Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

MutexaGPT: An Intuition-to-Design Translator for Physics-based Enzyme Engineering

Qianzhen Shao
Vanderbilt University https://orcid.org/0000-0002-7787-0966
Yinjie Zhong
Vanderbilt University
Sebastian Stull
Vanderbilt University
Xinchun Ran
Vanderbilt University
Ning Ding
Vanderbilt University
Kieran Nehil-Puleo
Vanderbilt University
Ruizhe Yao
Vanderbilt University
Han Xu
Vanderbilt University
Zhongyue Yang
zhongyue.yang@vanderbilt.edu

Vanderbilt University

Research Article

Keywords: enzyme engineering, EnzyHTP, large-language model, LLM agent

Posted Date: June 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10072540/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

MutexaGPT: An Intuition-to-Design Translator for Physics-based Enzyme Engineering

Qianzhen Shao[1], Yinjie Zhong[1], Sebastian Stull[1,6], Xinchun Ran[1], Ning Ding[1], Kieran Nehil-Puleo[7], Ruizhe Yao[6], Han Xu[1], and Zhongyue J. Yang[1-5,*]

[1]*Department of Chemistry, Vanderbilt University, Nashville, Tennessee 37235, United States*

[2]*Center for Structural Biology, Vanderbilt University, Nashville, Tennessee 37235, United States*

[3]*Vanderbilt Institute of Chemical Biology, Vanderbilt University, Nashville, Tennessee 37235, United States*

[4]*Data Science Institute, Vanderbilt University, Nashville, Tennessee 37235, United States*

[5]*Department of Chemical and Biomolecular Engineering, Vanderbilt University, Nashville, Tennessee 37235, United States*

[6]*Department of Computer Science, Vanderbilt University, Nashville, Tennessee 37235, United States*

[7]*Interdisciplinary Materials Science Program, Vanderbilt University, Nashville, Tennessee 37235, United States*

**Corresponding author's email: zhongyue.yang@vanderbilt.edu*

**Lead contact's email: zhongyue.yang@vanderbilt.edu*

Abstract

Physical intuitions about how enzyme structure and dynamics influence its function and property have enabled successful engineering outcomes, yet a systematic approach remains unknown for translating these qualitative and abstract "thoughts" into quantitative, actionable principles that lead to designs. Here we introduce MutexaGPT, an open-access, multi-agent large language model (LLM) platform that translates enzyme engineering intuition to physics-based simulations and thus variant designs. Through a web interface, MutexaGPT takes users' intuition-driven requests (expressed in plain English) as input, and then leverages its LLM agents (i.e., QuestionAnalyzer, WorkPlanningBoard, and ResultExplainer) to comprehend the request and elicit missing information, construct physics-based models, configure and execute high-throughput molecular modeling workflows, and eventually convert the molecular modeling results into actionable design proposals (such as a smart mutation library). An automated evaluation

framework was established to systematically benchmark the prompt engineering strategies that allow each individual agent to achieve an optimal performance. We further demonstrate the utility of MutexaGPT in two protein engineering tasks. For the task of engineering halide methyltransferase towards bulkier substrates, MutexaGPT converted a cavity-engineering intuition into a smart library design that shows a 40% hit rate and around 4-fold activity improvement over a baseline strategy. For the task of engineering bidomain amylase for enhanced activity at lower temperature, MutexaGPT translated a statistics-based intuition into cold-adapted amylase variants that show 1.7-fold and 3.7-fold activity enhancement at 0 °C (experimentally validated). These results establish MutexaGPT as an intuition-to-design translator that integrates human creativity with high-throughput molecular modeling to democratize physics-guided, intuition-driven enzyme engineering.

1. Introduction

Physical intuition[1, 2] has long been a cornerstone of enzyme engineering. It arises from accumulated knowledge and experience in rationalizing how enzymes realize their catalytic functions through three-dimensional structural topology (e.g., active-site cavities and tunnels[3-11]), conformational dynamics (e.g., flexibility and domain motion[12-19]), electronic structure (e.g., electrostatics and electric fields[20-25]), and interactions with the surrounding environment (e.g., hydration shell). Such intuition helps protein engineers infer or anticipate the impact of mutations on enzyme activity[26-29], temperature adaptation[30], selectivity[31-35], promiscuity[36-41], and stability of enzymes[42, 43]. For instance, steric intuition in amine transaminases has led to a design rule for tuning stereoselectivity: by adjusting the volumes of the large and small binding pockets within the active site[3-8] through site-directed mutagenesis to better match the sizes of the corresponding ketone substituents, one can invert selectivity or expand substrate scope.[33, 44, 45] Such intuition also strengthens the screening-based, directed evolution approaches for effective function-enhancing mutant discovery. Reetz and coworkers exemplified this through developing the combinatorial

active-site saturation test (CASTing) method, which applies targeted saturation mutagenesis to small residue sets and recombines beneficial variants to capture cooperative effects for enhancing activity and selectivity toward non-native substrates.[32, 34, 35, 37, 46-48] In essence, physical intuition guides the construction of "smart" mutational libraries and facilitates effective enzyme engineering when high-throughput screening assays are not available.

Despite its importance, translating physical intuition into successful enzyme engineering outcomes remains difficult. In practice, most intuitive catalytic principles in enzymology are qualitative rather than quantitative. Gordon Hammes' famous notion that "For catalysis (of an efficient enzyme), flexible but not too flexible, as well as rigid but not too rigid, is essential" captures a conceptual truth yet provides no guidance for identifying the optimal degree of flexibility.[49] Even when an intuition has a quantitative foundation, such as the correlation between electric field strength and catalytic rate enhancement,[20-25] it remains unknown how specific mutations should be introduced to achieve the desired electrostatic environment. As a result, the unique intuition or a stroke-of-genius inspiration from protein engineers always fails to get translated into a "smart" mutational library or a concrete protein engineering strategy. This makes it inevitable to adopt a hypothesis-agonistic approach by screening against a larger library for mutant discovery, leading to greater labor/resource cost and longer enzyme development cycle in most enzyme engineering scenarios.

The pressing challenge facing the entire field of enzyme engineering is how to establish an enzyme engineering paradigm that translates physical intuitions into quantitative, actionable design principles. One promising approach to address this challenge requires high-throughput molecular simulations (htMS), which quantify how mutations alter enzyme structure, dynamics, and energetics to predict their effects on catalytic activity and selectivity. For example, SubTuner

leveraged htMS (specifically quantum mechanics (QM) and molecular dynamics (MD) simulations) to translate physical hypotheses (such as optimal transition state binding and electric field stabilization) into quantitative metrics in the *in silico* screening workflow and eventually yield smart libraries (~10 mutants for each task) that constantly achieved ~30% hit rate across three transferase engineering tasks. Similarly, CASCO[50] leveraged high-throughput MD simulations to quantify substrate positioning dynamics as near-attach-conformation frequencies in the engineering of the enantioselectivity of a limonene epoxide hydrolase; QzymeDesigner[44] used semi-empirical QM to quantify the substrate binding as free energies in the engineering of the substrate scope of the ω-transaminase from *Chromobacterium violaceum*; and AI.zymes[51] integrated MD screening into a multi-objective design workflow, using metrics such as electric field strength, ligand flexibility, etc. to prioritize variants for improving the promiscuous Kemp eliminase activity in a ketosteroid isomerase. Yet, despite its demonstrated success, htMS by itself cannot fully bridge the gap between intuition and enzyme engineering practice; its complexity and technical demands often limit accessibility beyond specialized computational experts. As a result, numerous potentially transformative approaches for intuition-guided, molecular modeling-based enzyme engineering have stalled as one-off proof-of-concepts rather than scalable solutions.

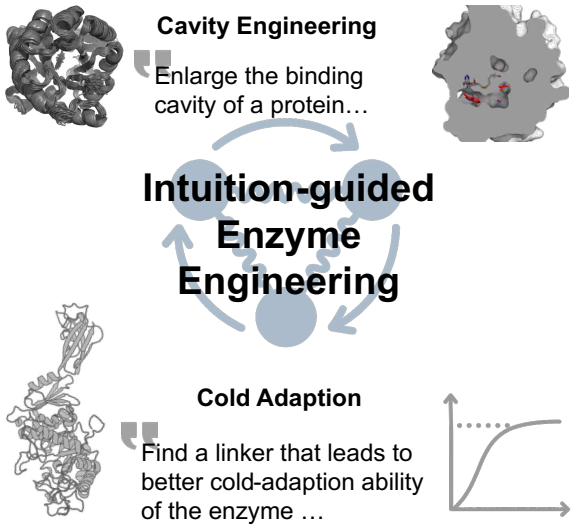


Figure 1. Physical Intuition-guided Enzyme Engineering.

To fully unlock the potential of intuition-guided enzyme engineering, we propose a computational platform that unites large language models (LLMs) with high-throughput molecular simulations. Our solution satisfies two critical needs simultaneously: first, translating qualitative and descriptive intuition into explicit, executable instructions; and second, converting those instructions into automated computational workflows that deliver quantitative, physics-based outputs. To this end, we developed MutexaGPT, an intuition-to-design translator of molecular simulation for enzyme engineering. MutexaGPT is built based on large language model (LLM) agents and works as an open-access web application (https://enzyhtp.app.vanderbilt.edu) that interacts with users in plain English, converts intuition-based requests from users into computational modeling plans, performs high-throughput MD simulations, analyzes the results, and summarizes them into actionable insights for enzyme engineering (such as a "smart" library pinpointing a few mutants). Note that although MutexaGPT ranks and filters enzyme variants based on computed metrics, it is not a predictive AI tool, but instead an integrated system for automation and intuition-translation. Designed for protein engineers and designers with little or no

molecular modeling expertise, MutexaGPT is distinct from existing computational chemists-oriented LLM-agentic molecular simulation tools (e.g., MDCrow[52], NAMD-agents[53], DynaMate[54], and ProtAgents[55]) that require users' domain knowledge to choose theoretical models, design simulation protocols, and interpret results. We achieve so by equipping MutexaGPT's LLM agents with necessary domain knowledge to guide users through the entire life cycle of high-throughput physics-guided enzyme engineering,[56, 57] from creating the scientific question to interpreting the results. We demonstrated the effectiveness of MutexaGPT through two cases that showcase how empirical intuition of cavity engineering and statistics-based intuition of cold-adaptation engineering can be converted to physics-based models and guide function-gaining mutant discovery (Figure 1B). Complementary to screening-dominated approaches, MutexaGPT provides a scalable route to operationalize intuition-driven, physics-guided enzyme engineering within the scope of currently implemented metrics and simulations.

2. Implementation

2.1 The workflow of the multi-agent system in MutexaGPT

At the heart of MutexaGPT is a multi-LLM-agent system that consists of QuestionAnalyzer, WorkPlanningBoard, and ResultExplainer (Figure 2A). They run, in the web-application backend, to co-create intuition-driven requests, supply necessary domain knowledge, convert the request into physical models, configure computational protocols, and execute the simulations. The users may begin by providing an intuition-driven request (e.g., "help me enlarge the active site") into the system and expect MutexaGPT to execute relevant computational modeling to guide their enzyme engineering tasks. In its well-versed version, these initial, intuition-driven requests should convey three elements: 1) enzymes of interest, 2) an optional, initial mutation library of interest,

and 3) engineering objectives such as enzyme efficiency, stability, etc. Real requests from users, however, are often vague (e.g., the user may just say "mutations" instead of having a specific mutation library in mind), physically inconsistent (e.g., "improve the temperature of the backbone" make no sense), or completely off-topic (e.g., user may misunderstand the specialization of the webapp and ask it to recommend a protein shake brand).

The QuestionAnalyzer agent steps in and starts a conversation with the user to address these problems: for those irrelevant inquiries, it explains the specialization of the webapp and asks the user to resubmit a new inquiry. For the relevant ones, it actively checks for the missing details, helps the user to articulate implicit assumptions, provides suggestions to reframe scientific terms for clarity, and eventually co-creates a scientifically sound and concrete request (see Text S1 for an example dialog). The revised question that satisfies QuestionAnalyzer will then be passed to the WorkPlanningBoard, which is composed of two agents, MetricsPlanner and MutantPlanner. The MetricsPlanner maps the engineering objectives of users to a list of physics-based, EnzyHTP[56, 57]-computable metrics. For example, a complex target such as "drug resistance of mutants" will be decomposed into a combination of enzyme activity, binding affinity, and thermostability, which are then mapped to internal electric field strength, binding free energy, and folding energy. MetricsPlanner functions through a conversation with the user, where it will first suggest combinations of metrics, as well as educate the user with necessary domain knowledge explaining the combination. If the user accepts a suggestion, the MetricsPlanner checks whether all mandatory arguments are present, including: selection masks, restraint definitions, protonation states, and so on. Any omissions are surfaced as explicit queries, each accompanied by a plausible default. This dialogue repeats until every parameter is well defined, at which point the agent compiles a JSON file as the final output in the back end (see Text S1 for an example dialog). In parallel to

MetricsPlanner, MutantPlanner executes silently in the background and maps the user's mutation plan into the EnzyHTP mutation pattern, a powerful syntax defined in EnzyHTP that describes a list of mutants each composed of a list of mutations. Combining the output of MetricsPlanner and MutantPlanner gives the final payload JSON object that is ready to be executed by EnzyHTP autonomously. Notably, MutexaGPT allows users to leverage resources from Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University for the requested computations. The platform also provides a choice for users to download a folder of prepared input files of all needed simulations and a simple script for configuring the environment, executing them using the users' local computing resources, and uploading them back to MutexaGPT if needed. Note that to further support reproducibility if users want to publish any data generated from MutexaGPT, in the workflow review page, the website allows users to export the exact EnzyHTP Python script that is executed. This script contains full simulation setup and parameters and can be directly rerun or cited in Methods/Supplementary sections to ensure reproducibility, especially for users without extensive computational background.

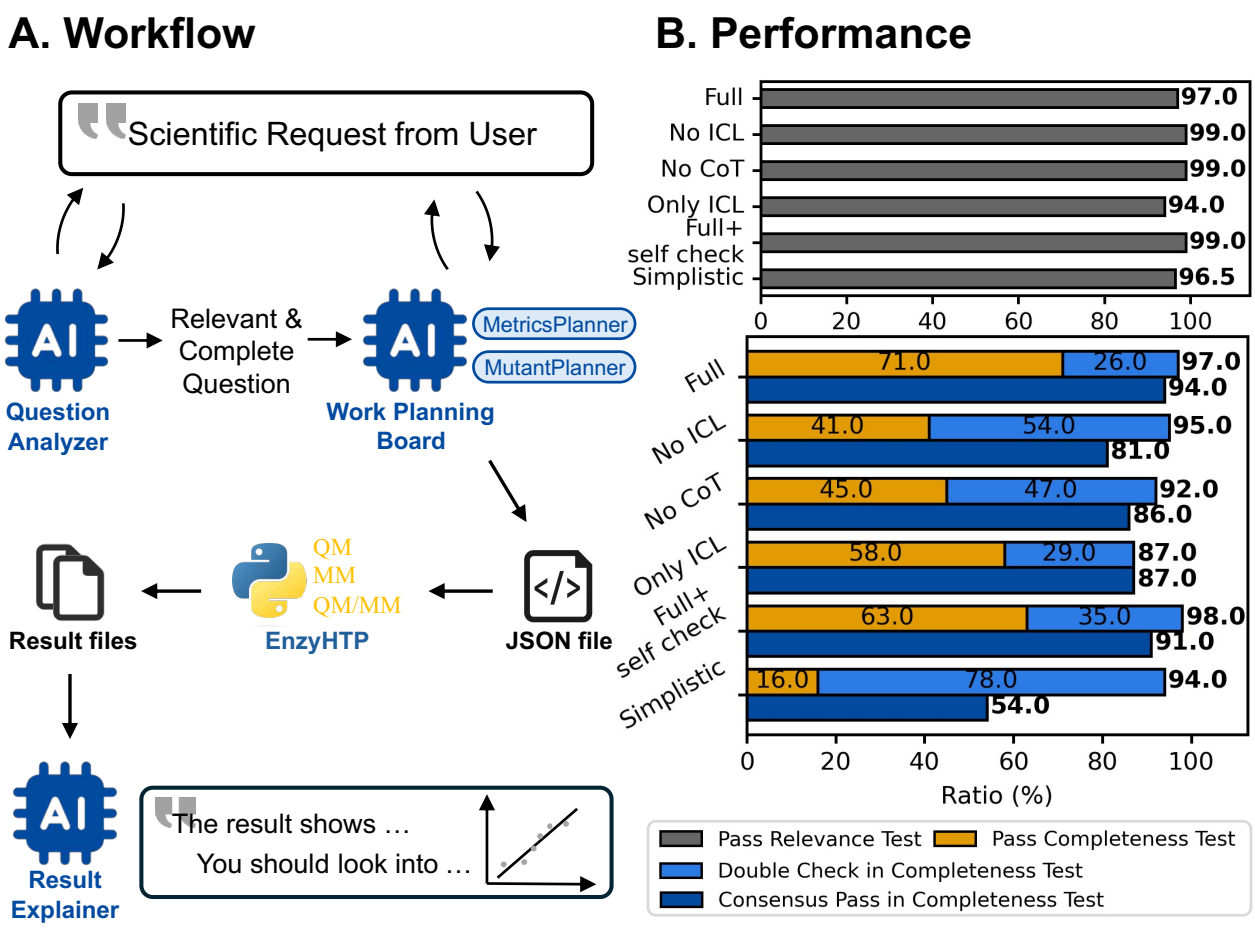


Figure 2. The Design of MutexaGPT. (A) The architecture and workflow of the multi-agent system in MutexaGPT. With a user posting their scientific request, QuestionAnalyzer checks the request for the missing details, provides suggestions to better define the scientific question, helps the user articulate their assumptions, and eventually co-creates a scientifically sound and concrete request. The request is then sent to WorkPlanningBoard (consisting of MetricsPlanner and MutantPlanner) to come up with a molecular modeling plan and seek approval from the user. The user-confirmed plan is then saved as a JSON file, which is sent to the EnzyHTP in the backend for performing the high-throughput simulations. ResultExplainer dissects the result files from the simulation and summarizes them into a concise report for the user. (B) Benchmark of system prompt engineering strategies in the QuestionAnalyzer for relevance (up) and completeness

(bottom). (Full = full prompt; No ICL = full prompt with in context learning part removed; No CoT = full prompt with the chain of thought part removed; Only ICL = a prompt containing only in context learning examples; Full + self-check = augmenting the Full prompt with a one-line self-reflection instruction; Simplistic = only contains a sentence of role description and the output format in the system prompt.) The full evaluation results for all agents are summarized in SI.zip/agent_evaluation.

Once the simulations are complete, their outputs are collected into a single JSON object that contains (i) the user's original scientific question, (ii) the mean ± standard-deviation values for every requested metric, (iii) URLs for any downloadable files (trajectory pickles, CSV tables for full results, etc.), and (iv) metadata of the simulation (force field, length of the MD, equilibration assessment, etc.). This JSON becomes the sole input to the terminal agent, ResultExplainer. Acting under the prompt shown in Supplementary Table S2, ResultExplainer handles five top-level goals: (1) translate raw modeling results into concise, domain-relevant takeaways; (2) anchor every explanation to the user's scientific question rather than to the data in isolation; (3) explain, for each reported metric, what it may indicate biologically or structurally and, equally importantly, what it does not directly measure or prove, thereby reducing the risk of over-interpretation by non-computational users; (4) communicate the confidence and limitations of the analysis in light of the available simulation context, such as simulation length and equilibration information; and (5) prioritize the most relevant variants and recommend follow-up experiments, additional simulations, or data checks (e.g., ResultExplainer can rank/filter enzyme variants based on computed metrics; see Text S10 for examples of ResultExplainer output). By this final message, MutexaGPT closes the loop from a vague, qualitative intuition to experiment-

ready design recommendations with molecular level insights (see Text S7 for the design of each individual LLM-agent involved in this workflow).

2.2 The performance of agents in MutexaGPT

To assess the performance of the three core agents in MutexaGPT (i.e., **QuestionAnalyzer**, **WorkPlanningBoard**, and **ResultExplainer**), we designed an automated evaluation framework for MutexaGPT (Text S8 and Figure S2). This framework leverages large language models for both generating diverse natural-language test cases (i.e., test-set curator agents) and evaluating agent outputs (i.e., answer classifier agents). Specifically, a group of test-set curator agents was deployed to create test cases tailored to each core ("testee") agent in MutexaGPT, and all generated cases were manually reviewed for accuracy. For evaluation, a set of classifier agents was used to categorize each testee agent's natural-language output into predefined labels, which were then compared with manually verified ground-truth labels to determine pass/fail outcomes (see Text S8 for details).

Using this evaluation framework, we benchmarked the performance of the core agents in MutexaGPT. We focus on **QuestionAnalyzer** here as a representative example (Figure 2B) because it is the most sophisticated agent and most directly tied to MutexaGPT's mission of converting intuition into design. Note that, an LLM agent in MutexaGPT is a large language model (e.g.: GPT-4o[58], etc.) configured with a system prompt that instruct and regulate its behavior. In this section, "prompt design" refers specifically to the design of the system prompt that governs the agent's behavior, rather than the natural-language inquiry/input provided by the user. We therefore benchmark how different prompt-engineering choices affect agent performance under the same task definition and curated test set. The performance was evaluated against six different

prompt designs (Figure 2B) on the same curated test set of 200 inquiries (100 irrelevant and 100 relevant, with the latter split evenly between fully specified and underspecified cases). The inquiries were generated by the test-set curator and manually reviewed for correctness. Each prompt design was tested across three independent runs, and a design was marked as failing a test if any of the three runs failed.

Evaluation is based on two criteria: relevance (Figure 2B, top) and completeness (Figure 2B, bottom). The relevance metric tests whether **QuestionAnalyzer** correctly classifies user inquiries as relevant or irrelevant. A design is counted as passing only if all three independent runs correctly reject every irrelevant question and accept every relevant one; the reported performance reflects the proportion of designs meeting this standard. The completeness metric assesses whether the agent correctly identifies missing information in under-specified inquiries. We report this metric in three forms: (1) the straight pass ratio, representing designs for which all three runs behave correctly (i.e., the agent asks for exactly the information categories that are actually missing in the inquiry: protein, mutation, and/or property; orange bars); (2) the double-checking pass ratio, a more permissive measure that also counts cases where the agent correctly identifies missing information but unnecessarily re-asks for existing information—behavior that is overcautious yet harmless to downstream automation (light-blue bars, counted together with orange to form the effective pass rate); and (3) the consensus pass ratio, a relaxed criterion that deems a completeness test successful if at least two of the three runs pass, reflecting tolerance toward stochastic errors (dark-blue bars).

The benchmark results show that the *Full* prompt (see Table S2), which contains chain-of-thought[59-61] (CoT; i.e., the system prompt contains a list of instructions directing the agent to follow a logic chain) scaffold and seven carefully selected in-context learning[62] examples (ICL;

i.e., examples, as input and output pairs, are provided in the system prompt, also known as few-shot learning), emerges as the best performing design. It accepts or rejects queries correctly in 97.0% of cases, achieves an 71.0% strict completeness pass rate, and climbs to 97.0% once double-checks are counted; its consensus robustness stands at 94.0%. Removing the ICL snippets (*No ICL*) leaves the relevance pass rate almost unchanged (99.0%) but cuts the strict completeness pass rate to 41.0%, because the model becomes uncertain and compensates with many redundant queries (54.0% in double-check rate). Suppressing the explicit reasoning scaffold instead (*No CoT*) has a similar although slightly less severe effect: strict completeness pass falls to 45.0%. The *Only ICL* prompt, which only contains the seven examples and a single-sentence role description, recovers some ground (58.0% strict completeness pass rate) yet remains less stable. Augmenting the *Full* prompt with a one-line self-reflection instruction (*Full + self-check*) did not translate into a net win. Although the effective pass rate remains high (98.0%), the strict completeness pass rate falls by roughly eight percentage points and the proportion of redundant double-checks grows, as well as pulling the consensus robustness down from 94.0% to 91.0%. In other words, the extra self-check mostly prompts the agent to question information that is already complete, offering little practical benefit while adding conversational noise. At the opposite extreme, a minimalist specification (*Simplistic*), which only contains a sentence of role description and the output format in the system prompt, still rejects irrelevant questions (96.5%) but collapses on strict completeness (16.0%), succeeding in barely half the cases even though the looser consensus pass is measured (54.0%). Error analysis of the result of the Full QuestionAnalyzer shows that relevance failures are exclusively false negatives, primarily arising from overly vague inquiry (e.g., “What effect do ligands have on proteins?”), while effective-pass completeness failures are rare (3/100) and mostly

associated with marginally ambiguous property definitions (e.g.: “flexibility of …”) rather than systematic inability to identify incomplete inquires.

Three design lessons follow. First, CoT and ICL contribute independently and additively to the agent’s ability to solicit precisely the missing information, removing either one reduces the performance by 25-30 percentage points. Second, a cheap self-check clause provides an easy safety net against occasional lapses. Third, a blanket “self-check before replying” clause is not an automatic upgrade. In our hands, it lowered strict completeness and inflated redundant follow-ups. Third, minimal prompts are insufficient for complex reasoning. The Simplistic design, removing CoT and examples, collapsed to one-third completeness pass rate and showed the highest rate of redundant clarifications, underscoring the necessity of structured guidance for complex tasks (e.g., CoT, ICL, etc.). Without them, the model defaults to hesitation and repetitive questioning instead of reliable information gathering. Note that, while the Full prompt design performs better in our context, a richer system prompt carries a potential downside: it improves in-domain precision by constraining the agent, but also reduces flexibility on atypical or out-of-distribution requests by anchoring behavior to the included examples and interaction schema.[63, 64]

Besides **QuestionAnalyzer**, every agent in the MutexaGPT workflow is subjected to its own benchmark similar to the one described above, albeit with a single, production-ready prompt rather than the full suite of design variants explored above (Text S8). For MetricsPlanner, the test-set curator generates inputs (that match the format of the QuestionAnalyzer’s final output, with 100 cases asking about properties that can map to one of our supported metrics, and 100 that are not) from ground-truth metric/MD-model labels, and a case passes only when the agent’s recommended candidates contain the labeled metric-model pair or inform user when none of the supported metrics can correspond to the inquiry. For MutantPlanner, curator-agent generated

inputs contain a balanced set of unrestricted, pocket, protein-protein interface, distal (relative to the pocket) mutational libraries, and designated site mutations (10 for each, 50 in total). The correctness is judged by whether the generated EnzyHTP mutation pattern is semantically equivalent to the labeled library (i.e., they are parsed by EnzyHTP and directly compared for equivalence). For ResultExplainer, a set of JSON payloads are curated by a curator agent (containing "scientific question", "metrics", "results", "downloadables", and "metadata"; 100 in total). The outputs of ResultExplainer are evaluated by two judge agents, one assessing how well the output aligns to the scientific question from the input and the other assessing interpretation-boundary discipline (i.e., avoidance of overclaiming beyond the provided data), and overall pass requires satisfying both criteria. Full benchmark inputs, pass definitions, and per-run outputs are provided in Text S8 and SI.zip/agent_evaluation.

The results show that the prompt design for each agent is highly effective in their respective roles: for **WorkPlanningBoard**, MetricsPlanner on average passes 91% of its test cases, MutantPlanner achieves a 95% success rate; **ResultExplainer** met all evaluation criteria in 99% of the test cases. While not directly comparable, these performance levels are strong in the broader context of LLM benchmarks for scientific tasks (for reference, recent scientific-discovery benchmarks[65] typically report only 0.60–0.75 question-level accuracy across domains, while the best model in a question-clarification benchmark[66] reaches a maximum success rate of 60.1%). One can find details in Text S9 for error analysis of each agent, and SI.zip for spreadsheets that contain the test input, ground truth label, agent's output, the judge agent's output, and the pass/failure of every test case of each agent in MutexaGPT.

3. Results

This section presents two MutexaGPT-enabled enzyme-engineering campaigns that translate intuition (written in plain-English) into physics-based models, molecular simulation workflows, and eventually actionable design recommendations. First, a cavity-engineering task uses high-throughput MD ensemble analysis to pinpoint mutations that enlarge an active site, improving accommodation of bulkier, non-native substrates. Second, a cold-adaptation engineering task employs high-throughput MD to design linker variants that enhance enzyme activity at low temperatures; the resulting design was also experimentally validated. In both cases, a shared obstacle that often deters enzyme engineers from translating their intuition into design is the absence of quantitative and actionable guidance and the steep expertise required for high-throughput physics-based modeling. Below, we demonstrate how MutexaGPT addresses these challenges and leads to new scientific discoveries and understandings.

3.1 MutexaGPT for enzyme cavity engineering to modify substrate specificity: translation of empirical intuition into mutation design.

In the first case, we asked MutexaGPT to engineer the cavity of a halide methyltransferase toward catalyzing bulkier substrates for group transfer. Cavity engineering targets the shape and volume of the interior voids of an enzyme, and designs or screens mutations accordingly to modulate the enzyme's functions and properties. An empirical intuition in cavity engineering is "larger cavity allows the enzyme to accommodate larger substrates with enhanced promiscuity". Although static active-site structures have enabled successful cavity-guided engineering in several systems,[32, 33, 44, 67, 68] such intuition is difficult to apply broadly. Simple mutations that appear to enlarge a cavity often fail because conformational dynamics can drastically reshape the pocket. As

a result, most cavity engineering efforts still rely on screening of a large mutational library, consuming substantial experimental resources.

MutexaGPT allows the user to translate such intuition into concrete physics models and testable designs through high-throughput MD simulations. The user can start a cavity engineering task by sending a plain-English request in the chat. In our demo, the user sent "*Help me find mutations that make the active site larger*" (Figure 3A). MutexaGPT evaluated the request and found missing scientific details that were essential to the subsequent modeling (i.e., a target enzyme structure and a list of desired mutations). As such, it asked the user to provide a PDB file of the target enzyme and a description of mutations that the user wants to include (see full conversation log in Text S1). To address this request, the user uploaded the PDB structure of the wild-type halide methyltransferase and supplied details of a mutation library (i.e., 600 combinatorial mutations from five sites, see Text S1) previously constructed for this enzyme by Schülke et al.[69] The experimental sequence–activity data derived from the library screening against five substrates of diverse sizes allow us to investigate how changes in cavity volume influence enzyme activity later on. In real-world applications, users can initiate their protein engineering campaigns with MutexaGPT by simply describing their desired mutation library, such as "1000 random mutations across the enzyme", "single-point saturation mutations within 5 Å of the heme cofactor", and so on.

After verifying that the user's input contained all necessary information to proceed with the modeling, MutexaGPT planned a relevant molecular simulation task: calculating the cavity volume of the active site of each mutant from MD-sampled conformational ensemble of the holo-enzyme (see Method section for full computational details). With the user's approval, the backend executed the corresponding computational workflow. For each of the 600 multi-point mutants in

the library, MutexaGPT generated a 50 ns MD trajectory and analyzed the distribution of active site volumes across the trajectory. The length of each MD simulation is an option that the user can specify in the frontend before submitting the job. We used 50 ns here to mimic a realistic scenario where most enzyme engineers do not have access to extensive computational resources and need to balance computational cost and sampling adequacy. Even with this moderate MD simulation length, it would still demand 30 μs sampling (i.e., 600 mutants × 50 ns) to cover the entire mutation library.

Eventually, MutexaGPT addressed the user's initial request, "*Help me find mutations that make the active site larger*", by ranking all 600 variants according to their ensemble-averaged cavity volumes and identifying the 10 candidates showing the greatest increases in volume (Figure 3D, see screenshot of the Summary page in Figure S3), thereby turning the original empirical intuition into testable mutation designs. To validate these designs, we compared MutexaGPT's top-ranked mutations with experimentally characterized variants reported by Schülke et al.[69] Among the ten top-ranked variants, four were confirmed experimental hits based on Schülke et al.'s prior study, yielding a 40% hit rate and around 4-fold improvement over the baseline (see Text S12 for details about the baseline). To investigate whether 50 ns MD simulations are sufficient for cavity-based smart library construction, we systematically examined how simulation length influences hit enrichment and ranking fidelity (Figure 3E). Very short simulations (5-10 ns) fail to enrich hits beyond the baseline expectation (i.e., ~1 hit), whereas simulations of 15 ns or longer consistently identify at least 4 hits, corresponding to about a 4-fold enrichment. Meanwhile, the Spearman rank correlation with the 50 ns reference increased monotonically with simulation length, indicating progressively more stable cavity-volume estimates. Together, these results show

that moderate-length (i.e., 15 ns or longer) MD simulations provide reliable quantitative guidance for mutation prioritization in cavity-engineering campaigns.

A case study of the top mutant, L39H/V11F, illustrates how mutations induce a drastic upshift in the active site's cavity volume distribution compared to the wild-type (WT) enzyme (Figure 3F), thereby reshaping its substrate preference. The cavity volume for the most probable WT conformation is approximately 542 $Å^3$, corresponding to a closed and compact active site. In contrast, the L39H/V11F mutant's cavity volume distribution is centered at a much larger value of ~2870 $Å^3$, reflecting a significantly larger and more accessible cavity. This transition from a closed to an open cavity state is primarily driven by the altered conformational dynamics of two key active site loops (residue 1-11 and residue 36-46; highlighted in red, Figure 3F). This cavity expansion in the active site directly translates to improved substrate promiscuity toward a variety of substrates, especially the sterically bulkier ones. While the WT is evolved by the nature to catalyze the methyl transfer from iodomethane to S-Adenosyl-L-homocysteine (SAH), L39H/V11F showed an substantially increased specific activity (relative to the WT) for bulkier substrates[69] such as iodoethane (10-fold activity improvement), 1-iodopropane (49-fold), 1-iodobutane (50-fold), (iodomethyl)cyclopropane (50-fold), and (2-iodoethyl)benzene (29-fold). These results demonstrate that MutexaGPT can effectively convert intuition into physics models and design through high-throughput molecular modeling, identifying function-enhancing mutations that alter substrate preference.

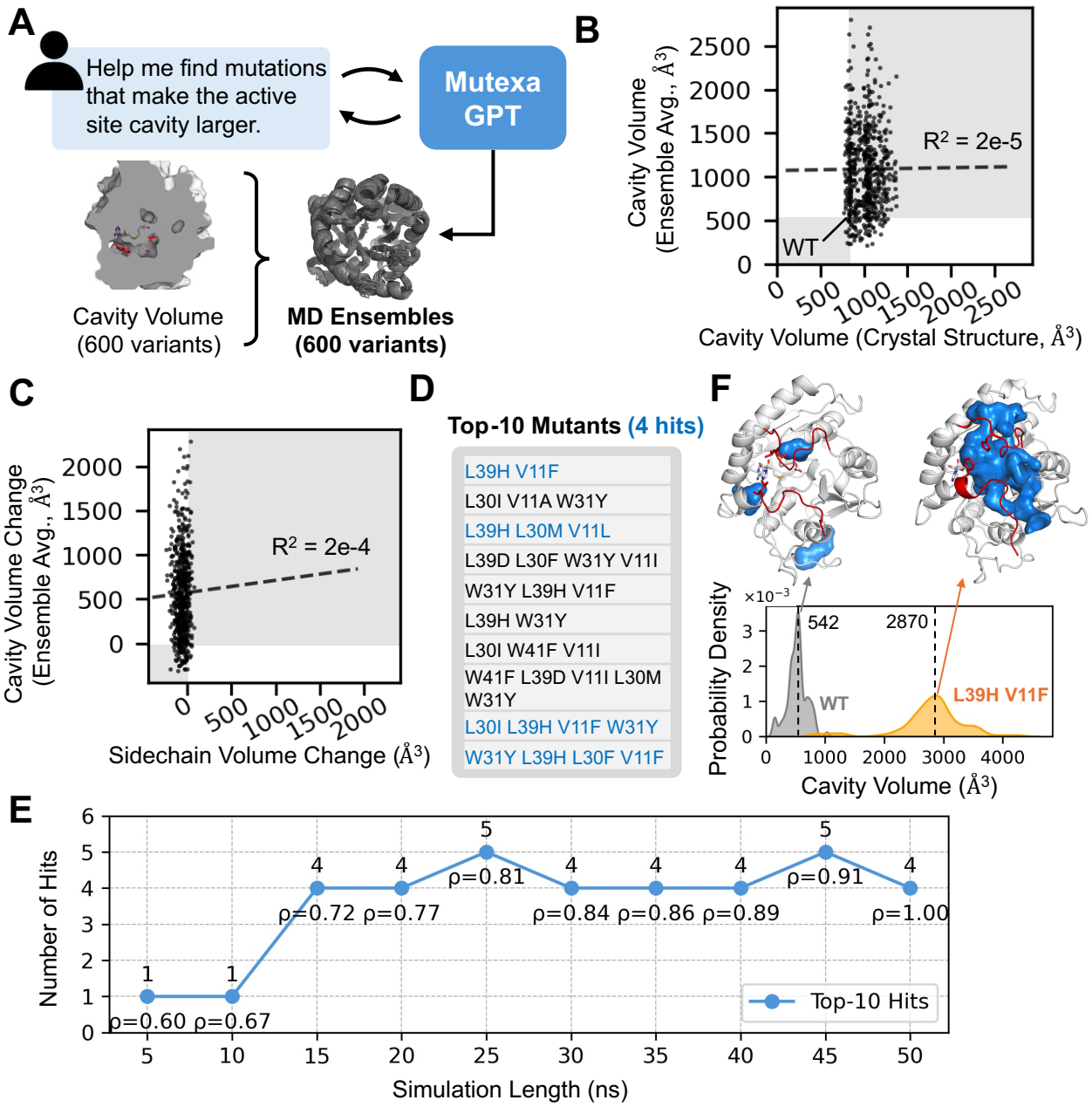


Figure 3. An overview of MutexaGPT-guided cavity engineering. (A) Workflow overview: the user sent a plain-English request; MutexaGPT asked for more information and discussed with the user about the modeling plan; MutexaGPT performed high-throughput MD simulations and calculated cavity volumes for the library of 600 mutants. (B) Correlation between cavity volumes calculated from direct geometric perturbation of static crystal structures (see Method for details)

and those calculated from MD ensemble averages; both for the library of 600 mutants. (C) Correlation between the change in side-chain volume versus the change in ensemble average cavity volume for the library of 600 mutants. (D) MutexaGPT's final list of 10 mutants with the largest predicted cavity volumes. Experimentally confirmed hits are highlighted in blue; these mutants show activity with bulkier substrates[69]. (E) The top-10 hit-rate plot against the MD simulation length of each mutant. The spearman correlation coefficient (ρ) is noted under each data point. (F) Cavity volume distribution of a top-ranked mutant, L39H/V11F against that of the WT. The probability distribution of the mutant's cavity volume (L39H/V11F, shown as orange) is shifted to a much larger value (~2870 Å^3) compared to the wild-type (WT, shown as grey) enzyme (~542 Å^3). Representative structures from the distribution center (dotted line, labeled) show how mutations (stick, red) induce a conformational change in an active site loop (cartoon, red), leading to a significantly larger and more open active site (surface, blue). The kernel density distribution is fitted using the *gaussian_kde* function from the *scipy* package. The fitting data are obtained from the cavity volume calculations of 100 snapshots sampled from the MD trajectory.

Given the computational expense of MD simulations, an important question arises: is MD truly necessary for quantitative cavity engineering, or can the ensemble-averaged cavity be reliably inferred from a single static structure as a shortcut? Using the MD trajectories generated by MutexaGPT for 600 variants, we first examined the correlation between the ensemble-averaged cavity volume from MD and that calculated from a single static PDB structure (with optimized side-chain rotamers), assessing whether static structures can adequately serve as surrogates for dynamic ensembles in computational enzyme engineering. The results show no correlation ($R^2 \approx 2\times10^{-5}$) between the static and MD-averaged volumes, where the cavity volumes derived from MD trajectories distribute significantly broader than from static PDB structures (standard deviation:

488 Å^3 vs. 145 Å^3, Figure 3B). This lack of correlation originates from the presence of cryptic cavities that emerge only in the MD ensemble through conformational fluctuations. These transient pockets remain closed or buried in the crystalline state represented by a single static PDB structure. As a second test, we evaluated another common cavity engineering shortcut that estimates the magnitude of cavity expansion in the mutant from the net decrease in side-chain volume due to point mutations. Specifically, we tested the correlation between the change in side-chain volume and the change in ensemble-averaged cavity volume derived from MD trajectories. This approximation also failed to show any meaningful correlation ($R^2 \approx 2\times10^{-4}$; Figure 3C), indicating that cavity change originates from collective and dynamic rearrangements of the protein active-site structure rather than from simple steric subtraction, which explained why the traditional qualitative intuition-based design often fails. Together, these two tests demonstrate that simple approximations cannot capture the intrinsic conformational plasticity of enzyme cavities, underscoring the necessity of physics-based modeling, specifically, MD simulations for predictive cavity engineering.

To further test whether MutexaGPT remains useful when prior information about mutational hotspot is minimal, we designed a test where the mutation library is selected solely based on a distance cutoff from the substrate. In this new case, the user starts by asking “help me find mutations that make my enzyme faster for an non-native substrate ethyl iodide.” and during the user’s conversation with the QuestionAnalyzer, the mutation library was defined by user as “I want to explore all single point mutations with in 4A of ETI.”, where ETI is the 3-letter name of the target substrate (i.e., ethyl iodide) in user’s PDB files. After the conversation, MutexaGPT plans a workflow that calculates the folding free energy and the electric field strength projected to the breaking C–I bond for each enzyme mutant. With the simulation data, MutexaGPT summarizes

the results by filtering candidates with folding free energy and ranking by electric field, yielding a top-10 mutant list, of which 3 mutants are experimentally active in a previous screening reported by Tang *et al.*[70], corresponding to a hit rate of 30%. Notably, this 30% value serves a lower bound because 4 mutants on this list were not experimentally characterized (not labeled in our test) and could be a hit. Overall, this low-prior-knowledge setup results in a similar prediction performance to the cavity engineering example, supporting MutexaGPT's ability to convert broadly-defined, less-informed design intent into actionable screening priorities.

3.2 MutexaGPT for amino acid linker engineering to enhance cold adaptation of bidomain enzymes: translation of statistical intuition into mutation design.

In the second case, we integrated MutexaGPT with experimental assays to engineer cold-adapted variants of a bidomain amylase by redesigning its interdomain linker, aiming to enhance enzymatic activity at low temperatures. Cold-adapted enzyme engineering, the process of identifying enzyme variants that maintain high activity at colder conditions (e.g., 0–15 °C), is crucial for low-temperature laundry[71], food processing[72, 73], bioremediation[74, 75], and single-cell sequencing[76], serving as a cornerstone for modern biomanufacturing and biotechnology. From MD modeling, we uncovered a "statistical intuition" that suggests the correlation between increased interdomain separation and enhanced cold adaptation in bidomain amylases and cellulases.[30, 77] This separation is quantified by the spatial proximity between the carbohydrate-binding module and the catalytic domain, defined as the domain separation index (DSI, Figure 4A). Guided by this statistical intuition, MutexaGPT can help user establish a closed-loop computational–experimental workflow. In this workflow, MutexaGPT identified linker variants of a maltotetraose-forming

amylase (Figure 4B) predicted to exhibit increased DSI, and these variants were subsequently tested for enhanced cold adaptation through enzyme activity assays.

The user initiated the task by sending the following request to MutexaGPT: "*I am studying a bidomain enzyme. This enzyme showed cold-adaptation behavior, that is, the catalytic activity reduces much slower at lower temperatures. We found that just by changing the linker, the cold-adaption changes, can you help me perform some modeling to find linkers that provide stronger cold-adaption?*". Similar to the first case, MutexaGPT asked for clarification of the target enzyme by requesting PDB files and the linker sequences of interest. The user provided the AlphaFold2-predicted structure of the mesophilic *Pseudomonas saccharophila* amylase (psA), as well as 10 psA variants with different linker sequences selected from natural peptide linkers in LinkDB[78] and SynLinker[79] databases based on the linker topology from our previous study (see Table S3 for linker sequences and Text S13 for the selection workflow). With this information, MutexaGPT devised a plan to identify linker variants predicted to exhibit higher DSI values than the wild type by performing 50 ns MD simulations for each candidate. Upon the user's confirmation, MutexaGPT executed the simulations, analyzed the trajectories, and summarized the DSI results for all variants (see Text S2 for the full dialogue; Table S3 for the list of DSI values).

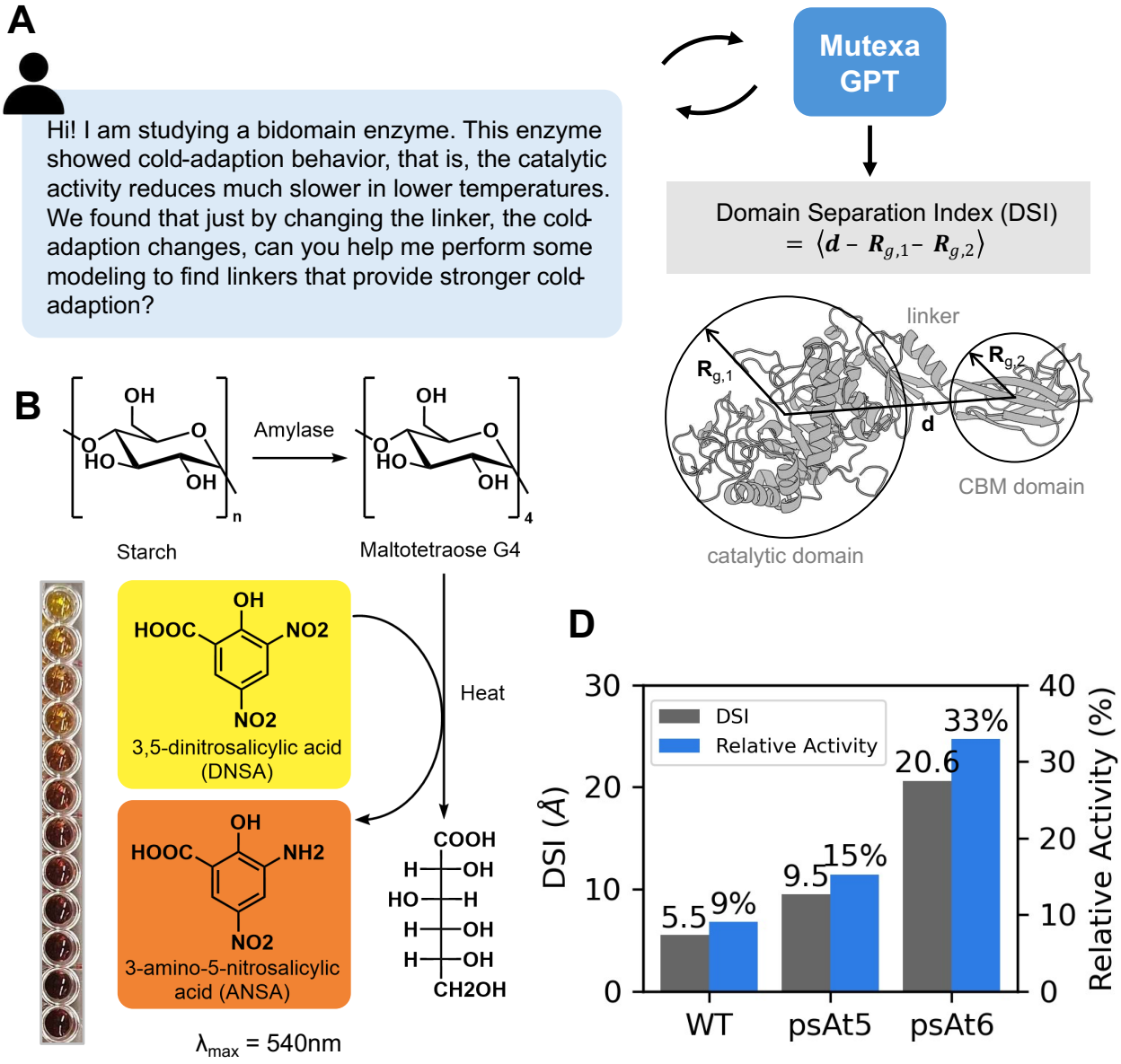

A
Hi! I am studying a bidomain enzyme. This enzyme showed cold-adaption behavior, that is, the catalytic activity reduces much slower in lower temperatures. We found that just by changing the linker, the cold-adaption changes, can you help me perform some modeling to find linkers that provide stronger cold-adaption?
Mutexa GPT
Domain Separation Index (DSI)
$= \langle d - R_{g,1} - R_{g,2} \rangle$
linker
$R_{g,1}$
$R_{g,2}$
d
CBM domain
catalytic domain
B
Amylase
Starch
Maltotetraose G4
Heat
3,5-dinitrosalicylic acid (DNSA)
3-amino-5-nitrosalicylic acid (ANSA)
λ_{max} = 540nm
D
DSI
Relative Activity
DSI (Å)
Relative Activity (%)
5.5
9%
9.5
15%
20.6
33%
WT
psAt5
psAt6
C
psAt6
psAt5
WT
psAt3
psAt10
psAt9
psAt8
psAt2
psAt1
psAt7
psAt4
ENTRGFHG
GSGDGGGNDGGEGGL
ITAKDE
Probability Density
Domain Separation Index (Å)

Figure 4. MutexaGPT-guided linker design for cold-adapted bidomain amylases. (A) Workflow overview: the user sent a request in plain text; MutexaGPT collected missing information from user, discussed and planned the modeling, performed MD for WT and 10 linker-replaced variants, and calculated the domain separation index (DSI). DSI is defined as $\langle d - R_{g,1} - R_{g,2} \rangle$ over the MD ensemble, where d is the distance between the two domain centroids and $R_{g,1}/R_{g,2}$ are the radii of gyration of the two domains, respectively. Unit: Å. (B) The experimental assay for the amylase activity: specific activity is measured via the 3,5-dinitrosalicylic acid (DNSA) reducing-sugar assay; cold-adaptation is evaluated as the "relative activity" = activity at 0 °C divided by activity at 45 °C. (C) The DSI distribution of the *Pseudomonas saccharophila* amylase wild type and its ten linker variants. For variants with the largest (psAt6) and smallest (psAt4) average DSI and the wild type, a representative structure of the distribution peak is attached. (grey: catalytic domain, red: linker, dark grey: CBM domain) The linker sequences are noted next to each structure. Note that the wild type shows two equally distributing peaks, see Figure S10 for representative structures of both peaks. The kernel density distribution is fitted using the *gaussian_kde* function from the *scipy* package. The fitting data are obtained from the DSI calculations of 100 snapshots sampled from the MD trajectory of each variant. (D) The experimental results of MutexaGPT suggested variants: two linkers (psAt5, psAt6) predicted to have higher DSI than WT also showed increased relative activity in experiments. (See full experimental and computational detail in Method Section)

The DSI distributions for all ten linker variants (psAt1–psAt10) display a distinct pattern (Figure 4C). We observe that variants with extreme DSI values tend to adopt more stable interdomain conformations. psAt6, for example, shows the highest average DSI (20.6 Å) with a very narrow distribution (variance: 0.2 Å), while psAt4, psAt7, psAt1, psAt2, psAt8, and psAt9

cluster near 0 Å with similarly tight distributions (variance: 0.1–0.2 Å). In contrast, the wild type and variants with moderate DSI values, including psAt3, psAt10, and psAt5, exhibit much broader DSI distributions (variance: 1.4–4.8 Å). These patterns arise from distinct linker scaffolds. In high-DSI variants, the two domains are well separated and connected by a linker that predominantly adopts an α-helical conformation (Figure 4C, right), restricting relative domain motion. In low-DSI variants, the two domains pack closely and form extensive interdomain contacts, likewise limiting flexibility (Figure 4C, left). By contrast, variants with intermediate DSI values, as well as the wild type, feature a more disordered linker that samples diverse orientations between the domains, leading to broader DSI distributions.

Two variants, psAt5 (blue) and psAt6 (deep blue), center around 9.5 Å and 20.6 Å, respectively, significantly exceeding the DSI range of the wild type (around 5.5 Å). These two variants were therefore suggested by MutexaGPT for further experimental characterization (Figure 4B). We then experimentally assessed the specific activity of these two variants at different temperatures by measuring the amount of reducing sugar produced from starch using a 3,5-dinitrosalicylic acid (DNSA) assay[80] (Figure 4B). The magnitude of cold adaptation was assessed by calculating the relative activity, defined as the ratio of the enzyme's activity at 0°C to its activity at 45°C. The experimental results (Figure 4C) show that psAt5 and psAt6 exhibit 1.7-fold and 3.7-fold increases in relative activity, respectively, compared to the wild-type enzyme, indicating that both variants involve enhanced cold adaptation (See Table S4 for activities under each temperature). Together, these results validate MutexaGPT's capability to seamlessly convert statistical-intuition into physics-based models, integrate with experimental workflows, and conduct physics-guided engineering of cold-adapted enzymes.

4. Discussion

In this work, we developed MutexaGPT as an intuition-to-design translator for enzyme engineering. The challenge of intuition-to-design conversion was addressed by integrating human intelligence with machine intelligence through a multi-LLM-agent system, where MutexaGPT assists enzyme designers and engineers to reframe their intuition-driven ideas into physics-based structure-function hypotheses via the QuestionAnalyzer agent, converts the hypothesis into a high-throughput molecular simulation task via the MetricsPlanner and MutantPlanner agent from the WorkPlanningBoard, operates on the modeling by calling our high-throughput molecular modeling toolkit EnzyHTP, and eventually analyzes the results using the ResultExplainer agent. As such, the user can convert their plain English requests into a "smart" library of enzyme mutant designs through the MutexaGPT web application. Notably, while MutexaGPT ranks and filters enzyme variants based on computed metrics, it is not a predictive AI tool, but instead an integrated system for automation and intuition-translation.

We evaluated the performance of MutexaGPT's core agents (i.e., QuestionAnalyzer, WorkPlanningBoard, and ResultExplainer) using an automated, LLM-based benchmarking framework. Curator agents generate diverse, manually verified natural-language test cases, while classifier agents assess outputs against ground truth to quantify relevance and completeness. Using QuestionAnalyzer as a representative and most intuition-critical example, we found that prompt designs combining structured reasoning scaffolds and in-context examples consistently outperform minimal or heuristic prompts, which suffer from incomplete information gathering and redundant clarification. Similar evaluations show that WorkPlanningBoard and ResultExplainer also achieve high reliability with their production-ready prompts. Overall, these assessments demonstrate that MutexaGPT's agent design and prompt strategies support the translation of user intuition into actionable decisions.

Eventually, we demonstrated MutexaGPT in two enzyme engineering case studies. First, we engineered the cavity of a halide methyltransferase to accommodate bulkier substrates. By converting a "cavity"-related empirical intuition into a physics-guided design, MutexaGPT yielded a smart library that achieves 40% hit rate with around 4-fold improvement over the baseline. Second, we engineered a bidomain amylase to enhance its activity under low temperature conditions. By translating a statistical structure-function intuition into physics-guided design, MutexaGPT assisted the user to identify amylase variants that exhibit up to 3.7-fold increase in cold adaptation at 0 °C in our experimental validation. Complementary to existing screening-heavy approaches, we believe MutexaGPT will enable a new paradigm for enzyme engineering where human creativity is seamlessly integrated with computational simulation to make new discoveries and designs.

How does MutexaGPT work beyond the two case studies? Currently, MutexaGPT supports high-throughput calculations of a list of enzyme function-related metrics: electric field strength, substrate positioning index (SPI), active site RMSD, MM-PB/GBSA ligand binding energy, relative folding free energy, cavity volume, and domain separation index (DSI). Beyond the non-native substrate accommodation (metric: cavity volume) and cold adaptation (metric: DSI) as demonstrated in the current study, this full metrics list can be applied to a large variety of enzyme engineering tasks. For example, the cavity volume metric guides the engineering of enantioselectivity.[33, 44, 45] Electric field strength can guide the engineering of rate enhancement for enzymes that catalyze reactions involving a substantial change of charge and polarity state from the reactant state to the transition state.[20-25] MMPB/GBSA ligand binding energy can guide the engineering of enzyme with enhanced binding affinity to the substrate or reduced product inhibition.[81] SPI can guide the engineering of catalytic efficiency for enzymes where precise

substrate positioning is critical for catalysis.[82] Relative folding free energy can be used to guide the engineering of enzyme stability. Combining these metrics is expected to address more complex challenges, such as studying drug resistance mutations. Note that a practical limitation of MutexaGPT also comes from this list of computable metrics. If a property of interest from the user cannot be mapped to a currently supported EnzyHTP-computable metric, MutexaGPT cannot produce a physically grounded simulation answer and should instead return an explicit unsupported-property notice. Nevertheless, MutexaGPT is designed to be extensible, where new metrics can be easily (by changing just two files if they are supported by EnzyHTP) added in the future to support more enzyme engineering tasks. (One can refer to Table S1 for currently supported metrics; We should note that the underlying EnzyHTP engine is developed with interface modules to common software such as Amber[83] for trajectory analysis, Gaussian[84]/xTB[85] for quantum mechanics, Multiwfn[86] for wavefunction analysis, Rosetta[87] for biophysics, etc. and thus inherits the metrics-support-boundary of these software) Overall, we envision MutexaGPT can be applied to a wide range of enzyme engineering tasks by leveraging the existing metrics and adding new metrics in the future.

Finally, we would like to discuss MutexaGPT's level of autonomy as an AI agent. Gao et al.[88] summarized the three levels of autonomy in AI agents: *assistant level*, which works with restricted range of tools (e.g., web search, docking, etc.) and relies on domain experts to define scientific hypotheses, propose theoretical models, and design specific tasks; *collaborator level*, which collaboratively refine hypotheses with human, design computational tasks, and interpret results; and *scientist level*, which develops and extrapolates scientific hypotheses beyond the scope of prior research, autonomously designs and performs experiments, synthesize concepts beyond summarizing the results, and eventually make new scientific discoveries all by itself. Existing

LLM-agentic molecular simulation tools mostly operate at the assistant level, designed to streamline the molecular modeling process for domain experts.[88] In contrast, MutexaGPT is featured by its functionalities that one would typically expect from a collaborator, such as refining intuition into physics-based hypotheses, executing the htMS, and generating design recommendations. For protein engineers with little-to-no experience in computational modeling, MutexaGPT, while remaining an assistant-level system, takes a step toward becoming a collaborator-level agent for enzyme engineering.

However, the level of hypothesis refinement is still limited to correcting non-scientific statements, retrieving missing information for a vague request, and mapping general properties of interest to specific computable metrics. Presumably, a complete collaborator level agent should also be able to proactively generate alternative mechanistic hypotheses rather than merely refining user-proposed ones, design and adapt multi-step computational and experimental workflows to discriminate among these hypotheses, integrate heterogeneous evidence across modalities to revise or reject initial ideas, and autonomously propose new lines of inquiry when the accumulated data reveal gaps or inconsistencies. Such limitation comes from the limited reliability and robustness of current foundation models for complex, open-ended scientific reasoning, the lack of mature methods for uncertainty quantification and self-evaluation, and the scarcity of standardized, high-quality multimodal datasets and workflows needed to safely couple *in silico* molecular modeling with real-world experimentation. With the unprecedented evolving speed in AI technology and foundation models, we expect these limitations to be progressively alleviated and MutexaGPT will move closer to a genuinely collaborator-level agent.

5. Method

5a. Computational Method

Protein structure preparation. In the cavity engineering use case of MutexaGPT, the uploaded wild-type structure of *Aspergillus clavatus* methyltransferase (*acl*-MT) is obtained from the AlphaFold Protein Structure Database (UNIPROT ID: A1CIS5). The S-Adenosyl-L-homocysteine (SAH) cofactor is placed in the structure referencing the binding position in the crystal structure of a homologous anion methyltransferase from *Arabidopsis thaliana* (PDB ID: 3LCC). In the cold-adaptation engineering use case of MutexaGPT, the uploaded wild-type structure of *Pseudomonas saccharophila* amylase (psA) is obtained from our previous study[30], which used AlphaFold2[89] via the ColabFold[90] server to predict the protein structure from its sequence. The sequence and parameters used for prediction are listed in Table S5 and Table S6. The rest of the structure preparation procedures, including determining the protonation states of titratable residues, generating structures of enzyme mutants, etc., are planned and performed automatically by MutexaGPT, powered by EnzyHTP[56, 57] in the backend. (via the *protonate_stru(), mutate_stru(), assign_mutant()*, etc. APIs in EnzyHTP. See full API parameters and underlying algorithms in https://github.com/ChemBioHTP/EnzyHTP-GPT/blob/main/flask-server/templates/slurm_run/md_main_script.py and https://github.com/ChemBioHTP/EnzyHTP)

Molecular dynamics. All MD simulations in this study are performed autonomously on the MutexaGPT server. In the server backend, EnzyHTP is used to perform the MD simulations via the *equi_md_sampling()* API. This API uses a default force field combination of AMBER ff14SB[91] for the protein, TIP3P[92] for water, and the generalized AMBER force field[93, 94] for ligands with the AM1-BCC[95] charge model. This API, by default, solvated the enzyme in a periodic truncated octahedron box with a 10 Å buffer of water and were neutralized with Na^+ or Cl^- counterions. The production time was set to 50ns via the web GUI in the frontend of MutexaGPT. In the

equi_md_sampling() API, via the underlying interface with AmberMD[96], each 50ns MD production run is preceded by 1) 20000 steps energy minimization of the entire simulation system using the steepest descent method for the first half and the conjugate gradient method for the other half. 2) 50 ps heating from 0 to 300 K with constant volume and the backbone C_α, C, and N of the amide group restrained with a weight of 2 $kcal \cdot mol^{-1} \cdot Å^{-2}$. 3) 500 ps equilibration at 300K and 1 atm with the same backbone restraint. 4) 500 ps equilibration at 300K and 1 atm without the backbone restraint. Chodera automated equilibration detection scheme[97] is used to analyze the equilibration statistics in the backend and the results are reported by ResultExplainer in the final Summary page. All MD simulations were performed with a time step of 2 fs with the SHAKE[98] algorithm applied. The API by default uses the Langevin thermostat[99] and Berendsen barostat[100] to control the temperature and pressure, respectively. (Source code: https://github.com/ChemBioHTP/EnzyHTP-GPT/blob/main/flask-server/templates/slurm_run/md_main_script.py)

Cavity volume calculation. All ensemble cavity volume calculations in this study are performed autonomously on the MutexaGPT server. In the server backend, MutexaGPT calls the *ensemble_cavity_volumes()* API from EnzyHTP when the cavity volume calculation is needed. The *ensemble_cavity_volumes()* API allows the upstream calls to specify the target cavity using a list of composing residues, the containing ligand, or an existing Cavity() object. For *acl*-MT, the active site cavity is defined as composed of residues 7, 11, 27, 40, 41, 43, 165, 168, 169, 199, 210, 211, and 289 (SAH). The API then tracks this cavity along the MD trajectory and obtain the cavity mesh as a *pyvista.PolyData* object[101], for each frame, using the default, Voronoi-diagram-based, MOLE2 algorithm via the corresponding MOLE2[102] interface in EnzyHTP. The final cavity volume is obtained through the *.volume* attribute of the mesh object. For the static structures cavity

volumes, we used the same protocol in EnzyHTP but with the static mutant structure as input. These structures are generated by replacing the mutating side chains in the wild-type structure and search through a rotamer library to find the optimal side-chain position. (using the *mutate_stru()* function in EnzyHTP)

Domain separation index (DSI) calculation. All DSI calculations in this study is performed autonomously on the MutexaGPT server. In the server backend, MutexaGPT calls the *dsi()* API from EnzyHTP when needed. For each MD trajectory, the *dsi()* API calls cpptraj to calculate, for each frame, the distance between the center of mass of the designated domains and the radius of gyration of each domain (R_g). Then DSI = domain distance – ($R_{g,1}$ + $R_{g,2}$).

Configurations of the large language model (LLM). All LLM agents used in this study is powered by the OpenAI Python API, specifically, the *Responses* API for the native agentic tool call support from OpenAI. The LLM version is set to *gpt-4o-2024-11-20* for the consistent performance besides for ResultExplainer which is set to *gpt-5.4* for large context window. (temperature: 1.00; Top P: 1.00) Note that users can also choose the LLM model they want to use on the MutexaGPT website. (current options include: GPT-4o, GPT-5.4, Qwen3.6 Plus, MiniMax-M2.5, GLM-5, Kimi K2.5, GLM-4.7, Deepseek v3.2, Gemma 4 26B A4B IT, Gemma 4 31B IT) When using the OpenAI models, users need to provide their own OpenAI API keys. Each LLM agent is equipped with tools that allows them to interact with system in the backend and the user experience in the frontend. (see design details in Text S7 and system prompts in Table S2)

5b. Experimental Method

Bacterial Strains, Plasmids and Cloning. Genes encoding psA and psA variants (psAt5 and psAt6) were synthesized by Integrated DNA Technologies (Morrisville, NC). We amplified the

synthesized genes through polymerase chain reaction using Q5 High-Fidelity Master Mix (New England Biolabs). We assembled the amplified genes and pET-22b vector using Gibson Assembly Master Mix (New England Biolabs). We transformed the recombinant plasmids into Escherichia coli JM109 and E. coli BL21 for DNA manipulations and protein expression, respectively.

Protein Expression. A single colony from the transformed cells was inoculated into Luria-Bertani medium (5 mL) supplemented with ampicillin (50 μg mL−1) and grown at 37 °C overnight with agitation at 220 rpm. The overnight culture (2 mL) was then inoculated into Terrific-Broth medium (200 mL) supplemented with ampicillin (50 μg mL−1). The culture was incubated at 30 °C with agitation at 160 rpm for 72 hours. After incubation, the cells were removed by centrifugation at 10,000 × g for 30 minutes at 4 °C, and the supernatant, which contained crude enzymes, was collected for the thermal adaptation characterization.

Thermal Adaptation Characterization. The hydrolytic activities of amylases were measured by determining the reducing sugars released from the hydrolysis of starch using the 3,5-dinitrosalicylic acid (DNS) method[80]. The reaction mixture containing 0.2 mL of the enzyme and 1.8 mL of 1% (w/v) potato starch (Fisher Scientific) in 15 mM PBS buffer (pH 7.0) was incubated at 0 °C and 45 °C, separately. After 15 minutes, the reaction was quenched by adding 2 mL of DNS reagent and boiled for 5 min. The absorbance was measured at 540 nm. One unit of hydrolytic activity was defined as the quantity of enzyme that released reducing sugar equivalent to one μmol of glucose per minute under the assay conditions. Relative activity was quantified as the percentage ratio of enzymatic activity observed at 0 °C to that at 45 °C (the optimal temperature of psA). Three biologically independent replicates were used to calculate mean and standard deviation.

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to the lead contact, Zhongyue J. Yang (zhongyue.yang@vanderbilt.edu)

Materials availability

Example dialog between user and MutexaGPT for setting up the cavity study; Example dialog between user and MutexaGPT for setting up the cold-adaption study; The design of each individual LLM-agent; The evaluation system of LLM-agents; An example output of ResultExplainer; Topology-based selection of psA linkers; System prompts of all LLM-agents involved in this study; Evaluation results for LLM-agents in MutexaGPT; Linker sequences used in cold-adaption engineering and their calculated Domain Separation Index (DSI); The DSI distribution of the *Pseudomonas saccharophila* amylase wild type and its ten linker variants; Amylase activity of psA wild-type, psAt5, and psAt6 under 0 °C and 45 °C; Full sequences of psA variants; Parameters used for structure prediction using ColabFold server. (Supporting Information PDF)

Data and code availability

The MutexaGPT webapp is available at https://enzyhtp.app.vanderbilt.edu/. To access the webapp, one can use the account name: mutexatest@testemail.com along with the password: mutexatest111. All test cases can be found under this account. The source code of MutexaGPT and the evaluation system of LLM-agents in MutexaGPT is available from https://github.com/ChemBioHTP/EnzyHTP-GPT/. EnzyHTP is available from https://github.com/ChemBioHTP/EnzyHTP. AMBER is available from http://ambermd.org/. The original input file for all MD involved is available from Zenodo (https://doi.org/10.5281/zenodo.17905121).

ACKNOWLEDGEMENT

This research was supported by the startup grant from Vanderbilt University and the fellowship of Vanderbilt Institute of Chemical Biology. The PI thanks the sponsorship from Rosetta Commons Seed Grant Award. This work was also supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number R35GM146982. This work used SDSC Dell Cluster with AMD Rome HDR IB at Expanse from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants BIO200057.[103]

AUTHOR CONTRIBUTIONS

Conceptualization: Q. S., Z. J. Y.; Methodology: Q. S., X. R., Z. J. Y; Software (frontend): Q. S., Y. Z., S. S, K. N.-P., R. Y., H. X.; Software (backend, database): Q. S., Y. Z., S. S; Validation: Q. S., Z. J. Y; Investigation (Computational Data Generation): Q. S., Y. Z.; Investigation (Kinetic Experiment): N. D.; Writing (original draft): Q. S., Z. J. Y.; Writing (review & editing): Q. S., Z. J. Y.; Supervision: Z. J. Y.

DECLARATION OF INTERESTS

The authors declare no competing financial interest.

REFERENCES

1. Wason, P. C.; Evans, J. S. B., Dual processes in reasoning? *Cognition* **1974**, *3*, 141-154.
2. Stanovich, K. E.; West, R. F., Advancing the rationality debate. *Behavioral and brain sciences* **2000**, *23*, 701-717.
3. Ao, Y. F.; Pei, S.; Xiang, C.; Menke, M. J.; Shen, L.; Sun, C.; Dörr, M.; Born, S.; Höhne, M.; Bornscheuer, U. T., Structure‐ and Data‐Driven Protein Engineering of Transaminases for Improving Activity and Stereoselectivity. *Angew. Chem. Int. Ed.* **2023**, *62*.

4. Yang, L.-C.; Deng, H.; Renata, H., Recent Progress and Developments in Chemoenzymatic and Biocatalytic Dynamic Kinetic Resolution. *Organic Process Research & Development* **2022**, *26*, 1925-1943.
5. Kelly, S. A.; Pohle, S.; Wharry, S.; Mix, S.; Allen, C. C. R.; Moody, T. S.; Gilmore, B. F., Application of ω-Transaminases in the Pharmaceutical Industry. *Chem. Rev.* **2018**, *118*, 349-367.
6. Guo, F.; Berglund, P., Transaminase biocatalysis: optimization and application. *Green Chemistry* **2017**, *19*, 333-360.
7. Fuchs, M.; Farnberger, J. E.; Kroutil, W., The Industrial Age of Biocatalytic Transamination. *Eur. J. Org. Chem.* **2015**, *2015*, 6965-6982.
8. Slabu, I.; Galman, J. L.; Lloyd, R. C.; Turner, N. J., Discovery, Engineering, and Synthetic Application of Transaminase Biocatalysts. *ACS Catal.* **2017**, *7*, 8263-8284.
9. Gaines, C. S.; Piccirilli, J. A.; York, D. M., The L-platform/L-scaffold framework: a blueprint for RNA-cleaving nucleic acid enzyme design. *RNA* **2020**, *26*, 111-125.
10. Zhang, J.; Kulik, H. J.; Martinez, T. J.; Klinman, J. P., Mediation of donor–acceptor distance in an enzymatic methyl transfer reaction. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112*, 7954-7959.
11. Biler, M.; Crean, R. M.; Schweiger, A. K.; Kourist, R.; Kamerlin, S. C. L., Ground-State Destabilization by Active-Site Hydrophobicity Controls the Selectivity of a Cofactor-Free Decarboxylase. *J. Am. Chem. Soc.* **2020**, *142*, 20216-20231.
12. Zaragoza, J. P. T.; Offenbacher, A. R.; Hu, S.; Gee, C. L.; Firestein, Z. M.; Minnetian, N.; Deng, Z.; Fan, F.; Iavarone, A. T.; Klinman, J. P., Temporal and spatial resolution of distal protein motions that activate hydrogen tunneling in soybean lipoxygenase. *Proc. Natl. Acad. Sci. U.S.A.* **2023**, *120*, e2211630120.
13. Osuna, S., The challenge of predicting distal active site mutations in computational enzyme design. *WIREs Computational Molecular Science* **2021**, *11*, e1502.
14. Corbella, M.; Pinto, G. P.; Kamerlin, S. C. L., Loop dynamics and the evolution of enzyme activity. *Nature Reviews Chemistry* **2023**, *7*, 536-547.
15. Riziotis, I. G.; Ribeiro, A. J. M.; Borkakoti, N.; Thornton, J. M., Conformational Variation in Enzyme Catalysis: A Structural Study on Catalytic Residues. *J. Mol. Biol.* **2022**, *434*, 167517.
16. Schenkmayerova, A.; Pinto, G. P.; Toul, M.; Marek, M.; Hernychova, L.; Planas-Iglesias, J.; Daniel Liskova, V.; Pluskal, D.; Vasina, M.; Emond, S.; Dörr, M.; Chaloupkova, R.; Bednar, D.; Prokop, Z.; Hollfelder, F.; Bornscheuer, U. T.; Damborsky, J., Engineering the protein dynamics of an ancestral luciferase. *Nat. Commun.* **2021**, *12*, 3616.
17. Yabukarski, F.; Biel, J. T.; Pinney, M. M.; Doukov, T.; Powers, A. S.; Fraser, J. S.; Herschlag, D., Assessment of enzyme active site positioning and tests of catalytic mechanisms through X-ray–derived conformational ensembles. *Proc. Natl. Acad. Sci. U.S.A.* **2020**, *117*, 33204-33215.
18. Wankowicz, S.; Fraser, J., Making sense of chaos: uncovering the mechanisms of conformational entropy. *Chemrxiv* **2023**, 10.26434/chemrxiv-2023-9b5k7.
19. Du, S.; Kretsch, R. C.; Parres-Gold, J.; Pieri, E.; Cruzeiro, V. W. D.; Zhu, M.; Pinney, M. M.; Yabukarski, F.; Schwans, J. P.; Martínez, T. J.; Herschlag, D., Conformational ensembles reveal the origins of serine protease catalysis. *Science* **2025**, *387*.
20. Fried, S. D.; Boxer, S. G., Electric Fields and Enzyme Catalysis. *Annu. Rev. Biochem* **2017**, *86*, 387-415.

21. Fried, S. D.; Bagchi, S.; Boxer, S. G., Extreme electric fields power catalysis in the active site of ketosteroid isomerase. *Science* **2014**, *346*, 1510-1514.
22. Fried, S. D.; Boxer, S. G., Measuring Electric Fields and Noncovalent Interactions Using the Vibrational Stark Effect. *Acc. Chem. Res.* **2015**, *48*, 998-1006.
23. Bhowmick, A.; Sharma, S. C.; Head-Gordon, T., The Importance of the Scaffold for *de Novo* Enzymes: A Case Study with Kemp Eliminase. *J. Am. Chem. Soc.* **2017**, *139*, 5793-5800.
24. Vaissier, V.; Sharma, S. C.; Schaettle, K.; Zhang, T.; Head-Gordon, T., Computational Optimization of Electric Fields for Improving Catalysis of a Designed Kemp Eliminase. *ACS Catal.* **2018**, *8*, 219-227.
25. Galmés, M. À.; Nödling, A. R.; He, K.; Luk, L. Y. P.; Świderek, K.; Moliner, V., Computational design of an amidase by combining the best electrostatic features of two promiscuous hydrolases. *Chem. Sci.* **2022**, *13*, 4779-4787.
26. Austin, H. P.; Allen, M. D.; Donohoe, B. S.; Rorrer, N. A.; Kearns, F. L.; Silveira, R. L.; Pollard, B. C.; Dominick, G.; Duman, R.; El Omari, K.; Mykhaylyk, V.; Wagner, A.; Michener, W. E.; Amore, A.; Skaf, M. S.; Crowley, M. F.; Thorne, A. W.; Johnson, C. W.; Woodcock, H. L.; McGeehan, J. E.; Beckham, G. T., Characterization and engineering of a plastic-degrading aromatic polyesterase. *Proc. Natl. Acad. Sci. U.S.A.* **2018**, *115*, E4350-E4357.
27. Tournier, V.; Topham, C. M.; Gilles, A.; David, B.; Folgoas, C.; Moya-Leclair, E.; Kamionka, E.; Desrousseaux, M. L.; Texier, H.; Gavalda, S.; Cot, M.; Guémard, E.; Dalibey, M.; Nomme, J.; Cioci, G.; Barbe, S.; Chateau, M.; André, I.; Duquesne, S.; Marty, A., An engineered PET depolymerase to break down and recycle plastic bottles. *Nature* **2020**, *580*, 216-219.
28. Song, Z.; Zhang, Q.; Wu, W.; Pu, Z.; Yu, H., Rational design of enzyme activity and enantioselectivity. *Frontiers in Bioengineering and Biotechnology* **2023**, *11*.
29. Xia, Y.; Guo, W.; Han, L.; Shen, W.; Chen, X.; Yang, H., Significant Improvement of Both Catalytic Efficiency and Stability of Fructosyltransferase from Aspergillus niger by Structure-Guided Engineering of Key Residues in the Conserved Sequence of the Catalytic Domain. *Journal of Agricultural and Food Chemistry* **2022**, *70*, 7202-7210.
30. Ding, N.; Jiang, Y.; Ge, R.; Shao, Q.; Shin, W.; Ran, X.; Yang, Z. J., Enhancing Cold Adaptation of Bidomain Amylases by High‐Throughput Computational Engineering. *Angew. Chem. Int. Ed.* **2025**, *64*.
31. Dodani, S. C.; Kiss, G.; Cahn, J. K. B.; Su, Y.; Pande, V. S.; Arnold, F. H., Discovery of a regioselectivity switch in nitrating P450s guided by molecular dynamics simulations and Markov models. *Nat. Chem.* **2016**, *8*, 419-425.
32. Wang, Y.; Feng, J.; Dong, W.; Chen, X.; Yao, P.; Wu, Q.; Zhu, D., Improving Catalytic Activity and Reversing Enantio‐Specificity of ω‐Transaminase by Semi‐Rational Engineering en Route to Chiral Bulky β‐Amino Esters. *ChemCatChem* **2021**, *13*, 3396-3400.
33. Musa, M. M.; Lott, N.; Laivenieks, M.; Watanabe, L.; Vieille, C.; Phillips, R. S., A Single Point Mutation Reverses the Enantiopreference of Thermoanaerobacter ethanolicus Secondary Alcohol Dehydrogenase. *ChemCatChem* **2009**, *1*, 89-93.
34. Jeon, H.; Pagar, A. D.; Kang, H.; Giri, P.; Nadarajan, S. P.; Sarak, S.; Khobragade, T. P.; Lim, S.; Patil, M. D.; Lee, S.-G.; Yun, H., Creation of a (R)-β-Transaminase by Directed Evolution of <scp>d</scp>-Amino Acid Aminotransferase. *ACS Catal.* **2022**, *12*, 13207-13214.
35. Novick, S. J.; Dellas, N.; Garcia, R.; Ching, C.; Bautista, A.; Homan, D.; Alvizo, O.; Entwistle, D.; Kleinbeck, F.; Schlama, T.; Ruch, T., Engineering an Amine Transaminase for the Efficient Production of a Chiral Sacubitril Precursor. *ACS Catal.* **2021**, *11*, 3762-3770.

36. Zhang, H.; Tian, S.; Yue, Y.; Li, M.; Tong, W.; Xu, G.; Chen, B.; Ma, M.; Li, Y.; Wang, J.-B., Semirational Design of Fluoroacetate Dehalogenase RPA1163 for Kinetic Resolution of α-Fluorocarboxylic Acids on a Gram Scale. *ACS Catal.* **2020**, *10*, 3143-3151.
37. Reetz, M. T.; Carballeira, J. D.; Peyralans, J.; Höbenreich, H.; Maichele, A.; Vogel, A., Expanding the Substrate Scope of Enzymes: Combining Mutations Obtained by CASTing. *Chem. Eur. J.* **2006**, *12*, 6031-6038.
38. Savile, C. K.; Janey, J. M.; Mundorff, E. C.; Moore, J. C.; Tam, S.; Jarvis, W. R.; Colbeck, J. C.; Krebber, A.; Fleitz, F. J.; Brands, J.; Devine, P. N.; Huisman, G. W.; Hughes, G. J., Biocatalytic Asymmetric Synthesis of Chiral Amines from Ketones Applied to Sitagliptin Manufacture. *Science* **2010**, *329*, 305-309.
39. Güclü, D.; Szekrenyi, A.; Garrabou, X.; Kickstein, M.; Junker, S.; Clapés, P.; Fessner, W.-D., Minimalist Protein Engineering of an Aldolase Provokes Unprecedented Substrate Promiscuity. *ACS Catal.* **2016**, *6*, 1848-1852.
40. Shao, Q.; Hollenbeak, A. C.; Jiang, Y.; Ran, X.; Bachmann, B. O.; Yang, Z. J., SubTuner leverages physics-based modeling to complement AI in enzyme engineering toward non-native substrates. *Chem Catalysis* **2025**, 101334.
41. Sun, C.; Lu, G.; Chen, B.; Li, G.; Wu, Y.; Brack, Y.; Yi, D.; Ao, Y.-F.; Wu, S.; Wei, R.; Sun, Y.; Zhai, G.; Bornscheuer, U. T., Direct asymmetric synthesis of β-branched aromatic α-amino acids using engineered phenylalanine ammonia lyases. *Nat. Commun.* **2024**, *15*.
42. Zhang, Z.; Long, M.; Zheng, N.; Lü, X.; Zhu, C.; Osire, T.; Xia, X., Inside Out Computational Redesign of Cavities for Improving Thermostability and Catalytic Activity of Rhizomucor Miehei Lipase. *Appl. Environ. Microbiol.* **2023**, *89*.
43. Pardo, I.; Rodríguez-Escribano, D.; Aza, P.; De Salas, F.; Martínez, A. T.; Camarero, S., A highly stable laccase obtained by swapping the second cupredoxin domain. *Scientific Reports* **2018**, *8*.
44. Voss, M.; Das, D.; Genz, M.; Kumar, A.; Kulkarni, N.; Kustosz, J.; Kumar, P.; Bornscheuer, U. T.; Höhne, M., In Silico Based Engineering Approach to Improve Transaminases for the Conversion of Bulky Substrates. *ACS Catal.* **2018**, *8*, 11524-11533.
45. Xiang, C.; Ao, Y.-F.; Höhne, M.; Bornscheuer, U. T., Shifting the pH Optima of (R)-Selective Transaminases by Protein Engineering. *International Journal of Molecular Sciences* **2022**, *23*, 15347.
46. Reetz, M. T.; Bocola, M.; Carballeira, J. D.; Zha, D.; Vogel, A., Expanding the Range of Substrate Acceptance of Enzymes: Combinatorial Active‐Site Saturation Test. *Angew. Chem.* **2005**, *117*, 4264-4268.
47. Kwon, S.; Park, H. H., Structural Consideration of the Working Mechanism of Fold Type I Transaminases From Eubacteria: Overt and Covert Movement. *Computational and Structural Biotechnology Journal* **2019**, *17*, 1031-1039.
48. Bezsudnova, E. Y.; Popov, V. O.; Boyko, K. M., Structural insight into the substrate specificity of PLP fold type IV transaminases. *Appl. Microbiol. Biotechnol.* **2020**, *104*, 2343-2357.
49. Hammes, G. G.; Benkovic, S. J.; Hammes-Schiffer, S., Flexibility, Diversity, and Cooperativity: Pillars of Enzyme Catalysis. *Biochemistry* **2011**, *50*, 10422-10430.
50. Wijma, H. J.; Floor, R. J.; Bjelic, S.; Marrink, S. J.; Baker, D.; Janssen, D. B., Enantioselective Enzymes by Computational Design and In Silico Screening. *Angew. Chem. Int. Ed.* **2015**, *54*, 3726-3730.

51. Merlicek, L. P.; Neumann, J.; Lear, A.; Degiorgi, V.; De Waal, M. M.; Cotet, T. S.; Mulholland, A. J.; Bunzel, H. A., AI.zymes: A Modular Platform for Evolutionary Enzyme Design. *Angew. Chem. Int. Ed.* **2025**, *64*.
52. Campbell, Q.; Cox, S.; Medina, J.; Watterson, B.; White, A. D. MDCrow: Automating Molecular Dynamics Workflows with Large Language Models. arXiv:2502.09565, 2025; https://ui.adsabs.harvard.edu/abs/2025arXiv250209565C (accessed February 01, 2025).
53. Chandrasekhar, A.; Barati Farimani, A. Automating MD simulations for Proteins using Large language Models: NAMD-Agent. arXiv:2507.07887, 2025; https://ui.adsabs.harvard.edu/abs/2025arXiv250707887C (accessed July 01, 2025).
54. Guilbert, S.; Masschelein, C.; Goumaz, J.; Naida, B.; Schwaller, P. DynaMate: An Autonomous Agent for Protein-Ligand Molecular Dynamics Simulations. arXiv:2512.10034, 2025; https://ui.adsabs.harvard.edu/abs/2025arXiv251210034G (accessed December 01, 2025).
55. Ghafarollahi, A.; Buehler, M. J., ProtAgents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery* **2024**, *3*, 1389-1409.
56. Shao, Q.; Jiang, Y.; Yang, Z. J., EnzyHTP Computational Directed Evolution with Adaptive Resource Allocation. *J. Chem. Inf. Model.* **2023**, *63*, 5650-5659.
57. Shao, Q.; Jiang, Y.; Yang, Z. J., EnzyHTP: A High-Throughput Computational Platform for Enzyme Modeling. *J. Chem. Inf. Model.* **2022**, *62*, 647-655.
58. OpenAI; Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; Mądry, A.; Baker-Whitcomb, A.; Beutel, A.; Borzunov, A.; Carney, A.; Chow, A.; Kirillov, A.; Nichol, A.; Paino, A.; Renzin, A.; Tachard Passos, A.; Kirillov, A.; Christakis, A.; Conneau, A.; Kamali, A.; Jabri, A.; Moyer, A.; Tam, A.; Crookes, A.; Tootoochian, A.; Tootoonchian, A.; Kumar, A.; Vallone, A.; Karpathy, A.; Braunstein, A.; Cann, A.; Codispoti, A.; Galu, A.; Kondrich, A.; Tulloch, A.; Mishchenko, A.; Baek, A.; Jiang, A.; Pelisse, A.; Woodford, A.; Gosalia, A.; Dhar, A.; Pantuliano, A.; Nayak, A.; Oliver, A.; Zoph, B.; Ghorbani, B.; Leimberger, B.; Rossen, B.; Sokolowsky, B.; Wang, B.; Zweig, B.; Hoover, B.; Samic, B.; McGrew, B.; Spero, B.; Giertler, B.; Cheng, B.; Lightcap, B.; Walkin, B.; Quinn, B.; Guarraci, B.; Hsu, B.; Kellogg, B.; Eastman, B.; Lugaresi, C.; Wainwright, C.; Bassin, C.; Hudson, C.; Chu, C.; Nelson, C.; Li, C.; Shern, C. J.; Conger, C.; Barette, C.; Voss, C.; Ding, C.; Lu, C.; Zhang, C.; Beaumont, C.; Hallacy, C.; Koch, C.; Gibson, C.; Kim, C.; Choi, C.; McLeavey, C.; Hesse, C.; Fischer, C.; Winter, C.; Czarnecki, C.; Jarvis, C.; Wei, C.; Koumouzelis, C.; Sherburn, D.; Kappler, D.; Levin, D.; Levy, D.; Carr, D.; Farhi, D.; Mely, D.; Robinson, D.; Sasaki, D.; Jin, D.; Valladares, D.; Tsipras, D.; Li, D.; Nguyen, D. P.; Findlay, D.; Oiwoh, E.; Wong, E.; Asdar, E.; Proehl, E.; Yang, E.; Antonow, E.; Kramer, E.; Peterson, E.; Sigler, E.; Wallace, E.; Brevdo, E.; Mays, E.; Khorasani, F.; Petroski Such, F.; Raso, F.; Zhang, F.; von Lohmann, F.; Sulit, F.; Goh, G.; Oden, G.; Salmon, G.; Starace, G.; Brockman, G.; Salman, H.; Bao, H.; Hu, H.; Wong, H.; Wang, H.; Schmidt, H.; Whitney, H.; Jun, H.; Kirchner, H.; Ponde de Oliveira Pinto, H.; Ren, H.; Chang, H.; Chung, H. W.; Kivlichan, I.; O'Connell, I.; O'Connell, I.; Osband, I.; Silber, I.; Sohl, I.; Okuyucu, I.; Lan, I.; Kostrikov, I.; Sutskever, I.; Kanitscheider, I.; Gulrajani, I.; Coxon, J.; Menick, J.; Pachocki, J.; Aung, J.; Betker, J.; Crooks, J.; Lennon, J.; Kiros, J.; Leike, J.; Park, J.; Kwon, J.; Phang, J.; Teplitz, J.; Wei, J.; Wolfe, J.; Chen, J.; Harris, J.; Varavva, J.; Lee, J. G.; Shieh, J.; Lin, J.; Yu, J.; Weng, J.; Tang, J.; Yu, J.; Jang, J.; Quinonero Candela, J.; Beutler, J.; Landers, J.; Parish, J.; Heidecke, J.; Schulman, J.; Lachman, J.; McKay, J.; Uesato, J.; Ward, J.; Kim, J. W. GPT-4o

System Card. arXiv:2410.21276, 2024; https://ui.adsabs.harvard.edu/abs/2024arXiv241021276O (accessed October 01, 2024).
59. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903, 2022; https://ui.adsabs.harvard.edu/abs/2022arXiv220111903W (accessed January 01, 2022).
60. Liu, H.; Sferrazza, C.; Abbeel, P. Chain of Hindsight Aligns Language Models with Feedback. arXiv:2302.02676, 2023; https://ui.adsabs.harvard.edu/abs/2023arXiv230202676L (accessed February 01, 2023).
61. Kang, J.; Laroche, R.; Yuan, X.; Trischler, A.; Liu, X.; Fu, J. Think Before You Act: Decision Transformers with Working Memory. arXiv:2305.16338, 2023; https://ui.adsabs.harvard.edu/abs/2023arXiv230516338K (accessed May 01, 2023).
62. Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; Amodei, D. Language Models are Few-Shot Learners. arXiv:2005.14165, 2020; https://ui.adsabs.harvard.edu/abs/2020arXiv200514165B (accessed May 01, 2020).
63. Qin, Y.; Zhang, T.; Zhang, T.; Shen, Y.; Luo, W.; sunhaoze; Zhang, Y.; Qiao, Y.; chen; Zhou, Z.; Zhang, W.; Cui, B. SysBench: Can LLMs Follow System Message? In The Thirteenth International Conference on Learning Representations, 2025.
64. Li, X.; Yu, Z.; Zhang, Z.; Chen, X.; Zhang, Z.; Zhuang, Y.; Sadagopan, N.; Beniwal, A. When Thinking Fails: The Pitfalls of Reasoning for Instruction-Following in LLMs. arXiv:2505.11423, 2025; https://ui.adsabs.harvard.edu/abs/2025arXiv250511423L (accessed May 01, 2025).
65. Song, Z.; Lu, J.; Du, Y.; Yu, B.; Pruyn, T. M.; Huang, Y.; Guo, K.; Luo, X.; Qu, Y.; Qu, Y.; Wang, Y.; Wang, H.; Guo, J.; Gan, J.; Shojaee, P.; Luo, D.; Bran, A. M.; Li, G.; Zhao, Q.; Lennon Luo, S.-X.; Zhang, Y.; Zou, X.; Zhao, W.; Zhang, Y. F.; Zhang, W.; Zheng, S.; Zhang, S.; Takrim Khan, S.; Rajabi-Kochi, M.; Paradi-Maropakis, S.; Baltoiu, T.; Xie, F.; Chen, T.; Huang, K.; Luo, W.; Fang, M.; Yang, X.; Cheng, L.; He, J.; Hassoun, S.; Zhang, X.; Wang, W.; Reddy, C. K.; Zhang, C.; Zheng, Z.; Wang, M.; Cong, L.; Gomes, C. P.; Hsieh, C.-Y.; Nandy, A.; Schwaller, P.; Kulik, H. J.; Jia, H.; Sun, H.; Mohamad Moosavi, S.; Duan, C. Evaluating Large Language Models in Scientific Discovery. arXiv:2512.15567, 2025; https://ui.adsabs.harvard.edu/abs/2025arXiv251215567S (accessed December 01, 2025).
66. Gan, Y.; Li, C.; Xie, J.; Wen, L.; Purver, M.; Poesio, M., ClarQ4LLM: A Benchmark for Models Clarifying and Requesting Information in Task-Oriented Dialog. *IEEE Transactions on Audio, Speech and Language Processing* **2026**, *34*, 1441-1456.
67. Sansen, S.; Hsu, M.-H.; Stout, C. D.; Johnson, E. F., Structural insight into the altered substrate specificity of human cytochrome P450 2A6 mutants. *Arch. Biochem. Biophys.* **2007**, *464*, 197-206.
68. Ma, B.; Niu, J.; Zhu, H.; Chi, H.; Lu, Z.; Lu, F.; Zhu, P., Engineering substrate specificity of quinone-dependent dehydrogenases for efficient oxidation of deoxynivalenol to 3-keto-deoxynivalenol. *Int. J. Biol. Macromol.* **2024**, *264*, 130484.
69. Schülke, K. H.; Fröse, J. S.; Klein, A.; Garcia‐Borràs, M.; Hammer, S. C., Efficient Transferase Engineering for SAM Analog Synthesis from Iodoalkanes. *ChemBioChem* **2024**, *25*.

70. Tang, Q.; Grathwol, C. W.; Aslan‐Üzel, A. S.; Wu, S.; Link, A.; Pavlidis, I. V.; Badenhorst, C. P. S.; Bornscheuer, U. T., Directed Evolution of a Halide Methyltransferase Enables Biocatalytic Synthesis of Diverse SAM Analogs. *Angew. Chem. Int. Ed.* **2021**, *60*, 1524-1527.
71. Al-Ghanayem, A. A.; Joseph, B., Current prospective in using cold-active enzymes as eco-friendly detergent additive. *Appl. Microbiol. Biotechnol.* **2020**, *104*, 2871-2882.
72. Chen, Q.; Wu, Y.; Huang, Z.; Zhang, W.; Chen, J.; Mu, W., Cold-active enzymes in the dairy industry: Insight into cold adaption mechanisms and their applications. *Trends in Food Science & Technology* **2022**, *125*, 126-135.
73. Ding, N.; Zhao, B.; Ban, X.; Li, C.; Venkataram Prasad, B. V.; Gu, Z.; Li, Z., Carbohydrate-Binding Module and Linker Allow Cold Adaptation and Salt Tolerance of Maltopentaose-Forming Amylase From Marine Bacterium Saccharophagus degradans 2-40T. *Frontiers in Microbiology* **2021**, *12*.
74. Liu, Y.; Jia, K.; Chen, H.; Wang, Z.; Zhao, W.; Zhu, L., Cold-adapted enzymes: mechanisms, engineering and biotechnological application. *Bioprocess Biosystems Eng.* **2023**, *46*, 1399-1410.
75. Meyer Cifuentes, I. E.; Wu, P.; Zhao, Y.; Liu, W.; Neumann-Schaal, M.; Pfaff, L.; Barys, J.; Li, Z.; Gao, J.; Han, X.; Bornscheuer, U. T.; Wei, R.; Öztürk, B., Molecular and Biochemical Differences of the Tandem and Cold-Adapted PET Hydrolases Ple628 and Ple629, Isolated From a Marine Microbial Consortium. *Frontiers in Bioengineering and Biotechnology* **2022**, *10*.
76. Adam, M.; Potter, A. S.; Potter, S. S., Psychrophilic proteases dramatically reduce single cell RNA-seq artifacts: A molecular atlas of kidney development. *Development* **2017**, *144*, 3625-3632.
77. Ge, R.; Ding, N.; Jiang, Y.; Yang, Z. J., Linker‐mediated domain separation enhances cold adaptation in cellulases. *Protein Sci.* **2025**, *34*.
78. George, R. A.; Heringa, J., An analysis of protein domain linkers: their classification and role in protein folding. *Protein Engineering, Design and Selection* **2002**, *15*, 871-879.
79. Liu, C.; Chin, J. X.; Lee, D.-Y., SynLinker: an integrated system for designing linkers and synthetic fusion proteins. *Bioinformatics* **2015**, *31*, 3700-3702.
80. Miller, G. L., Use of Dinitrosalicylic Acid Reagent for Determination of Reducing Sugar. *Anal. Chem.* **1959**, *31*, 426-428.
81. Wang, E.; Sun, H.; Wang, J.; Wang, Z.; Liu, H.; Zhang, J. Z. H.; Hou, T., End-Point Binding Free Energy Calculation with MM/PBSA and MM/GBSA: Strategies and Applications in Drug Design. *Chem. Rev.* **2019**, *119*, 9478-9508.
82. Jiang, Y.; Ding, N.; Shao, Q.; Stull, S. L.; Cheng, Z.; Yang, Z. J., Substrate Positioning Dynamics Involves a Non-Electrostatic Component to Mediate Catalysis. *The Journal of Physical Chemistry Letters* **2023**, *14*, 11480-11489.
83. D.A. Case, H. M. A., K. Belfon, I.Y. Ben-Shalom, S.R. Brozell, D.S. Cerutti, T.E. Cheatham, III, G.A. Cisneros, V.W.D. Cruzeiro, T.A. Darden, R.E. Duke, G. Giambasu, M.K. Gilson, H. Gohlke, A.W. Goetz, R. Harris, S. Izadi, S.A. Izmailov, C. Jin, K. Kasavajhala, M.C. Kaymak, E. King, A. Kovalenko, T. Kurtzman, T.S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, M. Machado, V. Man, M. Manathunga, K.M. Merz, Y. Miao, O. Mikhailovskii, G. Monard, H. Nguyen, K.A. O'Hearn, A. Onufriev, F. Pan, S. Pantano, R. Qi, A. Rahnamoun, D.R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C.L. Simmerling, N.R. Skrynnikov, J. Smith, J. Swails, R.C. Walker, J. Wang, H. Wei, R.M. Wolf, X. Wu, Y. Xue, D.M. York, S. Zhao, and P.A. Kollman Amber 2021.

84. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A. V.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ortiz, J. V.; Izmaylov, A. F.; Sonnenberg, J. L.; Williams; Ding, F.; Lipparini, F.; Egidi, F.; Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; Gao, J.; Rega, N.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; Montgomery Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M. J.; Heyd, J. J.; Brothers, E. N.; Kudin, K. N.; Staroverov, V. N.; Keith, T. A.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A. P.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. *Gaussian 16 Rev. C.01*, Wallingford, CT, **2016**.
85. Bannwarth, C.; Ehlert, S.; Grimme, S., GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions. *J. Chem. Theory Comput.* **2019**, *15*, 1652-1671.
86. Lu, T.; Chen, F., Multiwfn: A multifunctional wavefunction analyzer. *J. Comput. Chem.* **2012**, *33*, 580-592.
87. Leaver-Fay, A.; Tyka, M.; Lewis, S. M.; Lange, O. F.; Thompson, J.; Jacak, R.; Kaufman, K. W.; Renfrew, P. D.; Smith, C. A.; Sheffler, W.; Davis, I. W.; Cooper, S.; Treuille, A.; Mandell, D. J.; Richter, F.; Ban, Y.-E. A.; Fleishman, S. J.; Corn, J. E.; Kim, D. E.; Lyskov, S.; Berrondo, M.; Mentzer, S.; Popović, Z.; Havranek, J. J.; Karanicolas, J.; Das, R.; Meiler, J.; Kortemme, T.; Gray, J. J.; Kuhlman, B.; Baker, D.; Bradley, P. Rosetta3. In; Elsevier: 2011, pp 545-574.
88. Gao, S.; Fang, A.; Huang, Y.; Giunchiglia, V.; Noori, A.; Schwarz, J. R.; Ektefaie, Y.; Kondic, J.; Zitnik, M., Empowering biomedical discovery with AI agents. *Cell* **2024**, *187*, 6125-6151.
89. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D., Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583-589.
90. Mirdita, M.; Schütze, K.; Moriwaki, Y.; Heo, L.; Ovchinnikov, S.; Steinegger, M., ColabFold: making protein folding accessible to all. *Nat. Methods* **2022**, *19*, 679-682.
91. Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C., ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.* **2015**, *11*, 3696-3713.
92. Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L., Comparison of simple potential functions for simulating liquid water. *J. Chem. Phy.* **1983**, *79*, 926-935.
93. Wang, N. X.; Wilson, A. K., The behavior of density functionals with respect to basis set. I. The correlation consistent basis sets. *J. Chem. Phys.* **2004**, *121*, 7632-46.
94. Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A., Automatic atom type and bond type perception in molecular mechanical calculations. *J. Mol. Graphics Modell.* **2006**, *25*, 247-60.

95. Jakalian, A.; Jack, D. B.; Bayly, C. I., Fast, efficient generation of high-quality atomic charges. AM1-BCC model: II. Parameterization and validation. *J. Comput. Chem.* **2002**, *23*, 1623-41.
96. Case, D. A.; Ben-Shalom, I. Y.; Brozell, S. R.; Cerutti, D. S.; Cheatham, T. E., III; Cruzeiro, V. W. D.; Darden, T. A.; Duke, R. E.; Ghoreishi, D.; Gilson, M. K.; Gohlke, H.; Goetz, A. W.; Greene, D.; Harris, R.; Homeyer, N.; Huang, Y.; Izadi, S.; Kovalenko, A.; Kurtzman, T.; Lee, T. S.; LeGrand, S.; Li, P.; Lin, C.; Liu, J.; Luchko, T.; Luo, R.; Mermelstein, D. J.; Merz, K. M.; Miao, Y.; Monard, G.; Nguyen, C.; Nguyen, H.; Omelyan, I.; Onufriev, A.; Pan, F.; Qi, R.; Roe, D. R.; Roitberg, A.; Sagui, C.; Schott-Verdugo, S.; Shen, J.; Simmerling, C. L.; Smith, J.; SalomonFerrer, R.; Swails, J.; Walker, R. C.; Wang, J.; Wei, H.; Wolf, R. M.; Wu, X.; Xiao, L.; York, D. M.; Kollman, P. A. *AMBER 2018*, University of California, San Francisco, **2018**.
97. Chodera, J. D., A Simple Method for Automated Equilibration Detection in Molecular Simulations. *J. Chem. Theory Comput.* **2016**, *12*, 1799-1805.
98. Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. C., Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-Alkanes. *J. Comput. Phys.* **1977**, *23*, 327-341.
99. Loncharich, R. J.; Brooks, B. R.; Pastor, R. W., Langevin dynamics of peptides: the frictional dependence of isomerization rates of N-acetylalanyl-N'-methylamide. *Biopolymers* **1992**, *32*, 523-35.
100. Berendsen, H. J. C.; Postma, J. P. M.; Gunsteren, W. F. v.; DiNola, A.; Haak, J. R., Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, *81*, 3684-3690.
101. Sullivan, C.; Kaszynski, A., PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK). *Journal of Open Source Software* **2019**, *4*, 1450.
102. Sehnal, D.; Svobodová Vařeková, R.; Berka, K.; Pravda, L.; Navrátilová, V.; Banáš, P.; Ionescu, C.-M.; Otyepka, M.; Koča, J., MOLE 2.0: advanced approach for analysis of biomacromolecular channels. *J. Cheminf.* **2013**, *5*, 39.
103. Towns, J.; Cockerill, T.; Dahan, M.; Foster, I.; Gaither, K.; Grimshaw, A.; Hazlewood, V.; Lathrop, S.; Lifka, D.; Peterson, G. D.; Roskies, R.; Scott, J. R.; Wilkins-Diehr, N., XSEDE: Accelerating Scientific Discovery. *Computing in Science & Engineering* **2014**, *16*, 62-74.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SI.pdf