovery requires structured exploration over a discrete program space of possible mathematical relations, balancing expressivity, numerical accuracy, and simplicity. This setting naturally lends itself to iterative hypothesis generation and refinement in symbolic spaces, making it well suited for evaluating LLM-guided scientific discovery techniques. At initialization, the LLM-based framework (employed from [77]) constructs a prompt containing a concise description of the target scientific equation discovery task, variable definitions, and a small number of simple in-context examples (e.g., linear or low-degree polynomial relations). These examples seed a multi-island experience buffer that serves as the initial population for evolutionary refinement. At each iteration, the LLM samples candidate equation hypotheses as Python program skeletons of the form

$$\texttt{def f(x\_1, ..., x\_d, params): return y}, \tag{1}$$

where the symbolic structure and logic of the function is proposed by the LLM and the numeric parameters are left unspecified as placeholder parameters. For each proposed skeleton, these continuous placeholder parameters are then optimized against the observed data using off-the-shelf numeric optimizers (e.g., BFGS via `scipy`), yielding a fitted candidate equation. The optimized hypothesis is then evaluated by an oracle that executes the program and computes its fitness as the negative mean squared error (MSE) on the training data. Invalid programs (e.g., numerical instability, execution errors, or degenerate outputs) are discarded. High-scoring equation hypotheses are retained in the external experience buffer, which is organized as a multi-island population to maintain diversity and reduce premature convergence. The evolutionary loop proceeds by repeatedly sampling from this experience buffer to condition subsequent LLM prompts, enabling the model to refine promising structures while continuing to explore novel functional forms. This process is repeated for a fixed number of iterations (up to 1000 in our experiments), after which the highest-scoring equation is selected as the discovered scientific law.

We evaluate symbolic regression on a collection of physics nonlinear dynamical systems from recent benchmark [78]. For each system, trajectories are provided numerically with split into *in-distribution (ID)* and *out-of-distribution (OOD)* regimes, where OOD data corresponds to extrapolation beyond the training range for held-out test purpose of discovered equations. All datasets are normalized following standard practice, using the existing train–test splits across all methods to ensure fair comparison. We compare multiple large language model (LLM) backbones as hypothesis generators within the same symbolic discovery framework, including `claude-sonnet-4.5`, `gpt-5`, `deepseek-R1`, and `gpt-5-chat-latest`. At each iteration, the LLM proposes candidate symbolic programs expressed in a restricted grammar of mathematical operators. These candidates are evaluated numerically against the dataset, ranked by error, and the best-performing programs are retained to guide subsequent proposals.

All LLM-based methods share the same prompt structure, grammar constraints, evaluation pipeline, and iteration budget; only the underlying language model backbone differs. Each experiment is repeated with multiple random seeds, and results are aggregated across datasets and runs. As a non-LLM baseline, we include PySR, a widely used state-of-the-art symbolic regression method based on evolutionary search. PySR is run with recommended default hyperparameters and comparable computational budgets over all datasets. We report both accuracy-based and error-based metrics to capture complementary aspects of symbolic discovery. Specifically, we report accuracy at a relative error threshold $\tau = 0.1$ as a more strict metric of fitness with respect to data:

$$\mathrm{Acc}_\tau = \mathbb{I}\left(\max_{1 \le i \le N_{\text{test}}} \frac{|\hat{y}_i - y_i|}{|y_i|} \le \tau \right), \tag{2}$$

where $\hat{y}_i$ denotes model predictions and $\mathbb{I}(\cdot)$ is the indicator function. We additionally report the normalized mean squared error (NMSE): $\mathrm{NMSE} = \frac{\sum_{i=1}^{N_{\text{test}}} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{N_{\text{test}}} (y_i - \bar{y})^2}$, where $\bar{y}$ is the mean of the ground-truth outputs. Metrics are computed separately for ID and OOD splits of each dataset.

Table **??** summarizes the quantitative performance of all methods on symbolic regression. Among LLM-based approaches, `deepseek-R1` achieves the highest in-distribution accuracy and the lowest NMSE, while `gpt-5` attains comparable performance with stronger out-of-distribution generalization. `claude-sonnet-4.5` performs competitively in-distribution but exhibits slower convergence and higher residual error, particularly in OOD settings. In contrast, `gpt-5-chat-latest` shows substantially degraded performance. Compared to PySR, all LLM methods achieve markedly higher accuracy and mostly lower error. This suggests that LLM-guided discovery can surpass

leading evolutionary symbolic regression by leveraging global structural priors and iterative hypothesis refinement rather than relying solely on pure local search operators. Figure **??** further illustrates these trends through discovery curves, which track the best normalized data-driven error achieved as a function of iteration, averaged across datasets.

Beyond quantitative metrics, we examine the final symbolic programs discovered by each method. Qualitative inspection reveals that `deepseek-R1` and `gpt-5` show robustness in finding interpretable expressions faster in the process of discovery that closely match the true governing equations, often with minimal extraneous terms. `claude-sonnet-4.5` also frequently identifies partially correct structures but mostly with redundant components that are more sensitive to problems and initial populations. Example of representative discovered programs for each method are provided below (example problem PO37), illustrating qualitative differences in symbolic structure, compactness, and physical interpretability.

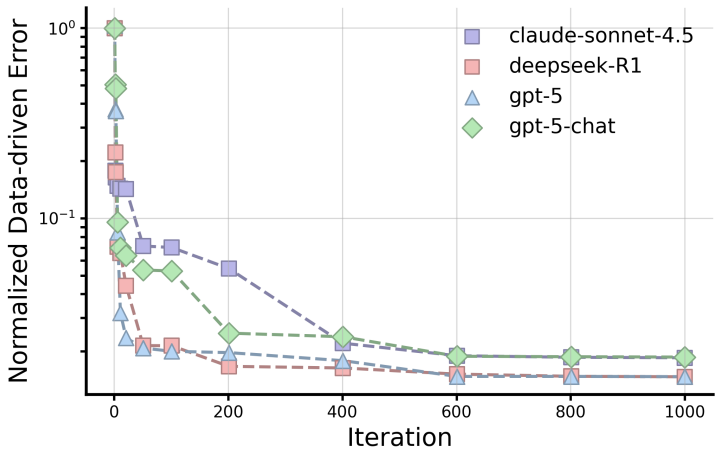

Figure 19. Discovery curves for symbolic regression across LLM backbones. Normalized data-driven error (lower is better) of the best-discovered equation as a function of LLM-guided iterations, averaged across all the benchmark datasets.

Table 6. Quantitative comparison results on symbolic regression. In-distribution (ID) and out-of-distribution (OOD) performance for symbolic regression, reporting accuracy to threshold 0.1 ($Acc_{0.1}$, higher is better) and normalized mean squared error (NMSE, lower is better). Results compare LLM-based methods and PySR, a non-LLM state-of-the-art baseline, highlighting differences in accuracy, error, and generalization to OOD data. Best values are highlighted in **bold**.

Model	$Acc_{0.1}$ **ID (%)**↑	$Acc_{0.1}$ **OOD (%)**↑	**NMSE ID**↓	**NMSE OOD**↓
PySR	13.79	3.44	1.0786	475.7
`claude-sonnet-4.5`	55.17	48.28	0.03896	274.5
`gpt-5`	55.17	**51.72**	0.00611	**266.3**
`deepseek-R1`	**58.62**	**51.72**	**0.00426**	1009.1
`gpt-5-chat`	48.28	34.48	0.03823	91118.4

Qualitative example for final discovered equation programs. We present a qualitative example for benchmark problem PO37 [78]. Figure 20 shows the phase-space trajectory of the ground truth non-linear oscillator generated dynamics, alongside its ground-truth equation skeleton.

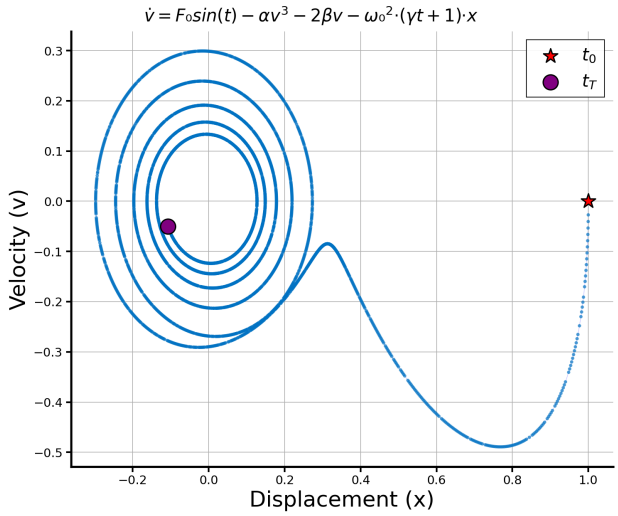

Figure 20. Phase-space trajectory (x vs. v) visualization of example non-linear oscillator problem PO37 [78] The plot annotates the ground-truth equation skeleton and indicates the initial state t_0 and final state t_T.

```
########## PySR #########
params[0]**exp(t)*x*t - params[1] - x/params[2] - x*params[3]
```

```
########## GPT-5 #########
def equation(x, t, v, params):
    # Unpack parameters
    k1, k2, k3, c1, c2, c3, F1, w1, phi1, bias = params
    # Stiffness terms (linear + time-varying)
    stiffness = -(k1 + k2 * t) * x
    # Nonlinear stiffness (Duffing)
    cubic_stiffness = -k3 * x**3
    # Damping (linear + position-dependent + cubic velocity)
    damping = -(c1 + c2 * x**2) * v - c3 * v**3
    # External periodic driving
    drive = F1 * np.sin(w1 * t + phi1)
    # Combine all terms
    acceleration = stiffness + cubic_stiffness + damping + drive + bias

    return acceleration
```

```
########## DeepSeek-R1 #########
def equation(x, t, v, params):
    # Unpack parameters
    w_sq, beta, delta1, delta2, alpha, F, omega, gamma, epsilon, lambda_ = params
    # Restoring forces
    linear_restoring = -w_sq * x
    cubic_restoring = -beta * x**3
    quintic_restoring = -epsilon * x * np.abs(x)**4
    # Damping forces
    linear_damping = -delta1 * v
    quadratic_damping = -delta2 * v * np.abs(v)
    position_dependent_damping = -alpha * x**2 * v
    # Position-velocity coupling
    position_velocity_coupling = -gamma * x * np.abs(v)
    # External forcing with modulation
    sinusoidal_forcing = F * np.sin(omega * t)
    envelope = np.exp(-lambda_ * np.abs(t))
    modulated_forcing = sinusoidal_forcing * envelope
```

```
    # Mixed cubic term
    cubic_mixed = -0.1 * (v**3) * np.sign(x) * np.exp(-0.01 * x**2)
    # Combine all terms
    acceleration = (
        linear_restoring +
        cubic_restoring +
        quintic_restoring +
        linear_damping +
        quadratic_damping +
        position_dependent_damping +
        position_velocity_coupling +
        modulated_forcing +
        cubic_mixed
    )

    return acceleration
```

```
########## Claude-Sonnet-4.5 #########
def equation(x, t, v, params):
    # Restoring forces (odd powers preserve symmetry)
    linear_restoring = params[0] * x
    cubic_restoring = params[1] * x**3
    quintic_restoring = params[2] * x**5
    # Linear damping
    linear_damping = params[3] * v
    # Non-linear damping mechanisms
    van_der_pol_damping = params[4] * x**2 * v
    cubic_velocity_damping = params[5] * v**3
    # Cross-coupling term
    velocity_position_coupling = params[6] * x * v**2
    # Asymmetric potential (breaks x -> -x symmetry)
    asymmetric_force = params[7] * x**2
    # Parametric forcing
    time_dependent_forcing = params[8] * np.cos(params[9] * t) * x
    # Combine all terms
    acceleration = (
        linear_restoring +
        cubic_restoring +
        quintic_restoring +
        linear_damping +
        van_der_pol_damping +
        cubic_velocity_damping +
        velocity_position_coupling +
        asymmetric_force +
        time_dependent_forcing
    )

    return acceleration
```

```
########## GPT-5-Chat #########
def equation(x, t, v, params):
    # Extract parameters
    inv_mass = params[0]  # 1/m (inverse mass)
    k = params[1]         # linear stiffness
    c = params[2]         # linear damping
    beta = params[3]      # cubic stiffness
    delta = params[4]     # cubic damping
    F1 = params[5]        # forcing amplitude 1
    omega1 = params[6]    # forcing frequency 1
    phi1 = params[7]      # forcing phase 1
    F2 = params[8]        # forcing amplitude 2
    omega2 = params[9]    # forcing frequency 2
    # Calculate force components
```

```
    restoring = -k * x - beta * (x ** 3)
    damping = -c * v - delta * (v ** 3)
    forcing = F1 * np.cos(omega1 * t + phi1) + F2 * np.sin(omega2 * t)
    # Calculate acceleration
    a = inv_mass * (restoring + damping + forcing)

    return a
```

3.8 Solving Ising model

In this project, each experiment targets the discovery of a low-energy configuration of a classical Ising model with unknown optimal bitstring structure, given a fixed but hidden set of pairwise couplings. The system is defined by a random all-to-all connected N-spin Ising Hamiltonian

$$H(\mathbf{s}) = \sum_{i<j} J_{ij} s_i s_j, \tag{3}$$

where $s_i \in \{+1, -1\}$ are spin variables and J_{ij} are symmetric couplings drawn randomly at the beginning of each experiment and held fixed throughout the run. The objective is to recover a bitstring $\mathbf{b} \in \{0, 1\}^N$, mapped to spins via $s_i = +1$ if $b_i = 1$ and $s_i = -1$ otherwise, that minimizes the Hamiltonian energy.

Unlike traditional combinatorial optimization methods that rely on local update rules or handcrafted heuristics, we formulate the problem as a LLM-guided evolutionary search. Candidate solutions are generated by a large language model (LLM), evaluated by a physics-based oracle, and iteratively refined through an experience-driven prompt-conditioning mechanism.

At initialization, the framework constructs a prompt describing the Ising optimization task, including the Hamiltonian definition, the spin–bit mapping, and the system size N. A small set of simple candidate bitstrings (e.g., all zeros, all ones, and alternating patterns) is provided as in-context examples to seed the search. These initial candidates define the starting population for the evolutionary process. At each iteration, the LLM is queried to propose exactly one new candidate bitstring of length N. The output is constrained to a raw binary string without explanation, ensuring that each proposal corresponds to a discrete global configuration of the Ising system. In this formulation, each LLM output represents a symbolic hypothesis over the full configuration space rather than a local modification rule.

Each candidate bitstring is evaluated by a deterministic oracle that computes its energy under the fixed Hamiltonian in Eq. 3. For benchmarking and convergence monitoring, the exact ground-state energy is precomputed via brute-force enumeration for the system sizes considered. The primary fitness signal is defined as the normalized energy gap above the ground state. Rather than employing explicit genetic operators such as mutation or crossover, evolutionary refinement is implemented implicitly through prompt conditioning and experience accumulation. The framework maintains an external experience buffer that stores previously generated bitstrings together with their associated fitness scores. At each iteration, information derived from this buffer, including previously explored configurations, the current best-so-far energy, and qualitative feedback on search progress, is incorporated into the LLM prompt. When the energy gap remains large, the prompt encourages exploration of qualitatively different bit patterns; as the gap narrows, the prompt biases the LLM toward local refinements around promising configurations. This mechanism induces selection pressure while maintaining diversity in the search process.

The experiment is repeated for 40 iterations per round and 4 rounds in total are conducted. Throughout the optimization, the best-so-far bitstring and its corresponding energy are tracked at each iteration, yielding a complete optimization trajectory. The proposed framework is evaluated on Ising models with system size $N = 12$ and a randomly generated coupling matrix. Performance is assessed using the final energy gap relative to the exact ground state and the convergence behavior across iterations. As a classical baseline, simulated annealing is implemented using single-spin-flip Metropolis updates, where at each step one randomly selected spin is flipped and the move is accepted according to the Boltzmann criterion. The temperature follows an exponential cooling schedule $T_k = T_0 \alpha^k$ with $T_0 = 1.0$ and $\alpha = 0.9$, and the total number of annealing steps k is matched to the number of LLM iterations to ensure a controlled comparison. By comparing the LLM-generated trajectories to the exact solution and the baseline simulated annealing, we quantify both optimization effectiveness and the ability of the method to navigate rugged, high-dimensional energy landscapes.

Overall, this framework demonstrates how LLMs can be repurposed as global, structure-aware proposal mechanisms for discrete physics optimization problems. By combining LLM-based hypothesis generation with exact physical evaluation and an implicit evolutionary memory, the method enables systematic exploration of exponentially large configuration spaces. We observe substantial variability in performance across different LLMs, and at the present stage no consistent advantage is found over conventional baselines such as simulated annealing. Future work aimed at understanding and improving the synergy between language-guided reasoning and physics-based evaluation may establish a general and effective paradigm for language-assisted scientific optimization.

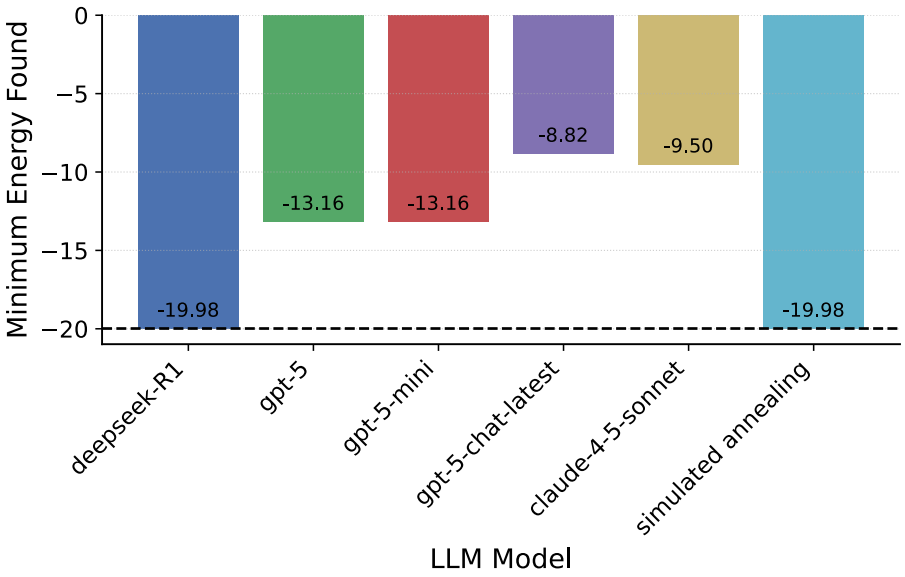

Figure 21. Bar chart of Ising model energy minimization across models. Bars report the score for each model, where lower is better. DeepSeek-R1 achieves the best score at -19.98, followed by GPT-5 at -13.16 and GPT-5-mini at -13.16, while the baseline (simulated annealing) attains -19.98. The ground truth energy is -19.98.

Table 7. Correspondence between projects and their top-3 scenarios in this work.

Project	Scenario
Protein sequence optimization	protein localization, CRISPR delivery, property matching
Gene editing	gwas causal gene, gene editing, molecular pair
Retrosynthesis pathway design	retrosynthesis, reaction mechanism, forward reaction prediction
Molecule optimization	descriptor prediction, fragment completion, molecular property
TMC optimization	TMC properties, redox potential, molecular property
Crystal structure discovery	general materials, PXRD lattice prediction, composite materials
Symbolic regression	computation, core knowledge, statistics
Solving Ising model	computation, condensed matter, quantum information

Supplementary References

80. Schwaller, P. *et al.* Mapping the space of chemical reactions using attention-based neural networks. *Nature machine intelligence* **3,** 144–152 (2021).
81. Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S. & Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *Journal of Chemical Information and Modeling* **52.** PMID: 22587354, 1757–1768. DOI: 10.1021/ci3001277. eprint: https://doi.org/10.1021/ci3001277 (2012).
82. Kydlíček, H. *Math-Verify: Math Verification Library* version 0.6.1.
83. Park, J. B. & Bronzino, J. D. *Biomaterials: principles and applications* (crc press, 2002).
84. Woźniak, A. *et al.* The influence of plasma-sprayed coatings on surface properties and corrosion resistance of 316L stainless steel for possible implant application. *Archives of Civil and Mechanical Engineering* **21,** 148 (2021).
85. Corliss, B. A., Azimi, M. S., Munson, J. M., Peirce, S. M. & Murfee, W. L. Macrophages: an inflammatory link between angiogenesis and lymphangiogenesis. *Microcirculation* **23,** 95–121 (2016).
86. Liu, G. *et al. NeurIPS – Open Polymer Prediction 2025* https://kaggle.com/competitions/neurips-open-polymer-prediction-2025. Kaggle competition dataset. 2025.
87. Jain, A. *et al.* Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL materials* **1** (2013).
88. Chung, Y. G. *et al.* Advances, updates, and analytics for the computation-ready, experimental metal–organic framework database: CoRE MOF 2019. *Journal of Chemical & Engineering Data* **64,** 5985–5998 (2019).
89. Ong, S. P. *et al.* Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science* **68,** 314–319 (2013).
90. Togo, A., Shinohara, K. & Tanaka, I. Spglib: a software library for crystal symmetry search. *Science and Technology of Advanced Materials: Methods* **4,** 2384822 (2024).
91. De Graef, M. & McHenry, M. E. *Structure of materials: an introduction to crystallography, diffraction and symmetry* (Cambridge University Press, 2012).
92. Khan, S. T. & Moosavi, S. M. Connecting metal-organic framework synthesis to applications using multimodal machine learning. *Nature Communications* **16,** 5642 (2025).
93. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* **17,** 261–272 (2020).
94. Lowe, D. M. *Extraction of chemical structures and reactions from the literature* PhD thesis (Apollo - University of Cambridge Repository, 2012). DOI: 10.17863/CAM.16293.
95. Yu, K. *et al. Double-ended synthesis planning with goal-constrained bidirectional search* in *Advances in Neural Information Processing Systems (NeurIPS)* **37** (2024), 112919–112949.
96. Coley, C. W., Rogers, L., Green, W. H. & Jensen, K. F. SCScore: synthetic complexity learned from a reaction corpus. *Journal of chemical information and modeling* **58,** 252–261 (2018).
97. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics* **1,** 8 (2009).
98. Software, N. *Pistachio (January 2024)*
99. Segler, M. H., Preuss, M. & Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **555,** 604–610 (2018).

100. Zhong, W., Yang, Z. & Chen, C. Y.-C. Retrosynthesis prediction using an end-to-end graph generative architecture for molecular graph editing. *Nature Communications* **14,** 3009 (2023).
101. Zhong, Z. *et al.* Root-aligned SMILES: a tight representation for chemical reaction prediction. *Chemical Science* **13,** 9023–9034 (2022).
102. Chen, S. & Jung, Y. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au* **1,** 1612–1620 (2021).
103. Ioannidis, E. I., Gani, T. Z. H. & Kulik, H. J. molSimplify: A toolkit for automating discovery in inorganic chemistry. *Journal of Computational Chemistry* **37,** 2106–2117 (2016).
104. Dunn, A., Wang, Q., Ganose, A., *et al.* Benchmarking Materials Property Prediction Methods: The Matbench Test Set and Automatminer Reference Algorithm. *npj Comput. Mater.* (2020).
105. Deng, B., Zhong, P., Jun, K., *et al.* CHGNet as a Pretrained Universal Neural Network Potential for Charge-Informed Atomistic Modelling. *Nat. Mach. Intell.* (2023).
106. Xie, T., Fu, X., Ganea, O.-E., *et al. Crystal Diffusion Variational Autoencoder for Periodic Material Generation* in *ICLR* (2022).
107. Jiao, R., Huang, W., Lin, P., *et al.* Crystal Structure Prediction by Joint Equivariant Diffusion. *NeurIPS* (2024).
108. Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O. & Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *Elife* **5,** e16965 (2016).
109. Johnston, K. E. *et al.* A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proceedings of the National Academy of Sciences* **121,** e2400439121 (2024).
110. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533,** 397–401 (2016).
111. Bryant, D. H. *et al.* Deep diversification of an AAV capsid protein by machine learning. *Nature Biotechnology* **39,** 691–696 (2021).
112. Kirjner, A. *et al. Improving protein optimization with smoothed fitness landscapes* in *The Twelfth International Conference on Learning Representations* (2023).