VIBEPROTEINBENCH: AN EVALUATION BENCHMARK FOR LANGUAGE-INTERFACED VIBE PROTEIN DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein design aims to compose amino-acid sequences that fold into stable three-dimensional structures while satisfying targeted functional properties. The field is increasingly shifting toward *vibe protein design*, where a single model is expected to generate novel sequences, engineer existing proteins, and reason about protein characteristics through flexible natural-language constraints. However, existing benchmarks typically evaluate isolated aspects of protein design or assume predefined structured inputs, making them ill-suited to assess this broad, open-ended setting. To address this gap, we present Vibe Protein Design Benchmark (VIBEPROTEINBENCH), a language-interfaced benchmark that evaluates whether a single model can operate across three complementary stages of a computational protein-design workflow: ***recognition***, ***engineering***, and ***generation***. Each stage is grounded in expert-curated mechanistic rationales and multi-faceted in silico validation, to computationally verify whether model outputs are biologically plausible. Across diverse general-purpose and domain-specialized LLMs, evaluated with and without tool access, no model performs strongly across all three stages at once. Beyond measuring end-task success, VIBEPROTEINBENCH exposes where current systems break: we identify recurring failure patterns and characterize how tool-augmented agents select tool arguments. Together, these findings show that vibe protein design remains a substantial open challenge.

1 INTRODUCTION

Protein design aims to compose amino-acid sequences that fold into stable three-dimensional structures while satisfying targeted functional properties. Owing to its broad applicability across therapeutic development, agriculture, and biotechnology, it has attracted significant attention as a promising direction for computational modeling and scientific discovery (Kyro et al., 2025; Cotet et al., 2025; Notin et al., 2024; Gruver et al., 2023). This interest has driven rapid progress in deep learning approaches to computational protein design, advancing tasks such as protein engineering (Hie et al., 2024; Hie & Yang, 2022; Dieckhaus et al., 2024) and de novo sequence generation (Stark et al., 2025; Didi et al., 2026; Pacesa et al., 2024). However, these approaches have largely been developed as task-specific models that consume *predefined, structured* inputs such as backbone coordinates, numerical descriptors, or other functional constraints (Dauparas et al., 2022; Hayes et al., 2025; Nijkamp et al., 2023; Kucera et al., 2022; Munsamy et al., 2022). Such formulations are effective in controlled settings, but offer limited flexibility for open-ended objectives, where practitioners may express more naturally at a high level such as designing an enzyme capable of degrading PFAS.

Recent work has begun to explore models that take flexible biological objectives expressed in natural language, infer the corresponding task requirements from relevant protein context, and carry out the appropriate protein-design task (Liu et al., 2025c; Ponnapati et al., 2025; Sinclair et al., 2025; Wang et al., 2025c). We refer to this setting as *vibe protein design*, drawing an analogy to *vibe coding* in software development (Gu et al., 2026; Li et al., 2026; Meske et al., 2025).

These objectives, however, are rarely satisfied by a single task. *Generating* a protein entails *recognizing* its biophysical and functional properties to validate its quality (He et al., 2026; Qing et al., 2022), and the resulting sequence often requires further *engineering* to enhance a targeted property (Sappington et al., 2026; Cho et al., 2025). Large language models (LLMs) have increasingly been investigated in this role, as their language interface admits biological intents of diverse tasks

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

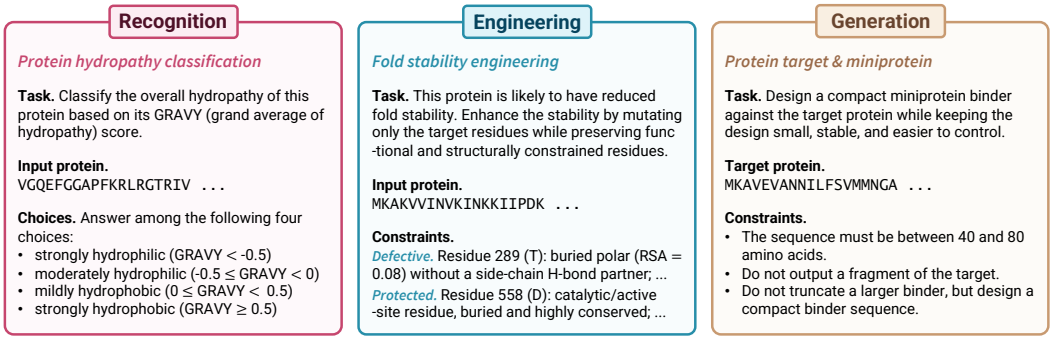

Figure 1: Representative queries from the three stages of VIBEPROTEINBENCH. Each query supplies a natural-language task, the protein or target context, and the stage-specific constraints.

and their reasoning enables orchestrating the tools each demands (OpenAI, 2025; 2026b; Lv et al., 2025; Zhuo et al., 2024; Xiao et al., 2024; Wang et al., 2025b; Zhou et al., 2025; Ma et al., 2025). Nevertheless, whether one model can perform reliably across all three capabilities, with or without access to external tools, remains unclear.

As this direction gains momentum, rigorously evaluating these models becomes increasingly essential. A benchmark for vibe protein design should therefore assess whether a single model can operate across this connected workflow under natural-language design intents. Existing benchmarks, however, only partially satisfy this requirement, either focusing on isolated aspects of protein design (Kuang et al., 2025; Rong et al., 2025) or relying on predefined structured inputs (Wang et al., 2023; Ye et al., 2024), and therefore fall short of an integrated framework for language-interfaced settings.

To address this gap, we present Vibe Protein Design Benchmark (VIBEPROTEINBENCH), a language-interfaced benchmark that operationalizes this workflow as three evaluation stages: ***recognition***, ***engineering***, and ***generation***. VIBEPROTEINBENCH instantiates each stage through natural-language queries paired with the protein context needed to express realistic design intents. Recognition uses question-answering over sequence, structural, and functional properties; engineering asks models to edit existing proteins using expert-curated mechanistic rationales; and generation evaluates novel protein design from functional or target-grounded specifications. Every stage carries a computationally verifiable evaluation protocol built on strong evaluators (Team et al., 2026b; Cho et al., 2026; Passaro et al., 2025), from exact-answer assessment to multi-faceted in silico validation. We validate queries and evaluators through human expert review and in silico evaluators comparison (Appendix B).

We evaluate general-purpose LLMs (OpenAI, 2026a; Anthropic, 2026; DeepMind, 2026; Liu et al., 2025a; Deepseek-AI, 2026; Qwen, 2026; Team et al., 2026a) on VIBEPROTEINBENCH in both tool-free and tool-augmented agent settings, and additionally evaluate domain-specialized LLMs (Xia et al., 2025; Wang et al., 2025c;a) to assess whether scientific pretraining confers an advantage. Our results reveal that *no single model performs strongly across all three stages at once*, with pass rates dropping sharply from recognition to engineering and generation. Beyond measuring end-task success, VIBEPROTEINBENCH exposes where current systems break: we identify recurring failure patterns and characterize how tool-augmented agents select and misuse tool arguments. We further find that scores are correlated across biologically linked subtasks spanning different stages, suggesting that VIBEPROTEINBENCH captures coherent protein-design competence rather than a collection of isolated probes. Together, these findings show that vibe protein design remains a substantial open challenge while validating the integrated structure of the benchmark.

2 RELATED WORK

Non-language-based protein design benchmarks. Early protein design benchmarks primarily evaluate models under structured input schemas, such as backbone coordinates, geometric constraints, motif residues, or predefined target properties. These benchmarks have been central to measuring core design capabilities such as inverse folding, foldability, structural recovery, and mutation-effect prediction (Wang et al., 2023; Gao et al., 2023; Ye et al., 2024; Tan et al., 2025). However, because their inputs are not expressed in language, they are limited in evaluating open-ended biological intents that practitioners may describe flexibly in text.

Language-interfaced protein design benchmarks. Recent benchmarks have begun to evaluate protein design through natural-language interfaces. One line of work evaluates sequence generation conditioned on a natural-language function description or a bound ligand (Kuang et al., 2025; Song et al., 2025; Fang et al., 2023), but limits its evaluation to a subset of generation tasks. A parallel line evaluates LLM agents that orchestrate external protein-design tools, scoring how they plan a design workflow, execute the appropriate software, and interpret the resulting validation outputs (Kim & Romero, 2026; Cai et al., 2025; Tan et al., 2026), yet reports performance only in the tool-augmented setting, leaving the internal design competence of LLMs unmeasured. In contrast, VIBEPROTEINBENCH queries a broad spectrum of the protein-design workflow, spanning recognition, engineering, and generation, and evaluates LLMs under both tool-free and tool-augmented settings. Its generation stage covers both semantic functional specifications and explicit binding targets. We provide a detailed comparison with more related literature in Appendix F.

3 TASK FORMULATION

We formulate vibe protein design as a setting in which a practitioner states an open-ended biological objective in natural language with the protein or target context, and the model must resolve it into a concrete, task-specific response. Conventional formulations instead assume a predefined input schema such as a numerical target or a geometric constraint (Wang et al., 2023; Ye et al., 2024; Xu et al., 2022; Yang et al., 2024b), which fixes both the form and the granularity of what can be stated. A language interface fixes neither, carrying a coarse functional goal and a mechanistic rationale in the prompt: a practitioner may ask whether an uncharacterized sequence shares a homolog's fold despite low sequence identity, how to improve the thermostability of an enzyme while preserving catalytic activity, or how to design an enzyme for a reaction with no natural counterpart.

Notably, these differing outputs correspond to distinct but interdependent design capabilities, spanning a broad workflow of understanding, refinement, and creation. Verifying a generated protein requires recognizing its biophysical and functional characteristics (He et al., 2026; Qing et al., 2022), and the protein itself frequently has to be engineered to satisfy the intended properties (Sappington et al., 2026; Cho et al., 2025). Such engineering decisions, in turn, depend on identifying which sequence or structural features to change and which to preserve (Goldenzweig et al., 2016), a recognition problem in its own right. We accordingly organize VIBEPROTEINBENCH around three complementary stages that together cover this workflow: protein recognition, rationale-guided engineering, and functional protein generation.

Protein recognition evaluates whether a model can infer biologically meaningful properties from a protein sequence. This stage probes understanding across the sequence–structure–function paradigm, which grounds downstream protein design decisions (Sandhya et al., 2016; Hamamsy et al., 2023).

Rationale-guided engineering evaluates whether a model can internalize mechanistic rationales and translate them into sequence modifications that enhance a target property. In practice, human experts identify residues responsible for a design deficiency (Chubb et al., 2026) while preserving residues critical to fold and function (Teufl et al., 2022). We capture this workflow by instructing the model to repair defective target positions and keep protected positions intact.

Functional protein generation evaluates whether a model can generate novel protein sequences from functional design intent. This stage covers diverse generation scenarios, ranging from semantic biological functions that are difficult to express as a rigid numerical or geometric constraint (Wang et al., 2025c; Yang et al., 2024a; Dai et al., 2024) to target-grounded binder design where the specific binding target serves as the design constraint (Stark et al., 2025; Didi et al., 2026; Pacesa et al., 2024).

4 BENCHMARK DESIGN AND CONSTRUCTION

In this section, we describe the construction of VIBEPROTEINBENCH across its three primary stages: protein recognition (Section 4.1), rationale-guided protein engineering (Section 4.2), and functional protein generation (Section 4.3). Our construction strategy is designed to make each stage cover representative and practically relevant scenarios that arise in language-interfaced protein design. To this end, we first consult domain experts to identify stage-specific subtasks that reflect practical expert intents and core design capabilities. We then curate the proteins, functional terms, or molecular targets needed to instantiate them from public repositories under a strict temporal cutoff that reduces

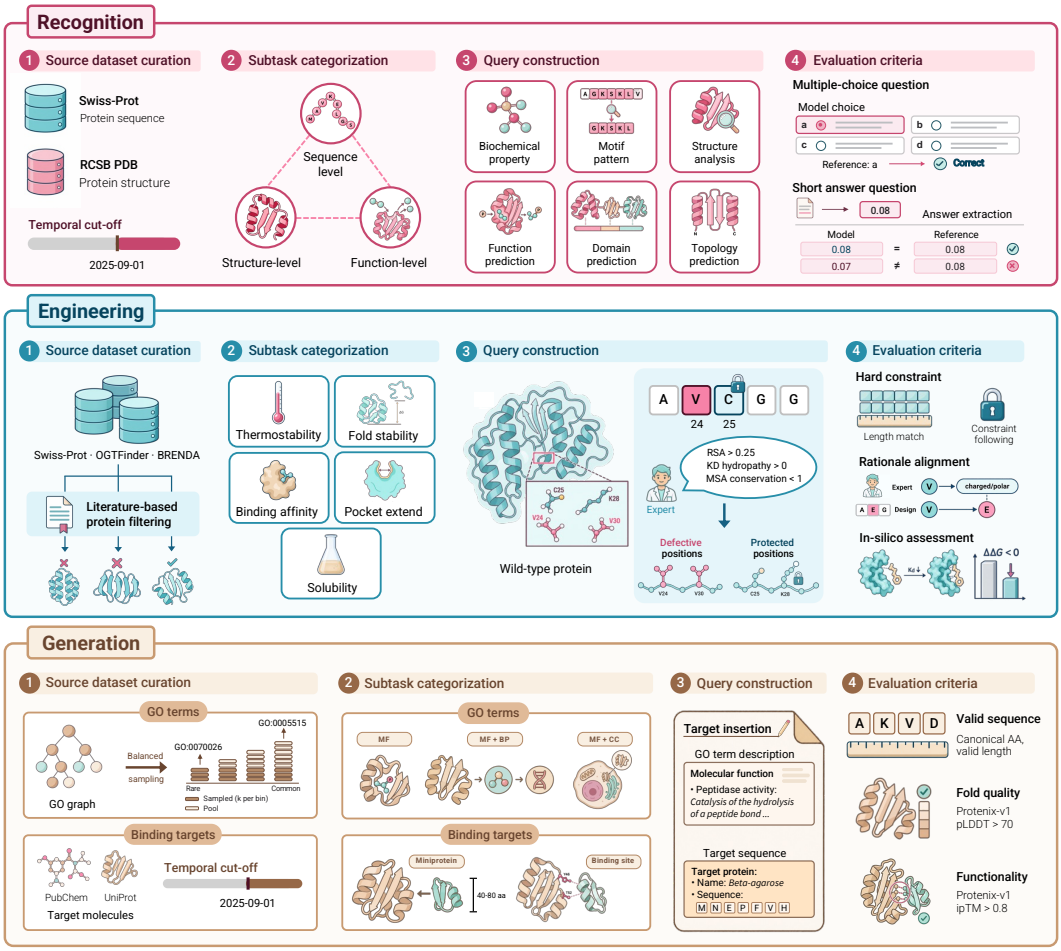

Figure 2: Evaluation dataset construction pipeline of VIBEPROTEINBENCH. VIBEPROTEINBENCH consists of three stages mirroring a computational protein design workflow: recognition (top), engineering (middle), and generation (bottom). Each stage is built through a shared four-step process: **source dataset curation** collects proteins, functional terms, or molecular targets from public repositories with stage-specific contamination control, **subtask categorization** defines target design intents, **query construction** illustrates constraints included in queries, and **evaluation criteria** assess model outputs through exact match or multi-faceted in silico evaluators.

contamination from the training corpora of contemporary LLMs. Finally, we establish stage-specific evaluation rubrics that combine hard constraints with multi-faceted in silico validation. Figure 1 illustrates representative queries and Figure 2 summarizes the construction pipeline.

4.1 PROTEIN RECOGNITION

Subtask categorization. The recognition stage is grounded in the canonical sequence–structure–function paradigm (Sandhya et al., 2016; Koehler Leman et al., 2023), which links protein sequences to three-dimensional structures and biological functions. We therefore organize subtasks into three corresponding levels. The sequence-level probes local and global physicochemical properties such as motif detection and net charge prediction; the structure-level evaluates secondary structure, residue burial, and disulfide bonding; and the function-level assesses evolutionary families and functional roles. These layers identify which positions are modifiable liabilities, which must be preserved, and which functional sites can be carried over from related families in downstream protein design.

Source dataset curation. We curate proteins from the UniProtKB/Swiss-Prot database (The UniProt Consortium, 2023) whose experimentally resolved 3D structures are deposited in the RCSB PDB

database (Burley et al., 2021). To minimize contamination risk from the training corpora of contemporary LLMs, we restrict the dataset to UniProt entries registered after September 1, 2025.

Query construction. Each query pairs a protein sequence with a natural-language question that names the property to be identified, phrased as a practitioner would pose it during analysis. We derive the reference property values from computational tools and curated databases, including Biopython descriptors (Cock et al., 2009), DSSP assignments (Kabsch & Sander, 1983) on experimentally resolved structures, and domain and function annotations from ECOD (Cheng et al., 2014), InterPro (Blum et al., 2021), and Gene Ontology (GO) (Gligorijević et al., 2021). Although these labels derive from experimental structures and curated annotations, each query exposes only the amino-acid sequence, so the model must recover the property from its own knowledge or by selecting an appropriate tool to analyze or fold the sequence. We pose categorical properties as multiple-choice questions, and elicit deterministic structured outputs such as residue-level attributes or functional position identifiers through short-answer generation.

Evaluation. We evaluate each query according to its format. For multiple-choice, a response is correct when the model selects the ground-truth option. For short-answer generation, we use an exact match against the deterministic reference answer after output normalization.

4.2 Rationale-guided engineering

Subtask categorization. The engineering stage is designed to reflect property-driven protein optimization, where practitioners aim to improve a specific biochemical property while preserving structural and functional integrity. We organize subtasks around three objectives that span the primary dimensions along which proteins are routinely engineered in practice (Korendovych, 2017; Grigorakis et al., 2025): solubility, stability (including fold stability and thermostability), and activity (including binding affinity and pocket expansion).

Source dataset curation. We draw wild-type proteins from the UniProt/Swiss-Prot database for solubility and fold stability, the OGTFinder database (Colette et al., 2026) for thermostability, and the BRENDA enzyme database (Schomburg et al., 2004) for activity. To reduce contamination from previously engineered targets, we apply a rule-based literature filter before constructing editing queries. We query PubMed and Semantic Scholar for each wild type using task-specific engineering search terms, and discard it when a returned title or abstract metadata simultaneously contains the protein name, a protein-design keyword (*e.g.*, rational protein design), and at least one task-context keyword (*e.g.*, stability engineering). We detail the full search terms in Appendix D.

Expert-curated design rationales. Protein engineering in practice is guided by mechanistic rationales that identify which residues carry a specific deficiency and in which direction to repair them, while leaving intact the residues critical to fold and function. To capture this process, we formalize each engineering task as a rationale-guided mutation design problem, in which the model must mutate a designated set of defective positions while leaving protected positions unchanged. Since each engineering objective recurs with similar mechanistic logic, we elicit rationale templates per subtask from domain experts. Each template names the liability to be repaired, the admissible mutation directions for the target property, the residue classes that must be preserved, and the concrete criteria by which each of these is recognized. We then instantiate each template on every wild-type protein by computing the features its criteria refer to, combining structural and biochemical analysis with Henikoff-weighted MSA conservation statistics (Henikoff & Henikoff, 1994). Applying the criteria to these features yields the disjoint defective and protected sets. The complete set of rationale templates and criteria is provided in Appendix D.

Query construction. Each query presents a wild-type sequence, a target property to optimize, and the rationale-derived defective and protected residues. The model is tasked with proposing a mutant that improves the property by engineering only the defective positions while leaving protected positions unchanged. This formulation evaluates whether the model can *internalize and apply mechanistic design rationales*, rather than relying on unconstrained sequence generation. To ensure that the provided rationales are actionable, we validate each query by substituting defective positions with admissible reference residues and retaining only those cases where the substitutions yield measurable improvements under our in silico evaluation pipeline. We additionally construct a constraint-free variant of each query that omits the defective and protected sets and supplies only the wild-type and the target property, requiring the model to localize the liability and choose the repair on its own.

Evaluation. We evaluate each proposed mutant using three criteria. Hard constraints (Hard const.) verify whether the mutant preserves the wild-type length, keeps protected positions unchanged, performs at least one mutation, and mutates only defective targets. Rationale alignment (Rat align.) checks whether each mutation follows the expert-curated repair direction for the target property. In silico validation (In silico valid.) assesses biological plausibility using confidence from co-folding models (Team et al., 2026b; Passaro et al., 2025) and property-specific validators (Cho et al., 2026; Passaro et al., 2025; Guerra et al., 2021). We omit rationale alignment for constraint-free queries, since a repair direction cannot be assessed automatically once a model mutates arbitrary positions. We provide the full rationale templates, literature search terms, and evaluator thresholds in Appendix D.

4.3 Functional protein generation

Subtask categorization. The generation stage covers two ways practitioners express open-ended design intent: semantic functional descriptions and explicit molecular targets. The first track addresses functional roles that are difficult to formalize as structured constraints, making natural language an ideal way to convey them (Kuang et al., 2025; Fang et al., 2023; Lv et al., 2025; Yang et al., 2024a; Dai et al., 2024). We draw these specifications from language-based functional definitions of Gene Ontology (GO) terms, ranging from a single molecular function (MF) to MF paired with biological process (BP) or cellular component (CC). The resulting terms span broad protein classes from enzymatic catalysis and nucleic-acid processing to transcription factor, transporter, and motor activity. The second track focuses on binder design for specific molecular targets, where the target representation itself (*e.g.*, SMILES or protein sequence) serves as the design specification (Ponnapati et al., 2025; Ye et al., 2024; Song et al., 2025; Stark et al., 2025; Pacesa et al., 2024). We further divide this track into small-molecule and protein-target settings, and the protein-target task additionally carries two variants drawn from practical laboratory demands: a miniprotein length constraint for controllability (Baker et al., 2017; Ożga & Berlicki, 2022), and a binding-site constraint naming the residues the binder must contact, reflecting therapeutic applications that depend on engaging a particular receptor surface (Bennett et al., 2026). Beyond these tasks, we report a supplementary set of antibody, DNA, and RNA binder design in Table 9.

Source dataset curation. For GO-conditioned generation, we sample single terms and co-occurring term pairs across information-content (IC) bins (Gligorijević et al., 2021), using marginal and joint annotation frequencies, respectively, so that queries are not dominated by overly common annotations (Appendix E). For binder generation, we apply a strict temporal holdout (2025-09-01) when curating novel protein targets from UniProt and small-molecule targets from PubChem.

Query construction. Each query provides either a natural-language functional specification or a molecular target representation, and asks the model to generate a novel protein sequence conditioned on that objective. The miniprotein subtask additionally requires a sequence of 40 to 80 residues, and the binding-site subtask enumerates the target residues the binder must contact, each written as a one-letter amino acid followed by its one-indexed position in the target sequence (*e.g.*, Y46).

Evaluation. We evaluate generated sequences along three axes. Sequence validity (Seq. valid.) checks canonical amino-acid composition and, where applicable, the miniprotein length constraint. Fold plausibility (Fold valid.) is assessed from Protenix-v1 (Team et al., 2026b) confidence. Functional consistency (Func. Cons.) uses a different measure for each track. For GO-conditioned generation, we measure functional consistency with an ESM-C-based (Candido et al., 2026) GO-reference matching rule that accepts a sequence when its minimum cosine similarity to ten reference proteins sharing the designated GO terms is at least the minimum among those references. For binder generation, we use ipTM from Protenix-v1, following established practice (Team et al., 2025; Fang et al., 2025), and the binding-site subtask further requires the binder to contact at least half of the specified site residues within a 5 Å heavy-atom cutoff. We additionally report novelty against natural proteins, and the provide full set of validators and thresholds in Appendix E.

4.4 Quality control

We apply a two-stage expert review to ensure that each subtask and query in VibeProteinBench reflects a meaningful protein design competence. In the primary consultation, domain experts select the subtasks that define the scope of each stage. In a secondary pass, a separate group of experts who did not participate in the primary consultation scores each query with its evolutionary and structural context, the tools used to construct it, and its answer evaluation method. For each query, experts give

Table 1: Pass rates (%) for each evaluation criterion across the three stages of VIBEPROTEINBENCH, reported for general-purpose and domain-specialized LLMs without tools. Pass-rate columns are shaded, and within each, the best and second-best results are shown in **bold** and underline.

	Recognition	Engineering				Generation				
Models	Pass rate	Hard const.	Rat. align.	In silico valid.	Pass rate	Seq. valid.	Fold valid.	Func. Cons.	Novelty	Pass rate
General-purpose										
GPT-5.4	**82.6**	86.7	**76.7**	**56.7**	**56.7**	95.7	54.8	18.6	67.2	7.6
Gemini3.1-Pro	56.0	63.3	51.7	38.3	38.3	99.0	**74.8**	18.1	61.2	14.8
Opus-4.6	75.4	5.0	5.0	3.3	3.3	**99.5**	38.6	13.3	81.7	2.9
DeepSeek-V4-Pro	73.8	66.7	48.3	40.0	40.0	**99.5**	65.7	17.6	70.9	11.4
DeepSeek-V3.2	70.7	65.0	55.0	40.0	40.0	98.6	61.4	11.4	79.9	5.7
Kimi-K2.5	58.4	6.7	3.3	1.7	1.7	94.8	39.0	10.5	87.1	4.8
Qwen3.5-397B-A17B	72.1	63.3	58.3	45.0	45.0	**99.5**	56.7	12.4	89.3	5.7
Qwen3.5-9B	40.8	1.7	0.0	0.0	0.0	82.9	43.3	6.2	88.8	2.9
Domain-specialized										
NatureLM-8×7B	26.5	73.3	3.3	3.3	3.3	96.7	55.2	16.7	78.8	**15.7**
SciReasoner-8B	40.5	**91.7**	5.0	3.3	3.3	98.6	46.2	**19.5**	74.6	13.8
TxGemma-9B	40.8	5.0	0.0	0.0	0.0	**99.5**	49.0	8.6	92.0	3.8
TxGemma-27B	33.7	13.3	0.0	0.0	0.0	93.3	37.1	5.2	**96.5**	1.0

Table 2: Pass rates for each evaluation criterion across the three stages of VIBEPROTEINBENCH with and without tool use. The best performance is represented by bold.

		Recognition	Engineering				Generation				
Models	Setup	Pass rate	Hard const.	Rat. align.	In silico valid.	Pass rate	Seq. valid.	Fold valid.	Func. Cons.	Novelty	Pass rate
GPT-5.4	w/o tool	82.6	86.7	76.7	56.7	56.7	95.7	54.8	18.6	67.2	7.6
	w/ tool	**84.7**	95.0	75.0	60.0	**60.0**	97.1	57.6	23.8	82.5	**12.4**
Gemini3.1-Pro	w/o tool	56.0	63.3	51.7	38.3	38.3	99.0	74.8	18.1	61.2	14.8
	w/ tool	**82.3**	68.3	60.0	45.0	**45.0**	75.2	65.2	22.9	70.1	**20.0**
DeepSeek-V4-Pro	w/o tool	73.8	66.7	48.3	40.0	40.0	99.5	65.7	17.6	70.9	11.4
	w/ tool	**84.8**	86.7	75.0	55.0	**55.0**	94.8	76.7	28.6	78.9	**23.8**
DeepSeek-V3.2	w/o tool	70.7	65.0	55.0	40.0	40.0	98.6	61.4	11.4	79.9	5.7
	w/ tool	**74.0**	73.3	65.0	56.7	**56.7**	59.5	45.2	14.3	88.2	**12.4**
Qwen3.5-397B-A17B	w/o tool	72.1	63.3	58.3	45.0	45.0	99.5	56.7	12.4	89.3	5.7
	w/ tool	**76.2**	88.3	80.0	65.0	**65.0**	63.3	43.3	8.6	84.2	**6.2**

a yes/no judgment against 5-6 rubric items spanning realism/meaningfulness, clarity, and in silico verifiability. We retain subtasks whose mean score across evaluators is at least 0.5, ranging from 0 (unanimous rejection) to 1 (unanimous endorsement). We provide the full rubric in Appendix G.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate eight general-purpose LLMs on VIBEPROTEINBENCH under two settings: a tool-free setting that isolates their internal protein-design competence, and a tool-augmented agent setting based on ReAct (Yao et al., 2023) whose tool list we detail in Appendix A. The general-purpose group comprises GPT-5.4 (OpenAI, 2026a), Gemini3.1-Pro (DeepMind, 2026), Opus-4.6 (Anthropic, 2026), DeepSeek-V4-Pro (Deepseek-AI, 2026), DeepSeek-V3.2 (Liu et al., 2025a), Kimi-K2.5 (Team et al., 2026a), and Qwen3.5 (Qwen, 2026) at 397B-A17B and 9B scales. We further evaluate a domain-specialized group: NatureLM-8×7B (Xia et al., 2025), SciReasoner-8B (Wang et al., 2025c), and TxGemma-chat (Wang et al., 2025a) at 9B and 27B scales. We additionally include five non-LLM reference baselines, the protein-text multimodal models ProteinDT (Liu et al., 2025c), ProDVa (Liu et al., 2025b), and PAAG (Yuan et al., 2025), and the binder design models BoltzGen (Stark et al., 2025) and BindCraft (Pacesa et al., 2024). We report pass rate as the primary measure, counting a query as passed only when the output satisfies every criterion defined for its subtask in Section 4, and counting a query declined by internal safety filters as a failure. Appendix B reports further analyses, including error-level agent comparisons, unbounded iteration under an advanced setup, tool-argument choices, a temporal-cutoff ablation, and antibody, RNA, and DNA binder generation.

5.2 MAIN RESULTS

Table 1 reports all LLM baselines without tools. Overall, **no model achieves strong performance across all three stages**: the best pass rate reaches 82.6 in recognition but drops to 56.7 in engineering and 15.7 in generation, leaving a substantial portion of engineering and generation queries unsolved even for the strongest models. These failures share a common shape, in which a model satisfies the

Table 3: Pass rates for each evaluation criterion on engineering and generation tasks for non-LLM baselines. BindCraft is evaluated only on protein-target binder generation queries, whereas BoltzGen is evaluated on all binder generation queries.

	Engineering				Generation				
Models	Hard const.	Rat. align.	In silico valid.	Pass rate	Seq. valid.	Fold valid.	Func. Cons.	Novelty	Pass Rate
ProteinDT	0.0	0.0	0.0	0.0	100.0	30.5	4.8	94.2	0.0
PAAG	0.0	0.0	0.0	0.0	100.0	24.3	2.9	95.6	0.5
ProDVa	0.0	0.0	0.0	0.0	82.9	65.2	31.0	43.1	19.0
BindCraft	N/A	N/A	N/A	N/A	98.9	91.1	21.1	96.5	20.0
BoltzGen	N/A	N/A	N/A	N/A	100.0	95.8	36.7	69.7	36.7

surface requirements of a query while missing its substantive one. In engineering, hard-constraint compliance does not entail adherence to the mechanistic rationale: SciReasoner-8B and NatureLM-8×7B reach 91.7 and 73.3 on hard constraints, yet their rationale alignment falls to 5.0 and 3.3. Opus-4.6 fails for two distinct reasons, as its content filter declines 35.0% of its engineering queries and its edits land outside the permitted set in 36.7% of cases. Generation exhibits the same dissociation in a more extreme form. Nearly all models produce valid sequences, yet functional consistency stays below 20 for every model. NatureLM-8×7B and SciReasoner-8B lead the generation pass rate, indicating that pretraining on protein data does transfer, but their functional consistency of 16.7 and 19.5 stays in the similar range as the general-purpose models. Together, these results identify mechanistic and functional grounding as a key bottleneck for current models.

Table 2 further compares the five strongest models with their tool-augmented counterparts. Tool access raises every model-by-stage pass rate, to 84.8 in recognition and 23.8 in generation for DeepSeek-V4-Pro and to 65.0 in engineering for Qwen3.5-397B-A17B. Nevertheless, a substantial fraction of engineering and generation queries remains unsolved, with the best models still failing 35.0% and 76.2%, respectively. Tool access therefore complements but does not substitute for the design competence VIBEPROTEINBENCH probes.

5.3 QUANTITATIVE ANALYSIS

Results of non-LLM generative models. To examine whether the difficulty of VIBEPROTEINBENCH extends beyond LLMs, we evaluate three multimodal text–protein generation models, ProteinDT, PAAG, and ProDVa, which map short functional descriptions to sequences through contrastive text–protein alignment or fragment retrieval (Table 3). Despite their domain-specific architectures, all three score zero on engineering, as they treat engineering queries as de novo design requests and emit sequences distinct from the wild-type. In generation, ProteinDT and PAAG produce highly valid and novel sequences, but their outputs rarely fold or function. Two established binder design models, BindCraft and BoltzGen, predict well-folded binder complexes (91.1 and 95.8 fold validity) and surpass all LLM baselines on binder queries (Table 16), yet their pass rates remain at 20.0 and 36.7, bounded by functional consistency. VIBEPROTEINBENCH therefore remains challenging even for models purpose-built for protein design.

Results on constraint-free engineering. To assess whether models can engineer a protein without expert guidance on where to intervene, we evaluate the constraint-free variant of the engineering queries introduced in Section 4.2 under the same tool-augmented setting. As shown in Table 4, removing the defective and protected positions lowers the pass rate of four out of five models, from 45.0–65.0 to 30.0–55.0. The two largest drops arise from opposite failure modes. Specifically, Qwen3.5-397B-A17B applies generic heuristics such as removing Gly/Cys or substituting alanine without structural verification, which reduces in silico validity. DeepSeek-V3.2 instead withholds answers, with empty responses rising from 8/60 to 24/60 and its median tool calls from 5 to 14 as it searches without committing to a candidate. The difficulty of engineering therefore lies not only in choosing how to mutate a residue, but also in identifying which residues to modify.

Table 4: Pass rates for constraint-free engineering in tool-augmented setting.

Models	Hard const.	In silico valid.	Pass rate
GPT-5.4	96.7	55.0	55.0
Gemini3.1-Pro	56.7	30.0	30.0
DeepSeek-V4-Pro	75.0	55.0	55.0
DeepSeek-V3.2	46.7	30.0	30.0
Qwen3.5-397B-A17B	75.0	38.3	38.3

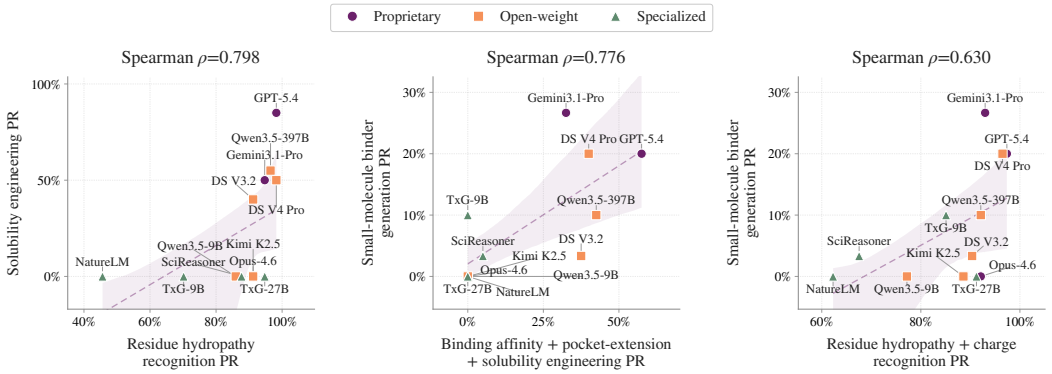

Figure 3: Cross-stage correlations between recognition, engineering, and generation performance of baseline LLMs. DS denotes DeepSeek and TxG indicates TxGemma-chat models.

Cross-stage correlation. Our three-stage design raises the question of whether model performance is associated across stages. We therefore compute Spearman correlations between the pass rates (PR) of biologically motivated subtask pairings. We pair residue hydropathy recognition with solubility engineering, since the latter requires identifying residues whose hydropathy creates surface aggregation liabilities. We pair the aggregate of solubility, binding affinity, and pocket-extension engineering with small-molecule binder generation, since these tasks involve complementary aspects of surface hydropathy, chemical complementarity, and pocket geometry. Finally, we pair residue hydropathy and charge recognition with small-molecule binder generation to examine a direct recognition–generation association. As Figure 3 shows, all three pairs correlate positively, which is consistent with shared physicochemical demands across stages. These cross-stage associations provide empirical support for our benchmark's three-stage design, which connects foundational recognition skills with engineering and generation in an integrated assessment of protein-design competence. Figure 7 reports additional biologically linked pairings.

Failure analysis. To characterize recurring problems in the tool-augmented agent setting, we use GPT-5.6-Sol (OpenAI, 2026c) to annotate each trajectory from the five models in Table 2 with the workflow-level patterns it exhibits. As Figure 4 shows, the most frequent pattern is submitting a sequence without any structural check (55.4%), followed by leaving the binder–target interface unassessed (29.5%). On miniprotein binder generation for 9DQ0, for instance, DeepSeek-V4-Pro folds the binder alone (pLDDT 86.2) and submits it without assessing its interaction with the target. Beyond validation, agents also redesign the target chain rather than the binder (12.9%) or drop the requested ligand or binding-site conditioning from a generation step (6.2%). The latter arises either from unconditional backbone generation, or from conditioning one tool on the requested sites and then submitting a candidate produced by a second generation tool called without that conditioning. Agentic protein design therefore demands both explicit guidance on which properties to verify and a mechanism that maintains the requested constraints attached to the sequence finally submitted, where tool access alone provides neither. Additional examples in Table 5.

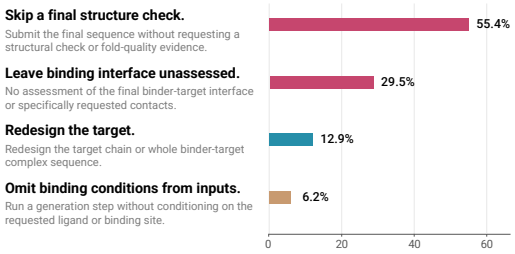

Figure 4: Top four recurring workflow-level failure patterns of the tool-augmented agents. Each trajectory can exhibit multiple patterns, and the reported shares are therefore not mutually exclusive.

6 Conclusion

We presented VibeProteinBench, a language-interfaced benchmark that probes generalist protein design capabilities through three complementary stages: recognition, rationale-guided engineering, and functional generation. Our evaluation shows that generalist vibe protein design remains a substantial open challenge, motivating language-interfaced models that integrate biological understanding, mechanistic reasoning, and grounded generation within a unified framework.

AI USE STATEMENT

In this work, we used generative AI tools for supporting the qualitative analysis and interpretation of our experimental results to help characterize the representative failure types and agent behaviors observed in the tool-augmented setting. We have not used generative AI tools for cleaning or reformatting existing dataset, developing ideas, or designing the experimental methodology, and developing theoretical models, formulating or proving mathematical claims, and translation, are not applicable to this work. Additionally, we used generative AI tools for implementing code for automatically constructing queries, creating and modifying a subset of the images in Figure 2, and editing the manuscript to improve grammar and readability (*e.g.*, correcting awkward phrasing). We have reviewed all AI-assisted work. The AI-assisted failure characterization was cross-checked against the raw trajectories, the AI-generated or AI-edited codes and figure elements were manually inspected, and all AI-suggested text edits were reviewed and approved by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

VIBEPROTEINBENCH is an evaluation benchmark rather than a design system, and we expect its primary effect to be a clearer account of what language-interfaced models can and cannot do in protein design, which in turn supports more careful deployment of such models in therapeutic and industrial workflows. We release the benchmark and evaluation code for research use, and encourage pairing it with the biosecurity screening practices established for protein design.

REPRODUCIBILITY STATEMENT

We include the source code for constructing and evaluating the benchmark in the supplementary materials, and report the full configurations in the appendix.

REFERENCES

Anthropic. Introducing Claude Opus 4.6, February 2026. URL https://www.anthropic.com/news/claude-opus-4-6.

Emily G Baker, Gail J Bartlett, Kathryn L Porter Goff, and Derek N Woolfson. Miniprotein design: past, present, and prospects. *Accounts of Chemical Research*, 50(9):2085–2092, 2017.

Nathaniel R. Bennett, Brian Coventry, Inna Goreshnik, Buwei Huang, Aza Allen, Dionne Vafeados, Ying Po Peng, Justas Dauparas, Minkyung Baek, Lance Stewart, Frank DiMaio, Steven De Munck, Savvas N. Savvides, and David Baker. Improving de novo protein binder design with deep learning. *Nature Communications*, 14(1):2625, 2023.

Nathaniel R Bennett, Joseph L Watson, Robert J Ragotte, Andrew J Borst, DéJenaé L See, Connor Weidle, Riti Biswas, Yutong Yu, Ellen L Shrock, Russell Ault, et al. Atomically accurate de novo design of antibodies with rfdiffusion. *Nature*, 649(8095):183–193, 2026.

Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. Genbank. *Nucleic acids research*, 41(D1):D36–D42, 2012.

Matthias Blum, Hsin-Yu Chang, Sara Chuguransky, Teresa Grego, Selvam Kandasaamy, Alex Mitchell, Gaurav Nuka, Typhaine Paysan-Lafosse, Matloob Qureshi, Surabhi Raj, et al. The interpro protein families and domains database: 20 years on. *Nucleic Acids Research*, 49(D1): D344–D354, 2021.

David F. Burke, Patrick Bryant, Inigo Barrio-Hernandez, Danish Memon, Gabriele Pozzati, Aditi Shenoy, Wensi Zhu, Alistair S. Dunham, Pascal Albanese, Andrew Keller, Richard A. Scheltema, James E. Bruce, Alexander Leitner, Petras Kundrotas, Pedro Beltrao, and Arne Elofsson. Towards a structurally resolved human protein interaction network. *Nature Structural & Molecular Biology*, 30(2):216–225, 2023.

Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M Duarte, Shuchismita Dutta, Zukang Feng, Sai Ganesan, David S Goodsell, Sutapa Ghosh, Rachel Kramer Green, Vladimir Guranović, Dmytro Guzenko, Brian P Hudson, Catherine L Lawson, Yuhe Liang, Robert Lowe, Harry Namkoong, Ezra Peisach, Irina Persikova, Chris Randle, Alexander Rose, Yana Rose, Andrej Sali, Joan Segura, Monica Sekharan, Chenghua Shao, Yi-Ping Tao, Maria Voigt, John D Westbrook, Jasmine Y Young, Christine Zardecki, and Marina Zhuravleva. RCSB protein data bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Research*, 49(D1):D437–D451, 2021.

Jasper Butcher, Rohith Krishna, Raktim Mitra, Rafael Isaac Brent, Yanjing Li, Nathaniel Corley, Paul T. Kim, Jonathan Funk, Simon Valentin Mathis, Saman Salike, Aiko Muraishi, Helen Eisenach, Tuscan Rock Thompson, Jie Chen, Yuliya Politanska, Enisha Sehgal, Brian Coventry, Odin Zhang, Bo Qiang, Kieran Didi, Maxwell Kazman, Frank DiMaio, and David Baker. De novo design of all-atom biomolecular interactions with RFdiffusion3. *bioRxiv*, 2025.

Bryce Cai, Geetha Jeyapragasan, Samira Nedungadi, Jake Yukich, and Seth Donoughe. Agentic BAIM-LLM evaluation (ABLE): Benchmarking LLM use of protein design tools. In *NeurIPS 2025 Workshop on Biosecurity Safeguards for Generative AI*, 2025.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z. Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language modeling materializes a world model of protein biology. *bioRxiv*, 2026.

Eli M Carrami and Sahand Sharifzadeh. Pqa: Zero-shot protein question answering for free-form scientific enquiry with large language models. *arXiv preprint arXiv:2402.13653*, 2024.

Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J Gray. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics*, 26(5):689–691, 2010.

Hua Cheng, R. Dustin Schaeffer, Yuxing Liao, Lisa N. Kinch, Jimin Pei, Shuoyong Shi, Bong-Hyun Kim, and Nick V. Grishin. Ecod: An evolutionary classification of protein domains. *PLOS Computational Biology*, 10(12):e1003926, 2014.

Yehlin Cho, Justas Dauparas, Kotaro Tsuboyama, Gabriel J Rocklin, and Sergey Ovchinnikov. Stable de novo protein design via joint conformational landscape and sequence optimization. *Nature Communications*, 2025.

Yehlin Cho, Kotaro Tsuboyama, Theodore J. Litberg, Michelle D. Jung, Adunoluwa Obisesan, Qian Wang, Claire M. Phoumyvong, Jane Thibeault, Sergey Ovchinnikov, and Gabriel J. Rocklin. Accurate protein stability prediction for small domains using mega-scale experiments. *bioRxiv*, 2026. Preprint.

Joel J Chubb, Aimee L Boyle, and Katherine I Albanese. Rational protein design. *Current Opinion in Structural Biology*, 97:103224, 2026.

Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel JL de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

Sophie Colette, Jaldert François, Bart De Moor, and Vera van Noort. Ogtfinder: A curated growth temperature data set and its application to predict optimal growth temperatures of bacteria and archaea. *Journal of Chemical Information and Modeling*, 2026.

Tudor-Stefan Cotet, Igor Krawczuk, Martin Pacesa, Lennart Nickel, Bruno E Correia, Nikhil Haas, Ahmad Qamar, Chance A Challacombe, Patrick Kidger, Constance Ferragu, et al. Crowdsourced protein design: Lessons from the adaptyv egfr binder competition. *bioRxiv*, pp. 2025–04, 2025.

Fengyuan Dai, Shiyang You, Yudian Zhu, Yuan Gao, Lihao Fu, Xibin Zhou, Jin Su, Chentong Wang, Yuliang Fan, Xiaoxiao Ma, et al. Toward de novo protein design from natural language. *BioRxiv*, 2024.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Justas Dauparas, Gyu Rie Lee, Robert Pecoraro, Linna An, Ivan Anishchenko, Cameron Glasscock, and David Baker. Atomic context-conditioned protein sequence design using ligandmpnn. *Nature Methods*, 22(4):717–723, 2025.

Google DeepMind. Gemini 3.1 pro - model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, February 2026.

Deepseek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf, 2026.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M. Bronstein, Martin Steinegger, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *The Fourteenth International Conference on Learning Representations*, 2026.

Henry Dieckhaus, Michael Brocidiacono, Nicholas Z Randolph, and Brian Kuhlman. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proceedings of the national academy of sciences*, 121(6):e2314853121, 2024.

Adibvafa Fallahpour, Arman Seyed-Ahmadi, Parsa Idehpour, Omar Ibrahim, Purav Gupta, Jack Naimer, Kevin Zhu, Arnav Shah, Shihao Ma, Abhinav Adduri, et al. Bioreason-pro: Advancing protein function prediction with multimodal biological reasoning. *bioRxiv*, 2026.

Minchao Fang, Chentong Wang, Jungang Shi, Fengbai Lian, Qihan Jin, Zhe Wang, Yanzhe Zhang, Peipei Chen, Zhanyuan Cui, Yanjun Wang, et al. Halludesign: Protein optimization and de novo design via iterative structure hallucination and sequence design. *bioRxiv*, pp. 2025–11, 2025.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*, 2023.

Zhangyang Gao, Cheng Tan, Yijie Zhang, Xingran Chen, Lirong Wu, and Stan Z. Li. Proteininvbench: Benchmarking protein inverse folding on diverse tasks, models, and metrics. In *Advances in Neural Information Processing Systems*, 2023.

Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12 (1):3168, 2021.

Adi Goldenzweig, Moshe Goldsmith, Shannon E. Hill, Or Gertman, Paola Laurino, Yacov Ashani, Orly Dym, Tamar Unger, Shira Albeck, Jaime Prilusky, Raquel L. Lieberman, Amir Aharoni, Israel Silman, Joel L. Sussman, Dan S. Tawfik, and Sarel J. Fleishman. Automated structure- and sequence-based design of proteins for high bacterial expression and stability. *Molecular Cell*, 63 (2):337–346, 2016.

Konstantinos Grigorakis, Christina Ferousi, and Evangelos Topakas. Protein engineering for industrial biocatalysis: principles, approaches, and lessons from engineered petases. *Catalysts*, 15(2):147, 2025.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. In *Advances in Neural Information Processing Systems*, 2023.

Fei Gu, Zi Liang, Jiahao Ma, and Hongzong Li. The matthew effect of ai programming assistants: A hidden bias in software evolution. In *International Conference on Learning Representations*, 2026.

João Victor da Silva Guerra, Helder Veras Ribeiro-Filho, Gabriel Ernesto Jara, Leandro Oliveira Bortot, José Geraldo de Carvalho Pereira, and Paulo Sérgio Lopes-de Oliveira. pykvfinder: an efficient and integrable python package for biomolecular cavity detection and characterization in data science. *BMC bioinformatics*, 22(1):607, 2021.

Tymor Hamamsy, Meet Barot, James T Morton, Martin Steinegger, Richard Bonneau, and Kyunghyun Cho. Learning sequence, structure, and function representations of proteins with language models. *bioRxiv*, 2023.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Bing He, Chenchen Qin, Yu Zhao, Long-Kai Huang, Zihan Wu, Fang Wang, Fandi Wu, Fan Yang, and Jianhua Yao. Functional protein design and enhancement with ontology reinforcement iteration. *Nature Communications*, 17:4158, 2026.

Steven Henikoff and Jorja G. Henikoff. Position-based sequence weights. *Journal of Molecular Biology*, 243(4):574–578, 1994. doi: 10.1016/0022-2836(94)90032-9.

Brian L Hie and Kevin K Yang. Adaptive machine learning for protein engineering. *Current opinion in structural biology*, 72:145–152, 2022.

Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature biotechnology*, 42(2):275–283, 2024.

Ala Jararweh, Oladimeji Macaulay, David Arredondo, Yue Hu, Luis E Tafoya, Kushal Virupakshappa, and Avinash Sahu. Protein2text: Resampling mechanism to translate protein sequences into human-interpretable text. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pp. 918–937, 2025.

Wolfgang Kabsch and Chris Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

Jeonghyeon Kim and Philip Romero. Benchmarking and behavioral characterization of llm agents for protein design. *bioRxiv*, pp. 2026–05, 2026.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. Pubchem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 2023.

Julia Koehler Leman, Pawel Szczerbiak, P Douglas Renfrew, Vladimir Gligorijevic, Daniel Berenberg, Tommi Vatanen, Bryn C Taylor, Chris Chandler, Stefan Janssen, Andras Pataki, et al. Sequence-structure-function relationships in the microbial protein universe. *Nature communications*, 14(1): 2351, 2023.

Ivan V Korendovych. Rational and semirational protein design. *Protein engineering: methods and protocols*, pp. 15–23, 2017.

Jiahao Kuang, Nuowei Liu, Jie Wang, Changzhi Sun, Tao Ji, and Yuanbin Wu. Pdfbench: A benchmark for de novo protein design from function. *arXiv preprint arXiv:2505.20346*, 2025.

Tim Kucera, Matteo Togninalli, and Laetitia Meng-Papaxanthos. Conditional generative modeling for de novo protein design with hierarchical functions. *Bioinformatics*, 38(13):3454–3461, 2022.

Maxat Kulmanov, Francisco J Guzmán-Vega, Paula Duek Roggli, Lydie Lane, Stefan T Arold, and Robert Hoehndorf. Protein function prediction as approximate semantic entailment. *Nature Machine Intelligence*, 6(2):220–228, 2024.

Gregory W Kyro, Tianyin Qiu, and Victor S Batista. A model-centric review of deep learning for protein design. *arXiv preprint arXiv:2502.19173*, 2025.

Michael C. Lawrence and Peter M. Colman. Shape complementarity at protein/protein interfaces. *Journal of Molecular Biology*, 234(4):946–950, 1993.

Jie Li, Youyang Hou, Laura Lin, Ruihao Zhu, Hancheng Cao, and Abdallah El Ali. Vibe coding in product teams: Reconfiguring ai-assisted workflows, prototyping, and collaboration. In *Proceedings of the Annual Symposium on Human-Computer Interaction for Work*, 2026.

Linyi Li, Shijie Geng, Zhenwen Li, Yibo He, Hao Yu, Ziyue Hua, Guanghan Ning, Siwei Wang, Tao Xie, and Hongxia Yang. Infibench: Evaluating the question-answering capabilities of code large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Nuowei Liu, Jiahao Kuang, Yanting Liu, Tao Ji, Changzhi Sun, Man Lan, and Yuanbin Wu. Protein design with dynamic protein vocabulary. In *Advances in Neural Information Processing Systems*, 2025b.

Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, et al. A text-guided protein design framework. *Nature Machine Intelligence*, 7(4):580–591, 2025c.

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. Prollama: A protein large language model for multi-task protein language processing. *IEEE Transactions on Artificial Intelligence*, 2025.

Zicheng Ma, Chuanliu Fan, Zhicong Wang, Zhenyu Chen, Xiaohan Lin, Yanheng Li, Shihao Feng, Ziqiang Cao, Jun Zhang, and Yi Qin Gao. Prottex: Structure-in-context reasoning and editing of proteins with large language models. *Journal of Chemical Information and Modeling*, 65(13): 6599–6612, 2025.

Christian Meske, Tobias Hermanns, Esther Von der Weiden, Kai-Uwe Loser, and Thorsten Berger. Vibe coding as a reconfiguration of intent mediation in software development: Definition, implications, and research agenda. *IEEE access*, 13:213242–213259, 2025.

Geraldene Munsamy, Sebastian Lindner, Philipp Lorenz, and Noelia Ferruz. Zymctrl: a conditional language model for the controllable generation of artificial enzymes. In *NeurIPS machine learning in structural biology workshop*, 2022.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Divya Nori, Anisha Parsan, Caroline Uhler, and Wengong Jin. Bindenergycraft: Casting protein structure predictors as energy-based models for binder design. In *ICML Generative AI and Biology (GenBio) Workshop*, 2025.

Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nature biotechnology*, 42(2):216–228, 2024.

OpenAI. Codex CLI. https://developers.openai.com/codex/cli/. Accessed: September 12, 2026.

OpenAI. Accelerating life sciences research with retro biosciences. https://openai.com/index/accelerating-life-sciences-research-with-retro-biosciences/, August 2025.

OpenAI. Introducing GPT-5.4, March 2026a. URL https://openai.com/index/introducing-gpt-5-4/.

OpenAI. Introducing gpt-rosalind for life sciences research. https://openai.com/index/introducing-gpt-rosalind/, 2026b.

OpenAI. GPT-5.6 system card. Technical report, OpenAI, July 2026c. URL https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf. Updated August 19, 2026.

Katarzyna Ożga and Łukasz Berlicki. Design and engineering of miniproteins. *ACS bio & med Chem Au*, 2(4):316–327, 2022.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, et al. Bindcraft: one-shot design of functional protein binders. *BioRxiv*, pp. 2024–09, 2024.

Haining Pan, James V Roggeveen, Erez Berg, Juan Felipe Carrasquilla Alvarez, Debanjan Chowdhury, Surya Ganguli, Federico Ghimenti, Juraj Hasik, Henry S. Hunt, Hong-Chen Jiang, Mason Kamb, Ying-Jer Kao, Ehsan Khatami, Michael J Lawler, Di Luo, Titus Neupert, Xiaoliang Qi, Michael Brenner, and Eun-Ah Kim. CMT-benchmark: A benchmark for condensed matter theory built by expert researchers. In *The Fourteenth International Conference on Learning Representations*, 2026.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Douglas E. V. Pires, Tom L. Blundell, and David B. Ascher. Platinum: a database of experimentally measured effects of mutations on structurally defined protein–ligand complexes. *Nucleic Acids Research*, 43(D1):D387–D391, 2015.

Manvitha Ponnapati, Sam Cox, Cade W Gordon, Michael J Hammerling, Siddharth Narayanan, Jon M Laurent, James D Braza, Michaela M Hinks, Michael D Skarlinski, Samuel G Rodriques, et al. Proteincrow: A language model agent that can design proteins. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025.

Rui Qing, Shilei Hao, Eva Smorodina, David Jin, Arthur Zalevsky, and Shuguang Zhang. Protein design: From the aspect of water solubility and stability. *Chemical Reviews*, 122(18):14085–14179, 2022.

Qwen. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026.

RNAcentral Consortium. Rnacentral 2021: secondary structure integration, improved sequence search and new member databases. *Nucleic acids research*, 49(D1):D212–D220, 2021.

Dingyi Rong, Zijian Chen, Qi Jia, Kaiwei Zhang, Haotian Lu, Guangtao Zhai, and Ning Liu. Liveproteinbench: A contamination-free benchmark for assessing models' specialized capabilities in protein science. *arXiv preprint arXiv:2512.22257*, 2025.

Sankaran Sandhya, Richa Mudgal, Gayatri Kumar, Ramanathan Sowdhamini, and Narayanaswamy Srinivasan. Protein sequence design and its applications. *Current opinion in structural biology*, 37: 71–80, 2016.

Isaac Sappington, Martin Toul, David S Lee, Stephanie A Robinson, Inna Goreshnik, Clara McCurdy, Tung Ching Chan, Nic Buchholz, Buwei Huang, Dionne Vafeados, et al. Improved protein binder design using β-pairing targeted rfdiffusion. *Nature Communications*, 2026.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl_1):D431–D433, 2004.

Robert Shaw, Samuel D Love, and Claire D McWhite. Evaluating pretrained protein language model embeddings as proxies for functional similarity. *Journal of Molecular Evolution*, 93(6):765–776, 2025.

Yiqing Shen, Zan Chen, Michail Mamalakis, Luhan He, Haiyang Xia, Tianbin Li, Yanzhou Su, Junjun He, and Yu Guang Wang. A fine-tuning dataset and benchmark for large language models for protein understanding. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2024.

Christian J. A. Sigrist, Edouard de Castro, Lorenzo Cerutti, Béatrice A. Cuche, Nicolas Hulo, Alan Bridge, Lydie Bougueleret, and Ioannis Xenarios. New and continuing developments at PROSITE. *Nucleic Acids Research*, 41(D1):D344–D347, 2013.

Matthew Sinclair, Moeen Meigooni, Archit Vasan, Ozan Gokdemir, Xinran Lian, Heng Ma, Yadu Babuji, Alexander Brace, Khalid Hossain, Carlo Siebenschuh, et al. Scalable agentic reasoning for designing biologics targeting intrinsically disordered proteins. *arXiv preprint arXiv:2512.15930*, 2025.

Zhenqiao Song, Ramith Hettiarachchi, Chuan Li, Jianwen Xie, and Lei Li. Instructpro: Natural language guided ligand-binding protein design. *arXiv preprint arXiv:2506.09332*, 2025.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, et al. Boltzgen: Toward universal binder design. *bioRxiv*, pp. 2025–11, 2025.

Yang Tan, Chen Liu, Jingyuan Gao, Banghao Wu, Mingchen Li, Ruilin Wang, Lingrong Zhang, Huiqun Yu, Guisheng Fan, Liang Hong, et al. Venusfactory: A unified platform for protein engineering data retrieval and language model fine-tuning. *arXiv preprint arXiv:2503.15438*, 2025.

Yang Tan, Lingrong Zhang, Mingchen Li, Yuanxi Yu, Bozitao Zhong, Bingxin Zhou, Nanqing Dong, and Liang Hong. Self-evolving ai agents for protein discovery and directed evolution. *arXiv preprint arXiv:2603.27303*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026a.

Protenix Team, Milong Ren, Jinyuan Sun, Jiaqi Guan, Cong Liu, Chengyue Gong, Yuzhe Wang, Lan Wang, Qixu Cai, Wenzhi Ma, et al. Pxdesign: Fast, modular, and accurate de novo design of protein binders. *BioRxiv*, pp. 2025–08, 2025.

Protenix Team, Yuxuan Zhang, Chengyue Gong, Hanyu Zhang, Wenzhi Ma, Zhenyu Liu, Xinshi Chen, Jiaqi Guan, Lan Wang, Yanping Yang, et al. Protenix-v1: Toward high-accuracy open-source biomolecular structure prediction. *bioRxiv*, 2026b.

Magdalena Teufl, Charlotte U Zajc, and Michael W Traxlmayr. Engineering strategies to overcome the stability–function trade-off in proteins. *ACS synthetic biology*, 11(3):1030–1039, 2022.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Chuanrui Wang, Bozitao Zhong, Zuobai Zhang, Narendra Chaudhary, Sanchit Misra, and Jian Tang. Pdb-struct: A comprehensive benchmark for structure-based protein design. *arXiv preprint arXiv:2312.00080*, 2023.

Eric Wang, Samuel Schmidgall, Paul F Jaeger, Fan Zhang, Rory Pilgrim, Yossi Matias, Joelle Barral, David Fleet, and Shekoofeh Azizi. Txgemma: Efficient and agentic llms for therapeutics. *arXiv preprint arXiv:2504.06196*, 2025a.

Fiona Y Wang, Di Sheng Lee, David L Kaplan, and Markus J Buehler. Swarms of large language model agents for protein sequence design with experimental validation. *arXiv preprint arXiv:2511.22311*, 2025b.

Yizhou Wang, Chen Tang, Han Deng, Jiabei Xiao, Jiaqi Liu, Jianyu Wu, Jun Yao, Pengze Li, Encheng Su, Lintao Wang, et al. Scireasoner: Laying the scientific reasoning ground across disciplines. *arXiv preprint arXiv:2509.21320*, 2025c.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Juntong Wu, Zijing Liu, He Cao, Li Hao, Bin Feng, Zishan Shu, Ke Yu, Li Yuan, and Yu Li. Rethinking text-based protein understanding: Retrieval or llm? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23737–23757, 2025.

Yingce Xia, Peiran Jin, Shufang Xie, Liang He, Chuan Cao, Renqian Luo, Guoqing Liu, Yue Wang, Zequn Liu, Yuan-Jyue Chen, et al. Nature language model: deciphering the language of nature for scientific discovery. *arXiv preprint arXiv:2502.07527*, 2025.

Yijia Xiao, Edward Sun, Yiqiao Jin, Qifan Wang, and Wei Wang. Proteingpt: Multimodal llm for protein property prediction and structure understanding. *arXiv preprint arXiv:2408.11363*, 2024.

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Ma Chang, Runcheng Liu, and Jian Tang. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. In *Advances in Neural Information Processing Systems*, 2022.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. Protst: Multi-modality learning of protein sequences and biomedical texts. In *International conference on machine learning*, pp. 38749–38767. PMLR, 2023.

Jason Yang, Aadyot Bhatnagar, Jeffrey A Ruffolo, and Ali Madani. Function-guided conditional generation using protein language models with adapters. *arXiv preprint arXiv:2410.03634*, 2024a.

Jason Yang, Ariane Mora, Shengchao Liu, Bruce J Wittmann, Anima Anandkumar, Frances H Arnold, and Yisong Yue. Care: a benchmark suite for the classification and retrieval of enzymes. In *Advances in Neural Information Processing Systems*, 2024b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Fei Ye, Zaixiang Zheng, Dongyu Xue, Yuning Shen, Lihao Wang, Yiming Ma, Yan Wang, Xinyou Wang, Xiangxin Zhou, and Quanquan Gu. Proteinbench: A holistic evaluation of protein foundation models. *arXiv preprint arXiv:2409.06744*, 2024.

Chaohao Yuan, Songyou Li, Geyan Ye, Yikun Zhang, Long-Kai Huang, Wenbing Huang, Wei Liu, Jianhua Yao, and Yu Rong. Annotation-guided protein design with multi-level domain alignment. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2025.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, 2024.

Xibin Zhou, Chenchen Han, Yingqi Zhang, Huan Du, Jiayuan Tian, Jin Su, Renju Liu, Kai Zhuang, Shiyu Jiang, Anthony Gitter, et al. Decoding the molecular language of proteins with evolla. *bioRxiv*, pp. 2025–01, 2025.

Le Zhuo, Zewen Chi, Minghao Xu, He-Yan Huang, Jianan Zhao, Heqi Zheng, Conghui He, Xian-Ling Mao, and Wentao Zhang. Protllm: An interleaved protein-language llm with protein-as-word pre-training. In *Annual Conference of the Association for Computational Linguistics*, 2024.

SUPPLEMENTARY MATERIALS

A EXPERIMENTAL SETTING

We evaluate all LLMs in a zero-shot setting using benchmark queries as the user prompt, without task-specific fine-tuning or in-context examples. GPT-5.4 (OpenAI, 2026a), Gemini3.1-Pro (DeepMind, 2026), Claude Opus-4.6 (Anthropic, 2026), DeepSeek-V4-Pro (Deepseek-AI, 2026), DeepSeek-V3.2 (Liu et al., 2025a), Kimi-K2.5 (Team et al., 2026a), and Qwen3.5 (Qwen, 2026) are accessed via the OpenRouter API. NatureLM-8×7B (Xia et al., 2025), SciReasoner-8B (Wang et al., 2025c), and TxGemma (Wang et al., 2025a) are run locally through HuggingFace Transformers with SGLang (Zheng et al., 2024). Unless otherwise specified, decoding uses deterministic inference with temperature 0 and a maximum of 65,536 new tokens, with up to five retries using exponential backoff. We enable internal reasoning (*i.e.*, thinking tokens) for GPT-5.4, Gemini3.1-Pro, DeepSeek-V4-Pro, DeepSeek-V3.2, Kimi-K2.5, and Qwen3.5-397B-A17B. For Opus-4.6 and Qwen3.5-9B, however, we disable reasoning, as enabling it yielded an excessive number of empty responses. Local models are run in bfloat16. For non-LLM multimodal baselines, we use the released checkpoints and inference implementations without task-specific fine-tuning. PAAG (Yuan et al., 2025) is evaluated through its native text-conditioned sequence generation interface with stochastic sampling. ProDVa (Liu et al., 2025b) uses the released inference configuration: stochastic decoding with temperature 0.7, top-k 950, and a maximum of 256 new tokens. ProteinDT-T5 (Liu et al., 2025c) follows the released T5 decoder configuration with temperature 1.0, top-k 40, top-p 0.9, repetition penalty 1.0, and one beam; we generate 8 candidates and select the one with the highest CLAP similarity. For non-LLM binder generation baselines, we use structure-conditioned binder design pipelines for BoltzGen (Stark et al., 2025) and BindCraft (Pacesa et al., 2024), with 500 diffusion steps for BoltzGen and up to 140 sequence-optimization iterations across four stages (75, 45, 5, and 15 iterations, respectively) for BindCraft. BindCraft is evaluated on the 90 protein-target binder queries, whereas BoltzGen additionally supports the 30 small-molecule-target queries. We provide the corresponding targets and binding-site constraints, with binder lengths restricted to 40–80 residues for miniproteins and 30–150 residues otherwise. Both baselines request one design per query without benchmark-score-based candidate selection.

For the tool-augmented setting, we use the ReAct (Yao et al., 2023) framework with a maximum of 30 iterations per query. Each model has access to the same tool set spanning protein design, validation, analysis, and workflow management:

- Backbone generation and redesign: RFdiffusion (Watson et al., 2023), RFdiffusion3 (Butcher et al., 2025)
- Inverse folding and sequence redesign: ProteinMPNN (Dauparas et al., 2022), LigandMPNN (Dauparas et al., 2025)
- Sequence masked-redesign: ESM2 (Lin et al., 2023)
- Binder design and redesign: BoltzGen (Stark et al., 2025), PXDesign (Team et al., 2025)
- Structure prediction: Boltz2 (Passaro et al., 2025), ESMFold2 (Candido et al., 2026)
- Small-molecule docking: AutoDock Vina (Trott & Olson, 2010)
- Physics-based scoring and structural relaxation: PyRosetta (Chaudhury et al., 2010)

We additionally provide analysis tools: `family_profile` (sequence composition, conservation, and candidate similarity to a family profile), `sequence_analysis` (composition, charge, motifs, residue properties, and sequence liabilities), `structure_analysis` (residue exposure, contacts, interfaces, and structural liabilities), `protein_qc` (sequence-level biophysical properties and liabilities), `residue_lookup` (residue identities and properties at specified positions), and `tmalign` (pairwise structural alignment and similarity).

Workflow tools include `sequence_edit` (applies specified substitutions and checks wild-type residue consistency), `merge_candidates` (merges and deduplicates candidate lists), `objective_rank` (ranks candidates using agent-specified criteria), and `search_sequence` (retrieves sequences matching GO or InterPro terms, with a default cutoff retaining UniProt entries dated on or before 2025-09-01).

Table 5: Examples of recurring workflow patterns in failed ReAct trajectories. Each trajectory can exhibit multiple patterns, and the reported shares are therefore not mutually exclusive.

Patterns	Examples	Share (%)
Skip a fold check	**GPT-5.4 / 4AT2:** Used `residue_lookup` to inspect the input sequence, then `sequence_edit` to introduce five substitutions. Submitted the resulting 490-residue mutant directly, without predicting or checking its structure.	55.4
Leave binding interface unassessed	**DeepSeek-V4-Pro / 9DQ0:** Folded the target with ESMFold2 and used PXDesign to generate a 60-residue binder against that structure. Folded the binder alone with ESMFold2 (pLDDT 86.2), then submitted it without assessing the binder–target interface.	29.5
Redesign the target	**Qwen3.5-397B / 7M59:** Used RFdiffusion3 to generate a 60-residue binder backbone alongside the 65-residue target. Passed the entire complex to ProteinMPNN without restricting redesign to the binder chain, changing both chains, then submitted the combined 125-residue output as the binder.	12.9
Omit binding conditions from design inputs	**Qwen3.5-397B / 8HE6:** Generated a backbone with RFdiffusion3 conditioned on ten target-site residues, then a sequence with ProteinMPNN. Subsequently started a separate PXDesign run from the original target structure without specifying those site constraints, and submitted its new 60-residue candidate instead.	6.2
Pass JSON/TSV files as FASTA inputs	**DeepSeek-V4-Pro / 2ZEJ:** Retrieved candidates with `search_sequence`, then passed the returned JSON file to `family_profile` through `alignment_path`, which explicitly requires aligned FASTA. After the file-format rejection, submitted its final answer without retrying the tool.	4.3
Pass structure or search-result files as YAML configurations.	**Qwen3.5-397B / CID 177867006:** Obtained a structure from Boltz2, then passed its mmCIF file to boltzgen through `target_yaml`, which explicitly requires a YAML design specification. After the parsing error, switched to PXDesign and RFdiffusion3 without correcting the BoltzGen input.	1.5

B Additional Experimental Results

Trajectory examples for each workflow failure pattern. We present per-pattern examples of the recurring workflow patterns of Figure 4 in Table 5. GPT-5.4 introduces five substitutions and submits the resulting mutant without ever predicting its structure, and DeepSeek-V4-Pro obtains a confidently folded binder yet never evaluates the complex it was asked to bind. In the remaining two examples, the design object itself drifts from the specification, as ProteinMPNN is applied to the entire complex rather than the binder chain, and a second generation run is launched without the site constraints that conditioned the first. Additionally, agents pass JSON/TSV files as FASTA inputs (4.3%) or structure/search-result files as YAML configurations (1.5%), but these two argument-specification errors are less frequent than the aforementioned four patterns. Overall, these examples motivate explicit guidance on which properties to verify and a mechanism that keeps the requested constraints attached to the sequence finally submitted.

Effect of unbounded iteration with an advanced agent setup. To test whether the remaining failures in Table 2 reflect a limited iteration budget and agent setup rather than a limit of design com-

Table 6: Generation performance and token usage of ReAct and advanced Codex CLI harness on the same 30-query subset. Performance values are percentages; token counts are summed over all 30 queries and reported in millions.

		Generation				Token usage (M)			
Models	Setup	Seq. valid.	Fold valid.	Func. Cons.	Pass rate	Input	Output	Total	vs. ReAct
DeepSeek-V4-Pro	ReAct (30 iterations)	100.0	93.3	46.7	43.3	5.3	0.3	5.6	–
	Codex (iteration-free)	100.0	90.0	56.7	**50.0**	904.4	7.3	911.8	×**163.6**
DeepSeek-V3.2	ReAct (30 iterations)	63.3	50.0	26.7	26.7	9.5	0.2	9.7	–
	Codex (iteration-free)	100.0	90.0	33.3	**30.0**	1,309.7	12.6	1,322.2	×**136.6**
Qwen3.5-397B-A17B	ReAct (30 iterations)	73.3	53.3	13.3	13.3	3.0	0.1	3.1	–
	Codex (iteration-free)	100.0	90.0	43.3	**40.0**	308.4	1.2	309.6	×**101.0**

Table 7: Selected observable workflow failures and top-five tools of ReAct-based DeepSeek-V4-Pro and Qwen3.5-397B-A17B on generation tasks. This comparison highlights primary failure categories with higher shares in Qwen3.5-397B-A17B.

Selected observable workflow failures (%)

Failure types	DeepSeek-V4-Pro	Qwen3.5-397B-A17B
End without an evaluable final sequence	5.2	**36.7**
Submit the final sequence without directly validating it	20.0	**48.1**
Inverse-fold both target and binder to produce the final answer	0.0	**31.1**
Finish without obtaining the requested site-contact evidence	10.0	**93.3**

Top five tools by query-level usage

DeepSeek-V4-Pro: `protein_qc`, `esmfold2`, `sequence_analysis`, `search_sequence`, `pxdesign`

Qwen3.5-397B-A17B: `esmfold2`, `search_sequence`, `inverse_folding`, `rfdiffusion3`, `pxdesign`

petence, we re-run the 30-query generation subset under an advanced Codex CLI harness (OpenAI), which exposes the same tool list as ReAct but adds skills instructing the agent to keep iterating until its acceptance criteria are met instead of terminating after a failed attempt. As shown in Table 6, this raises the pass rate for all three models, from 43.3 to 50.0, from 26.7 to 30.0, and from 13.3 to 40.0, but sacrifices at 101.0 to 163.6 times the token cost. Despite this increase in token usage, the best pass rate remains 50.0 for these three models, indicating that neither an advanced agent setup nor a greatly enlarged iteration budget is sufficient to solve vibe protein design.

What separates a stronger agent from a weaker one. To identify what distinguishes agents that succeed under tool use from those that do not, we use GPT-5.6-Sol (OpenAI, 2026c) to characterize representative failures in the ReAct trajectories of DeepSeek-V4-Pro (generation pass rate 23.8) and Qwen3.5-397B-A17B (6.2). Table 7 shows that the gap is one of verification rather than tool choice. The two agents share three of their five most-used tools, and the tools that distinguish them divide along functional lines: DeepSeek-V4-Pro additionally relies on `protein_qc` and `sequence_analysis`, which inspect intermediate outputs, whereas Qwen3.5-397B-A17B relies on `inverse_folding` and `rfdiffusion3`, which produce them. Moreover, Qwen3.5-397B-A17B finishes without obtaining the requested site-contact evidence in 93.3% of its generation queries against 10.0% for DeepSeek-V4-Pro, and submits a final sequence without directly validating it in 48.1% against 20.0%. It also loses the design object outright, inverse-folding target and binder together to form the answer in 31.1% of queries where DeepSeek-V4-Pro never does so, and ends without an evaluable sequence in 36.7% against 5.2%.

The examples in Figure 5 illustrate these differences. On 7M59, DeepSeek-V4-Pro restricts inverse folding to the 65-residue binder, verifies contacts with 9 of 10 requested residues, validates the final sequence, and submits a binder that passes the benchmark. Qwen instead inverse-folds the 60-residue binder and 65-residue target together, then submits the complete 125-residue output as the binder. Although all of its tool calls succeed, it neither checks the requested contacts nor validates

DeepSeek-V4-Pro / 7M59 • Pass

Task: Design a binder contacting at least half of the specified target-site residues: I32, Q5, Y38, R12, T31, D7, V3, M20, Q16, and I4.

```
[2] rfdiffusion(
  target_chain="A",
  target_hotspots=[32,5,38,...,4], ...)
→ hotspot-conditioned complex backbone
[3] inverse_folding(
  chains=["B"],
  pdb_path=..., num_designs=16, ...)
→ 65-aa binder sequence:  MVV...GTR
[9] esmfold2(
  target_sequence=...,
  binder_sequence=MVV...GTR, ...)
→ pLDDT=80.42, pTM=0.755, iPTM=0.704
[10] structure_analysis(
  interface_chain_ids=["A","B"],
  positions=[3,4,5,...,38], ...)
→ site_contacts.requested_position_count=10
→ site_contacts.contact_count=9
```

Reason: …used hotspot-guided binder design …inverse-folded only the binder chain …the submitted sequence directly contacts 9 of the 10 site residues …
Answer: MVVVELVL…APPYDGTR

Qwen3.5-397B-A17B / 7M59 • Fail

Task: Design a binder contacting at least half of the specified target-site residues: I32, Q5, Y38, R12, T31, D7, V3, M20, Q16, and I4.

```
[2] rfdiffusion3(binder_length=60,
  target_hotspots=[32,5,38,...,4],
  target_pdb=..., ...)
→ complex backbone:  A=60 aa binder, B=65
aa target
[3] inverse_folding(chains=None,
pdb_path=...,  num_designs=8)
```

[No binder-chain restriction is supplied.]

```
→ 125-aa whole-complex sequence
```

Reason: …designed sequences for the binder backbone …The top-ranked sequence was selected.
Answer: AATPEEIA…LPDYPGDR
[Answer is the entire A+B output from call 3.]

Figure 5: Matched trajectories for the same binding-site-conditioned generation query.

Table 8: Non-default tool arguments requested by tool-augmented agents. Share is the percentage of all tool-augmented trajectories in which the pattern occurs.

Arguments	Examples	Share
Fewer diffusion steps	400 / 200 / 500 → 20–100 (PXDesign, RFdiffusion3, BoltzGen)	24.1
Lower retrieval cap	`max_sequences` 100 → 1–10; `max_downstream_candidates` 10 → 1	12.5
More designs per call	`num_samples` 5 → 16–512 (PXDesign)	10.3
Fewer designs per call	8 / 10 / 8 / 10 → 1–2 (MPNN, RFdiffusion, RFdiffusion3, BoltzGen)	10.2
Higher sampling temperature	0.1 → 0.5–1.0 (MPNN); 1.0 → 1.2 (PXDesign diffusion)	8.8
Higher retrieval cap	`max_sequences` 100 → 500–2,000; `max_downstream_candidates` 10 → 50–500	6.0

the submitted sequence. This suggests that what sets a stronger agent apart is not its choice of tools but its ability to preserve the design object and verify task-specific requirements before submission.

Tool argument selection. We characterize how agents configure the tools they call by categorizing, with GPT-5.6 Sol, every request that departs from a tool's default. As shown in Table 8, agents most often reduce the parameters that govern a tool's compute budget, suggesting that they optimize for speed rather than design quality. Cutting the number of diffusion steps is the most frequent override, appearing in 24.1% of trajectories: agents request 20–100 steps where PXDesign, RFdiffusion3, and BoltzGen default to 400, 200, and 500. At the same time, they pursue sequence diversity through the settings, raising the sampling temperature in 8.8% of trajectories. Search breadth shows a similar bias toward narrowing, as the cap on retrieved candidates is lowered about twice as often as it is raised (12.5% versus 6.0%). Overall, these tendencies indicate that agents consistently trade design quality for speed and diversity.

Continuity between the tool-free and tool-augmented settings. Here, we show that evaluating LLMs both with and without tools as done in our evaluations is necessary, as a model's internal design knowledge shapes its behavior once tools are available. To this end, we compare the tool-free and ReAct trajectories of one model on a single GO-conditioned generation query. Without tools, on potassium channel activity (GO:0005267), DeepSeek-V4-Pro names the two-transmembrane-helix pore architecture and the TVGYG selectivity filter, and attributes energy-independent conduction to

DeepSeek-V4-Pro / GO:0005267 • Tool-free

Task: Design a protein sequence satisfying the molecular function *potassium channel activity*: enables the energy-independent facilitated diffusion of a potassium ion through a transmembrane aqueous pore or channel.

Reason: . . . the characteristic K+ channel fold with two transmembrane helices (M1 and M2) flanking a re-entrant pore loop . . . The pore loop contains a short pore helix (ALWSTVEA) and the highly conserved selectivity filter sequence TVGYG, whose backbone carbonyls coordinate dehydrated K+ ions, enabling energy-independent diffusion. The sequence is designed to tetramerize, forming a central ion conduction pore.

Answer: MLVGLV. . . LWSTEATVGYGDLYPVT. . . VGREQ

DeepSeek-V4-Pro / GO:0005267 • ReAct

Hidden reasoning: A minimal K+ channel could be designed based on the KcsA-like architecture: two transmembrane helices (TM1, TM2), a pore helix (P), and the selectivity filter with the signature GYG sequence. . . . I could use search_sequence to retrieve real potassium channel sequences and use family_profile to build a profile, then use that to guide a de novo design.

```
[1] search_sequence(go_term="GO:0005267", reviewed=true, max_downstream_candidates=40)
→ 21 candidates with full sequences
[2] family_profile(candidates=..., align=true, min_conservation=0.75, min_coverage=0.5,
...)
→ MAFFT alignment of 21 reference sequences
→ consensus motif:  TIGYGD
→ conserved positions (≥ 75%; alignment columns):  T(864), G(866), G(868), D(869)
```

Hidden reasoning: . . . The conserved positions listed: T(864), G(866), G(868), D(869). The classic selectivity filter is TVGYG. . . . KcsA sequence (for reference, I know it roughly): . . . I shouldn't just copy KcsA (that's not de novo). However, the selectivity filter TVGYG must be present . . .

Reason: Potassium channels share a minimal architecture: an outer transmembrane helix (M1), a turret, a short pore helix, the signature TVGYG selectivity filter . . . and an inner transmembrane helix (M2). I authored a de novo sequence that assembles this KcsA-like topology, embedding the conserved TVGYG selectivity filter and placing hydrophobic membrane-spanning helices around it.
Answer: MALLGL. . . TVGYGDLYPVT. . . WFVGR

Figure 6: The same model on the same GO-conditioned generation query under tool-free (top) and ReAct-based tool-augmented (bottom) settings. Highlighted text marks the mechanistic commitment that recurs across both settings, the TVGYG selectivity filter and the two-transmembrane-helix pore architecture.

backbone carbonyls coordinating dehydrated K^+ ions. Given tools on the identical prompt, the same commitments determine its actions: it retrieves reviewed sequences for the corresponding GO term, builds an alignment-based family profile, and combines the resulting conservation evidence with its prior knowledge of the TVGYG filter and KcsA-like architecture to construct a candidate sequence (Figure 6). Both the translation of the semantic objective into a tool query and the interpretation of the returned evidence draw on what the model already knows about the target function. This case shows that competence observed without tools strongly shapes agent behavior once tools are available. Evaluating both settings therefore allows us to assess intrinsic protein reasoning alongside its translation into agentic protein design.

Results for antibody, DNA, and RNA binder generation. To cover binder design against a wider range of target types, we extend VIBEPROTEINBENCH with antibody, RNA, and DNA binder queries. These classes could not be included under our original temporal cutoff criterion, as both a SAbDab-based search and an independent RCSB–UniProt search identified no eligible antigen whose UniProt accession was first released on or after September 1, 2025, while an exhaustive search over structurally eligible nucleic-acid targets identified no corresponding RNA or DNA target independently registered after the cutoff in RNAcentral (RNAcentral Consortium, 2021) or GenBank (Benson et al., 2012).

We therefore report an auxiliary set comprising 30 antibody, 30 RNA-binder, and 30 DNA-binder queries under a structure-based temporal guard. Each query is derived from an X-ray diffraction or electron-microscopy complex first released by RCSB on or after September 1, 2025, with resolution ≤ 3 Å, and its prompt-visible target sequence or binding-region strand has no gapless, fully identical match in a pre-cutoff RCSB polymer entity. Antibody queries request a CDR-H3 sequence given the fixed heavy- and light-chain context and the antigen, whereas nucleic-acid queries provide the experimentally contacted single- or double-stranded target regions in the $5'$-to-$3'$ direction.

In this experiment, sequence validity requires a non-empty sequence composed exclusively of the 20 canonical amino acids. For RNA- and DNA-binder queries, outputs composed solely of the corresponding nucleotide alphabet, $\{A, C, G, U\}$ or $\{A, C, G, T\}$, are rejected; no sequence-length threshold is imposed. Fold validity requires the global complex pLDDT to exceed 70, without an

Table 9: Additional results on antibody, RNA, and DNA binder generation in no-tool setting.

Models	Antibody CDR-H3			RNA binder			DNA binder			Overall
	Seq. valid.	Fold valid.	Func. Cons.	Seq. valid.	Fold valid.	Func. Cons.	Seq. valid.	Fold valid.	Func. Cons.	
General-purpose LLMs										
GPT-5.4	100.0	96.7	3.3	96.7	3.3	0.0	80.0	73.3	46.7	<u>16.7</u>
Gemini3.1-Pro	100.0	100.0	0.0	96.7	60.0	3.3	96.7	96.7	66.7	**23.3**
Opus-4.6	46.7	43.3	0.0	73.3	16.7	0.0	100.0	70.0	33.3	11.1
DeepSeek-V4-Pro	100.0	96.7	0.0	100.0	36.7	10.0	100.0	70.0	33.3	14.4
DeepSeek-V3.2	100.0	100.0	10.0	100.0	50.0	3.3	100.0	33.3	6.7	6.7
Kimi-K2.5	100.0	96.7	0.0	83.3	20.0	0.0	93.3	70.0	23.3	7.8
Qwen3.5-397B-A17B	100.0	96.7	0.0	96.7	50.0	0.0	100.0	76.7	10.0	3.3
Qwen3.5-9B	60.0	56.7	3.3	73.3	56.7	0.0	60.0	56.7	0.0	1.1
Domain-specialized LLMs										
NatureLM-8×7B	100.0	96.7	0.0	70.0	33.3	0.0	73.3	20.0	0.0	0.0
SciReasoner-8B	100.0	93.3	0.0	100.0	33.3	0.0	100.0	43.3	3.3	1.1
TxGemma-9B	100.0	96.7	3.3	100.0	10.0	0.0	100.0	46.7	0.0	1.1
TxGemma-27B	100.0	96.7	0.0	100.0	33.3	0.0	100.0	33.3	0.0	0.0

Table 10: Cutoff ablation for binder generation in the no-tool setting, comparing binding targets first registered in UniProt (proteins) or PubChem (small molecules) before September 1, 2021 (pre-cutoff) and after September 1, 2025 (post-cutoff).

Models	Pre-cutoff				Post-cutoff			
	Seq. valid.	Fold valid.	Func. Cons.	Pass rate	Seq. valid.	Fold valid.	Func. Cons.	Pass rate
DeepSeek-V4-Pro	100.0	84.2	17.5	17.5	99.2	86.7	6.7	6.7
DeepSeek-V3.2	100.0	81.7	7.5	7.5	99.2	90.0	1.7	1.7
Qwen3.5-397B-A17B	100.0	83.3	2.5	2.5	99.2	82.5	2.5	2.5

additional pTM criterion. Functional consistency is evaluated using Protenix-v1. For RNA and DNA targets, it requires the maximum generated-protein–target-strand chain-pair iPTM to exceed 0.8. For antibody targets, we compute a regional iPTM from the Protenix-v1 PAE-bin probabilities after restricting the evaluated tokens to the designed CDR-H3 and the antigen, and require this score to exceed 0.8. This regional criterion prevents confidence attributable primarily to the fixed antibody framework from satisfying the functional-consistency criterion. A query passes only when sequence validity, fold validity, and functional consistency are all satisfied.

As shown in Table 9, these targets remain challenging for current LLMs, where the best overall pass rate is 23.3 from Gemini3.1-Pro. As in Table 16, functional consistency is the primary factor limiting performance. On antibody CDR-H3, ten of the twelve models attain 93.3–100.0 fold validity yet fall to 0.0–10.0 functional consistency, and on RNA binder generation eight models attain 33.3–60.0 fold validity while none exceeds 10.0. Domain-specialized models fail almost entirely, with NatureLM-8×7B and TxGemma-27B scoring 0.0 functional consistency on all three targets despite sequence validity of 70.0–100.0. These results indicate that the functional-grounding bottleneck identified on protein binder generation persists regardless of binding targets, and that the protein data in the pretraining mixture of these domain-specialized LLMs does not carry over to interface-level quality.

Effect of the temporal cutoff. To test whether prior exposure to a target inflates design performance, we compare binder generation on targets first registered in UniProt or PubChem before September 1, 2021 (pre-cutoff) against the post-cutoff targets used in VIBEPROTEINBENCH. As shown in Table 10, all three models attain higher or equal pass rates on pre-cutoff targets. The gap is most pronounced in functional consistency, which falls from 17.5 to 6.7 and 7.5 to 1.7 for DeepSeek-V4-Pro and DeepSeek-V3.2, respectively. This may suggest that prior exposure to a target could confer some advantage for binder design, though the effect is model-dependent.

Subtask-wise results for recognition stage. The fine-grained performance across recognition subtasks is detailed in Tables 11 to 13 for sequence-level, structure-level, and domain/function recognition, respectively. The results reveal a clear hierarchy where most baselines excel at sequence-level tasks but struggle with higher-order structural and functional reasoning. Specifically, the

Table 11: Pass rates for each sequence-level recognition subtask in tool-free setting.

Models	Motif detection	Charge state classification	Hydropathy classification	Aromaticity prediction	Residue charge identification	Residue hydropathy identification	Overall
General-purpose LLMs							
GPT-5.4	77.0	89.5	94.7	100.0	96.5	98.2	**91.9**
Gemini3.1-Pro	56.8	28.1	5.3	28.1	91.2	94.7	51.0
Opus-4.6	75.7	70.2	89.5	94.7	93.0	91.2	85.2
DeepSeek-V4-Pro	90.5	80.7	77.2	86.0	94.7	98.2	<u>88.0</u>
DeepSeek-V3.2	78.4	70.2	84.2	93.0	89.5	91.2	84.1
Kimi-K2.5	52.7	33.3	47.4	71.9	86.0	91.2	63.2
Qwen3.5-397B-A17B	85.1	77.2	80.7	94.7	87.7	96.5	86.9
Qwen3.5-9B	41.9	17.5	29.8	35.1	68.4	86.0	46.2
Specialized LLMs							
NatureLM-8×7B	0.0	56.1	14.0	0.0	78.9	45.6	30.9
SciReasoner-8B	27.0	21.1	50.9	8.8	47.4	87.7	39.8
TxGemma-9B	21.6	61.4	36.8	47.4	100.0	70.2	54.6
TxGemma-27B	16.2	31.6	7.0	12.3	87.7	94.7	40.4

Table 12: Pass rates for each structure-level recognition subtask in tool-free setting.

Models	Secondary structure identification	Burial classification	Disulfide bond identification	Overall
General-purpose LLMs				
GPT-5.4	44.0	82.4	40.0	**67.9**
Gemini3.1-Pro	32.0	80.4	40.0	63.0
Opus-4.6	44.0	80.4	60.0	**67.9**
DeepSeek-V4-Pro	20.0	66.7	40.0	50.6
DeepSeek-V3.2	28.0	80.4	60.0	63.0
Kimi-K2.5	28.0	80.4	20.0	60.5
Qwen3.5-397B-A17B	32.0	80.4	60.0	<u>64.2</u>
Qwen3.5-9B	12.0	60.8	40.0	44.4
Specialized LLMs				
NatureLM-8×7B	12.0	17.6	80.0	19.8
SciReasoner-8B	48.0	37.3	40.0	40.7
TxGemma-9B	36.0	41.2	60.0	40.7
TxGemma-27B	8.0	23.5	60.0	21.0

strongest models reach 84.1–91.9 at the sequence level but at most 67.9 at the structure level and 71.1 on domain and function recognition. Secondary structure identification is the hardest subtask throughout, never exceeding 48.0 across all twelve models, and only GPT-5.4 and Gemini3.1-Pro exceed 70.0 on both family and superfamily classification. We also observe significant within-model variance, particularly in specialized architectures. For instance, TxGemma-9B attains a perfect pass rate on cellular component prediction yet 0.0 on molecular function, and NatureLM-8×7B scores 0.0 on both motif detection and aromaticity prediction while achieving the highest disulfide bond identification rate of any model (80.0). These disparities suggest that specialized pretraining does not transfer uniformly to language-interfaced recognition.

Subtask-wise results for engineering stage. We decompose engineering stage into its subtasks and report their performance in Table 14. According to the table, engineering performance is bottlenecked by rationale alignment and in silico validity rather than hard constraint compliance. The dissociation is most pronounced for specialized models: NatureLM-8×7B and SciReasoner-8B satisfy hard constraints in 60.0–95.0% of queries across all three subtasks, yet score zero on rationale alignment in two of the three subtasks each, namely solubility and activity for NatureLM-8×7B and solubility and stability for SciReasoner-8B. General-purpose models instead degrade gradually, losing performance at each successive criterion rather than failing at a single one. Across both groups, activity is the hardest subtask, with in silico validity never exceeding 35.0 for any model.

Subtask-wise results for generation stage. We provide a detailed breakdown of generation performance across GO term-conditioned subtasks in Table 15 and binder design subtasks in Table 16. Across both generation tracks, performance is bottlenecked by functional grounding rather than by global sequence plausibility. Sequence validity and novelty remain high for nearly every model, yet GO match never exceeds 53.3 in GO-conditioned generation and interface validity never exceeds

Table 13: Pass rates for each domain/function recognition subtask across general-purpose and domain-specialized LLMs in tool-free setting.

Models	Family classification	Superfamily classification	GO term (MF)	GO term (BP)	GO term (CC)	Fold recognition	Overall
General-purpose LLMs							
GPT-5.4	77.3	76.6	72.7	71.4	66.7	50.0	**71.1**
Gemini3.1-Pro	71.2	79.7	36.4	42.9	66.7	27.8	62.6
Opus-4.6	57.6	73.4	54.5	42.9	33.3	47.2	59.9
DeepSeek-V4-Pro	63.6	68.8	36.4	42.9	0.0	36.1	56.7
DeepSeek-V3.2	54.5	53.1	36.4	42.9	33.3	33.3	48.1
Kimi-K2.5	56.1	59.4	18.2	42.9	0.0	27.8	48.1
Qwen3.5-397B-A17B	50.0	54.7	36.4	42.9	0.0	36.1	47.1
Qwen3.5-9B	31.8	32.8	27.3	28.6	0.0	19.4	28.9
Specialized LLMs							
NatureLM-8×7B	31.8	9.4	0.0	57.1	33.3	19.4	20.9
SciReasoner-8B	34.8	53.1	45.5	42.9	33.3	33.3	41.7
TxGemma-9B	7.6	14.1	0.0	57.1	100.0	16.7	14.4
TxGemma-27B	27.3	17.2	27.3	71.4	33.3	30.6	26.2

Table 14: Pass rates for each evaluation criterion within each engineering subtask across general-purpose and domain-specialized LLMs in tool-free setting.

Models	Solubility			Stability			Activity			Overall
	Hard const.	Rat. align.	In silico valid.	Hard const.	Rat. align.	In silico valid.	Hard const.	Rat. align.	In silico valid.	
General-purpose LLMs										
GPT-5.4	95.0	95.0	85.0	80.0	75.0	55.0	85.0	60.0	30.0	**56.7**
Gemini3.1-Pro	60.0	50.0	50.0	70.0	70.0	50.0	60.0	35.0	15.0	38.3
Opus-4.6	0.0	0.0	0.0	15.0	15.0	10.0	0.0	0.0	0.0	3.3
DeepSeek-V4-Pro	65.0	50.0	50.0	60.0	55.0	40.0	75.0	40.0	30.0	40.0
DeepSeek-V3.2	40.0	40.0	40.0	80.0	65.0	45.0	75.0	60.0	35.0	40.0
Kimi-K2.5	5.0	0.0	0.0	10.0	10.0	5.0	5.0	0.0	0.0	1.7
Qwen3.5-397B-A17B	60.0	60.0	55.0	65.0	65.0	50.0	65.0	50.0	30.0	45.0
Qwen3.5-9B	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Specialized LLMs										
NatureLM-8×7B	70.0	0.0	0.0	90.0	10.0	10.0	60.0	0.0	0.0	3.3
SciReasoner-8B	90.0	0.0	0.0	90.0	0.0	0.0	95.0	15.0	10.0	3.3
TxGemma-9B	5.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0
TxGemma-27B	15.0	0.0	0.0	5.0	0.0	0.0	20.0	0.0	0.0	0.0

26.7 in binder generation. The two bottlenecks respond differently to additional conditions, where adding a second GO term leaves GO match essentially unchanged and in some models slightly higher, whereas imposing a miniprotein length or binding-site constraint drives interface validity to 0.0 for nine and ten of the twelve models, respectively. The difficulty therefore lies in grounding function for GO-conditioned generation, and in meeting explicitly stated geometric specifications for binder generation.

Reliability of in silico evaluators. We select each in silico evaluator by calibrating candidate configurations against experimental measurements, drawing on thresholds and predictors established in prior protein-design studies (Table 17).

For binder screening, the $\mathrm{ipTM} > 0.8$ threshold has been adopted by PXDesign alongside pTM and structural-consistency filters (Team et al., 2025). Additionally, on 200 experimentally characterized binder–target pairs spanning four targets (Bennett et al., 2023), we compare this cutoff against $\mathrm{ipTM} > 0.5$ and the published cutoffs for pDockQ (Burke et al., 2023) and Rosetta shape complementarity (Lawrence & Colman, 1993; Pacesa et al., 2024). As shown, $\mathrm{ipTM} > 0.8$ threshold attains the highest precision, specificity, and balanced accuracy (BA; 80.0%) while retaining 94.1% recall.

For stability, the MGnify absolute stability predictor is the most reliable of the three estimators we examined. We adopt its released augmented SaProtΔG ensemble (Cho et al., 2026) and evaluate 200 WT–single-mutant pairs from its official homology-filtered MGnify test split, further excluding

Table 15: Pass rates for each evaluation criterion within each GO-conditioned generation subtask across general-purpose and domain-specialized LLMs in tool-free setting.

Models	GO-MF				GO-MF & GO-CC				GO-MF & GO-BP				Overall
	Seq. valid.	Fold valid.	GO match	Novelty	Seq. valid.	Fold valid.	GO match	Novelty	Seq. valid.	Fold valid.	GO match	Novelty	
General-purpose LLMs													
GPT-5.4	100.0	23.3	33.3	80.7	100.0	13.3	30.0	84.4	100.0	16.7	43.3	88.9	10.0
Gemini3.1-Pro	96.7	80.0	33.3	55.6	96.7	33.3	30.0	59.1	100.0	43.3	33.3	73.1	24.4
Opus-4.6	100.0	26.7	26.7	77.4	100.0	13.3	30.0	78.3	100.0	0.0	33.3	91.0	6.7
DeepSeek-V4-Pro	100.0	40.0	16.7	82.9	100.0	30.0	36.7	80.6	100.0	43.3	43.3	86.9	17.8
DeepSeek-V3.2	100.0	30.0	20.0	86.6	96.7	13.3	23.3	87.7	96.7	26.7	30.0	92.2	11.1
Kimi-K2.5	100.0	13.3	13.3	85.8	100.0	10.0	30.0	90.1	100.0	10.0	23.3	93.9	10.0
Qwen3.5-397B-A17B	100.0	30.0	23.3	84.3	100.0	16.7	23.3	86.9	100.0	20.0	30.0	92.6	10.0
Qwen3.5-9B	100.0	6.7	0.0	97.5	100.0	20.0	3.3	98.1	100.0	0.0	10.0	98.5	0.0
Specialized LLMs													
NatureLM-8×7B	96.7	50.0	30.0	88.8	90.0	46.7	33.3	74.4	96.7	40.0	26.7	94.1	<u>27.8</u>
SciReasoner-8B	100.0	60.0	53.3	86.3	100.0	46.7	43.3	88.1	96.7	23.3	33.3	90.4	**30.0**
TxGemma-9B	100.0	0.0	6.7	90.8	96.7	23.3	20.0	91.9	100.0	30.0	23.3	91.5	5.6
TxGemma-27B	80.0	6.7	3.3	90.8	86.7	6.7	13.3	92.0	93.3	6.7	20.0	91.7	2.2

Table 16: Pass rates for each evaluation criterion within each binder generation subtask across general-purpose and domain-specialized LLMs in tool-free setting.

Models	Protein target			Small-molecule target			Protein target & miniprotein			Protein target & binding site			Overall
	Seq. valid.	Fold valid.	Interface valid.	Seq. valid.	Fold valid.	Interface valid.	Seq. valid.	Fold valid.	Interface valid.	Seq. valid.	Fold valid.	Interface valid.	
General-purpose LLMs													
GPT-5.4	90.0	80.0	3.3	96.7	73.3	20.0	83.3	86.7	0.0	100.0	90.0	0.0	5.8
Gemini3.1-Pro	100.0	86.7	3.3	100.0	90.0	26.7	100.0	93.3	0.0	100.0	96.7	0.0	**7.5**
Opus-4.6	100.0	53.3	0.0	100.0	16.7	3.3	96.7	90.0	0.0	100.0	70.0	0.0	0.0
DeepSeek-V4-Pro	100.0	83.3	6.7	100.0	76.7	20.0	96.7	93.3	0.0	100.0	93.3	0.0	<u>6.7</u>
DeepSeek-V3.2	100.0	90.0	0.0	100.0	86.7	3.3	96.7	90.0	3.3	100.0	93.3	0.0	1.7
Kimi-K2.5	100.0	53.3	3.3	100.0	46.7	3.3	66.7	83.3	0.0	96.7	56.7	0.0	0.8
Qwen3.5-397B-A17B	100.0	83.3	0.0	100.0	66.7	10.0	96.7	93.3	0.0	100.0	86.7	0.0	2.5
Qwen3.5-9B	86.7	70.0	6.7	96.7	80.0	0.0	16.7	56.7	6.7	80.0	70.0	16.7	5.0
Specialized LLMs													
NatureLM-8×7B	100.0	70.0	16.7	100.0	16.7	0.0	93.3	93.3	0.0	100.0	70.0	10.0	<u>6.7</u>
SciReasoner-8B	96.7	36.7	0.0	96.7	16.7	3.3	100.0	86.7	3.3	100.0	53.3	0.0	1.7
TxGemma-9B	100.0	83.3	0.0	100.0	36.7	10.0	100.0	86.7	0.0	100.0	83.3	0.0	2.5
TxGemma-27B	100.0	70.0	0.0	100.0	3.3	0.0	96.7	90.0	0.0	96.7	76.7	0.0	0.0

homologs of ThermoMPNN training proteins by 30% sequence-identity screening and JackHMMER. Given independently predicted Protenix-v1 WT and mutant structures with MSAs enabled, it classifies the direction of stability change with 74.5% accuracy and 72.5% BA, ahead of ThermoMPNN and of unrelaxed PyRosetta energy differences.

For ligand affinity, the native Boltz-2 structure-and-affinity pipeline is the strongest configuration we tested. We predict WT–ligand and mutant–ligand complexes independently with MSAs enabled (Passaro et al., 2025) on 181 paired Platinum records (Pires et al., 2015). This recovers the direction of affinity change with 66.9% accuracy and 61.5% BA, against 60.8%/58.3% for Vina scored on the same Protenix-v1 structures.

For GO-conditioned generation, embedding-based reference matching gives the best balance between accepting annotated conditions and rejecting mismatched ones. We compare DeepGO-SE's ontology-aware sequence classifier (Kulmanov et al., 2024), GO-GPT's autoregressive GO-term predictor (Fallahpour et al., 2026), and our *ESM-C GO-reference matching* rule, which accepts a sequence when its minimum cosine similarity to ten proteins sharing all requested GO labels is at least the minimum similarity between 10 distinct reference proteins; MF+BP and MF+CC additionally require both labels to be recovered for the same protein. This choice of representation is supported by prior work showing that ESM-C embeddings track experimentally determined functional equivalence more faithfully than sequence identity (Shaw et al., 2025). Measuring BA as the average of annotated-condition acceptance and mismatched-condition rejection, the resulting rule attains the highest recall (73.3%) and BA (66.5%); the classifier baselines are considerably more precise (93.8% for DeepGO-SE) but recover only 27.6% and 14.9% of annotated conditions, which would leave most functionally valid designs unscored.

Cross-subtask correlations. To examine our coherence claim more broadly than the three representative pairs reported in the main paper, we measure pairwise Spearman correlations

Table 17: Calibration of in silico evaluators against experimental binding and stability measurements and annotated GO conditions. All values are percentages. Bold denotes the best value within each panel. Accuracy denotes the accuracy in predicting property direction relative to the wild-type (WT). Balanced accuracy (BA), shaded, averages the two class-specific recalls.

A Binder screening	Precision	Recall	Specificity	BA
Protenix ipTM > 0.5	46.1	**95.6**	42.4	69.0
Protenix ipTM > 0.8	**58.7**	94.1	**65.9**	**80.0**
pDockQ > 0.23	57.0	**95.6**	62.9	79.2
Rosetta $S_c \geq 0.55$	50.9	82.4	59.1	70.7
B Stability-change prediction			Accuracy	BA
PyRosetta $\Delta\Delta G$			56.0	55.8
ThermoMPNN			65.0	59.0
MGnify stability predictor			**74.5**	**72.5**
C Ligand-affinity change prediction			Accuracy	BA
Vina (Protenix-v1 WT structure + Rosetta relax)			56.4	49.8
Vina (Protenix-v1 WT/mutant structures)			60.8	58.3
Boltz-2 affinity (Protenix-v1 WT/mutant structures)			66.3	56.3
Boltz-2 affinity (Boltz-2 WT/mutant structures)			**66.9**	**61.5**
D GO-condition consistency	Precision	Recall	BA	
ESM-C GO-reference matching	64.5	**73.3**	**66.5**	
DeepGO-SE	**93.8**	27.6	62.9	
GO-GPT	89.2	14.9	56.6	

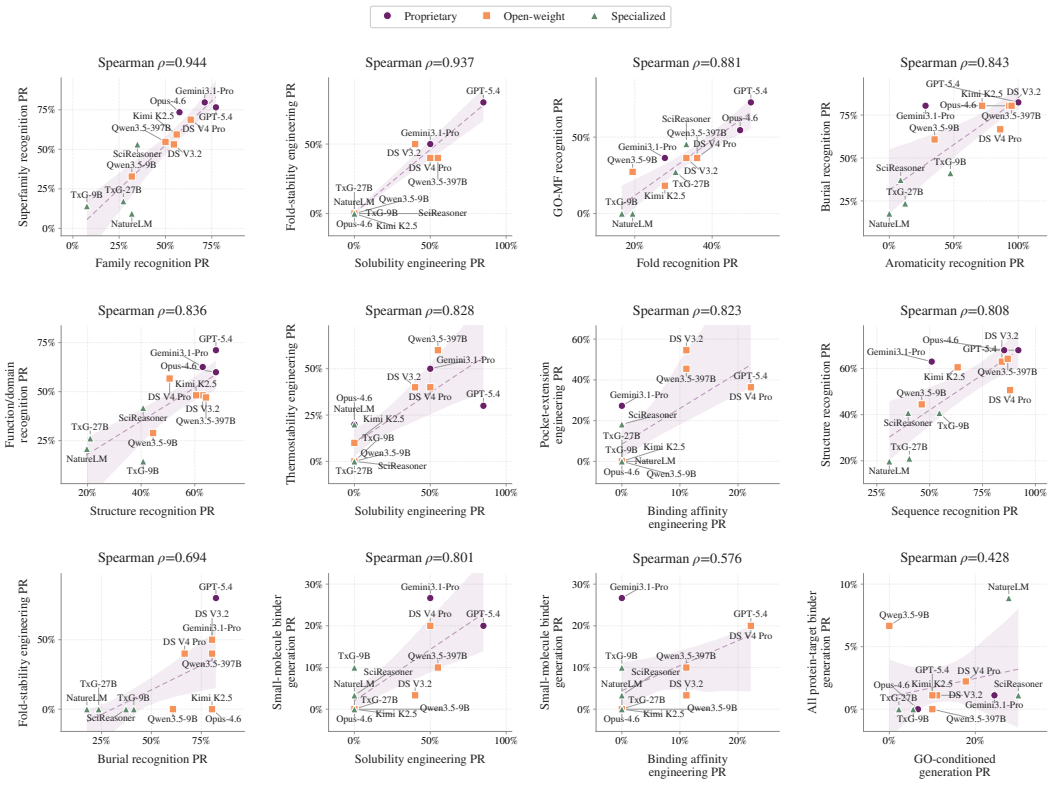

Figure 7: Correlations between different subtask performance of baseline LLMs. DS denotes DeepSeek and TxG indicates TxGemma-chat models.

across diverse mechanistically linked subtask pairs in VIBEPROTEINBENCH (Figure 7). The twelve pairs span recognition×recognition, engineering×engineering, recognition×engineering,

Table 18: Safety refusal rates (%) on our benchmark. Each entry gives the percentage of queries that returned no usable answer, with the raw count in parentheses. API-level safety blocks and explicit in-text refusals are counted together, and all such cases are scored as failures.

Models	Recognition	Engineering	Generation
Opus-4.6 (w/o tool)	0.0	35.0	0.0
GPT-5.4 (w/o tool)	0.0	3.3	7.1
GPT-5.4 (w/ tool; ReAct-based)	0.0	3.3	1.0

engineering×generation, and generation×generation combinations. All of them correlate positively and nine reach $\rho \geq 0.80$, with the strongest coupling found within stages: family× superfamily recognition ($\rho = 0.944$), solubility × fold-stability engineering ($\rho = 0.937$), fold × GO-MF recognition ($\rho = 0.881$), and binding-affinity × pocket-extension engineering ($\rho = 0.823$). The weakest pair is GO-conditioned × protein-target binder generation ($\rho = 0.428$), which is expected given that generation pass rates for protein binders are compressed near zero for most models and therefore admit little rank separation. The breadth of these positive correlations shows that the coupling reported in the main paper is not an artifact of the three pairs we chose to highlight, further supporting that recognition, engineering, and generation probe mutually dependent rather than independent competences, and are therefore evaluated together.

Safety refusals. A fraction of queries returned no usable answer because of safety filtering, and we score every such case as a failure (Table 18). We report the refusal rate per stage, counting API-level safety blocks together with explicit in-text refusals. Refusals are absent in recognition and concentrate in engineering for Opus-4.6, which declined 35.0% of its engineering queries.

C Protein Recognition Task Details

C.1 Sequence-level recognition

- **Motif detection:** This task probes whether the model can interpret biologically meaningful sequence patterns rather than recognizing individual residues in isolation. Each question presents a protein sequence with a PROSITE (Sigrist et al., 2013) sequence motif pattern and asks the model to determine whether the motif occurs in the sequence and, if so, to report all matching residue spans using one-indexed positions.
- **Physicochemical composition:** Given a protein sequence, the model predicts global biochemical properties of the entire sequence. We include isoelectric point (pI), hydropathy as measured by GRAVY score, and aromaticity, using sequence-property calculations implemented in BioPython's `ProteinAnalysis` module (Cock et al., 2009). We formulate pI and GRAVY as categorical recognition over predefined biochemical ranges, and query aromaticity as the numeric fraction of aromatic residues.
- **Residue-level property recognition:** To test residue-level understanding, we sample individual one-indexed residue positions and ask the model to classify their biochemical traits. Specifically, the model identifies net charge of each residue as positive, negative, or neutral, and its hydropathy as hydrophobic or hydrophilic.

C.2 Structure-level recognition

To construct queries in structure-level recognition task, we use experimentally determined 3D structures from reference sequences from the RCSB PDB database (Burley et al., 2021).

- **Secondary structure identification:** Each question asks the model to identify which candidate subsequence corresponds to a specified secondary-structure class, such as helix, sheet, or coil. We assign secondary-structure labels from the associated PDB structure using DSSP, and sample length-matched distractors from subsequences assigned to different structural classes.
- **Burial classification:** To evaluate structure-aware reasoning about solvent exposure, we ask the model to identify a subsequence that is either buried or exposed. We obtain residue-level burial labels from the DSSP-computed relative solvent accessibility (RSA), applying fixed thresholds to

distinguish buried and surface-exposed residues. We then sample length-matched distractors from regions of the opposite burial class.

- **Disulfide bond identification:** Each question asks the model to identify which residue pair forms a disulfide bond in the associated structure. Correct answers are Cys–Cys pairs confirmed by structural records or geometry, while distractors include both non-bonded Cys–Cys pairs and Cys–non-Cys pairs, to prevent the model from solving the task by cysteine frequency alone.

C.3 Domain/function recognition

- **Fold recognition:** Each question presents a reference protein sequence and asks the model to identify another sequence that shares the same topology group (T-group) in the ECOD database (Cheng et al., 2014), which classifies protein domains by their evolutionary relationships. To prevent the model from shortcutting through sequence homology, we constrain the correct answer to share the same T-group as the reference yet fall below 30% sequence identity.
- **Family/Superfamily classification:** Given a protein sequence, the model identifies the correct family and superfamily labels, which we curate from the InterPro database (Blum et al., 2021) – a resource that integrates diverse protein signatures into unified functional and evolutionary classifications. To construct hard negatives that are structurally similar yet functionally distinct, we draw distractors from the same superfamily as the reference for family classification, and from the same ECOD X-group (Cheng et al., 2014) for superfamily classification, a category that groups proteins with possible but inconclusive evolutionary relationships.
- **GO term identification:** To probe function-level recognition, we ask the model to identify the correct Gene Ontology (GO) (Gligorijević et al., 2021) annotations for a given protein sequence. For each reference protein, we select the top-3 leaf terms whose associated Swiss-Prot entries appear least frequently in the database, targeting specific rather than generic functional roles that would likely overlap with other proteins. We then draw distractors from other Swiss-Prot proteins that share the same superfamily as the reference but whose top-3 least frequent GO terms do not overlap with any of its curated terms.

D Rationale-guided Engineering Task Details

D.1 Expert-curated rationales

For each subtask, a domain expert encodes the design rationale as a structured pair of rules: a family of defective patterns that identifies residues M whose modification is expected to improve the target property, and a family of residues P that must be preserved to avoid disrupting function, fold, or oligomeric context. Every per-residue call is backed by a computationally grounded feature–including residue exposure, local structure, ligand geometry, residue chemistry, functional annotations, or group-wise evolutionary statistics–so that the rationale is inspectable rather than opaque. We provide the full rule sets below.

Solubility. The solubility rationale targets solvent-exposed chemical liabilities. A residue enters the defective pool if it is surface exposed, defined by relative SASA $\mathrm{RSA} > 0.25$, and is hydrophobic either by amino-acid class or Kyte–Doolittle hydropathy (> 0). These residues are assigned a charged-or-polar repair direction when homologous sequences support such substitutions. We also include exposed unpaired cysteines: a Cys with $\mathrm{RSA} > 0.25$ and no detected disulfide partner is treated as a liability and assigned a Cys→Ser repair direction.

The preserved pool contains residues whose mutation may compromise fold or function: disulfide-bonded cysteines, ligand/cofactor contacts, salt-bridge participants, oligomeric interface residues, annotated catalytic or active-site residues, annotated binding-site residues, buried hydrophobic core residues, highly conserved surface residues lacking polar/charged substitution support, and Pro/Gly residues with potential backbone-geometry roles. Structural contacts are computed from the input structure, while catalytic, active-site, binding-site, and disulfide annotations are also imported from UniProt when available.

Fold Stability. The fold-stability rationale targets local structural weaknesses with pattern-specific repair directions. We identify under-packed core sites as buried small hydrophobic residues ($\mathrm{RSA} <$

0.09) with low local C_β-neighbor packing density; these are assigned larger hydrophobic substitutions. Buried polar residues without a side-chain hydrogen-bond partner are assigned hydrophobic-isostere substitutions. Helix-related defects include Gly in helix interiors, suboptimal helix N-caps, and helix N-terminal dipole positions lacking acidic stabilization, with repairs such as Gly→Ala, N-cap-capable residues, or Asp/Glu. We also include loop positions with Pro-compatible ϕ angles, buried unpaired charged residues, and positions where the wild type is rare relative to a high-frequency consensus residue.

Protected residues for fold stability include catalytic or active-site residues, metal-coordinating residues, disulfide-bonded cysteines, ligand/cofactor contacts, oligomeric interface residues, buried highly conserved core residues, salt-bridge participants, Pro/Gly residues, and residues participating in satisfied side-chain hydrogen-bond networks in densely packed neighborhoods.

Thermostability. The thermostability rationale is based on mesophile–thermophile group contrasts. For each mesophilic target, homologs are partitioned into mesophilic and thermophilic groups using organism-level optimal growth temperature metadata. A residue is considered defective when the mesophilic wild-type amino acid is common in the mesophile group but depleted in the thermophile group, with a thermophile-enriched non-wild-type amino acid providing the suggested substitution. In the implemented rule, the mesophile wild-type frequency must be at least 0.4, the thermophile wild-type frequency at most 0.3, the mesophile–thermophile frequency gap at least 0.2, and the thermophile consensus frequency at least 0.2.

The preserved pool contains catalytic or active-site residues, metal-proximal residues, disulfide-bonded cysteines, ligand/cofactor-proximal residues, interface residues, residues conserved in both temperature groups, Pro/Gly residues, and salt-bridge participants when detected.

Activity–binding affinity. The binding-affinity rationale targets chemical-complementarity mismatches at the substrate-binding site. The pipeline selects a substrate ligand from the structure, classifies ligand atoms by local chemistry, and computes the closest substrate heavy atom for each protein residue. Residues are binned as contacting, near, or far using minimum heavy-atom distance thresholds of 4.5 Å and 6.5 Å. A contacting residue enters the defective pool when its residue class mismatches the nearest substrate atom chemistry: hydrophobic residues facing polar atoms, hydrophobic residues facing charged groups, polar residues facing nonpolar or aromatic atoms, or same-sign charge repulsion. The repair direction is the complementary residue class, such as polar, hydrophobic, or the opposite charge.

Protected residues include annotated catalytic or active-site residues, metal-coordinating residues, disulfide-bonded cysteines, already complementary substrate-contacting residues, highly conserved substrate-contacting residues, non-binding residues far from the substrate, interface residues, Pro/Gly residues near the binding site, and salt-bridge participants. If a residue is both a candidate mismatch and protected, the protection rule takes precedence.

Activity–pocket expansion. The pocket-expansion rationale targets steric restriction in the ligand-binding pocket. Pocket membership is defined directly from ligand geometry: residues with minimum heavy-atom distance to the ligand at most 6.0 Å are pocket-lining residues, residues within 8.0 Å are pocket-adjacent, and all others are non-pocket residues. Each residue is assigned a side-chain volume and size class; residues with volume below 120 Å^3 are small, 120–170 Å^3 are medium, and at least 170 Å^3 are large. A defective pocket-expansion target must be a medium or large pocket-lining residue with homologous support for smaller substitutions. Candidate substitutions are amino acids whose side-chain volume is at least 10 Å^3 smaller than the wild type.

The preserved pool includes catalytic residues, metal-coordinating residues, disulfide-bonded cysteines, essential substrate contacts, conserved pocket residues lacking smaller-residue support, non-pocket residues, interface residues, Pro/Gly residues in or near the pocket, and salt-bridge participants. Pocket expansion is therefore defined by ligand-distance bins and residue volume.

D.2 Answer evaluation

All proposed mutants are evaluated through a multi-facet pipeline that proceeds sequentially: hard constraints, then rationale alignment, and finally in silico validation. We adopt this dependent

structure because each preceding rubric establishes a precondition for the next to be meaningful. Mutating a protected residue, such as a catalytic residue or a binding-site contact, can disrupt the very function the engineering task aims to preserve, so a property gain achieved at such a cost cannot be credited as successful engineering. Likewise, even when the hard constraints are satisfied, a property gain that does not follow the expert-specified rationale is achieved by a mechanism unrelated to the intended design intent, and therefore does not reflect the rationale-guided competence the task is meant to probe. We accordingly skip downstream rubrics for mutants that fail an earlier one. A prediction is counted as correct only when it passes every applicable rubric.

Hard constraints. This rubric verifies the prerequisite conditions a mutant must satisfy to qualify for further in silico evaluation.

- Mutant sequence length must be identical to the WT sequence.
- Mutant sequence must not be identical to the WT sequence.
- Protected positions must remain unchanged.
- The number of mutations must not exceed the predefined maximum.
- Mutations are allowed only at specified target positions.
- Non-target and non-mutated positions must remain WT.

In silico validation. This rubric combines a common fold-quality gate with a task-specific property metric. The common pLDDT gate passes when $\text{pLDDT}_{\text{mutant}} \geq \text{pLDDT}_{\text{WT}}$ or $\text{pLDDT}_{\text{mutant}} > 70$.

- **Solubility:** The mutant must pass the pLDDT gate from Protenix-v1, decrease surface GRAVY, and increase the fraction of surface residues that are charged or polar. Surface residues are recomputed from predicted structures using RSA > 0.25.
- **Fold stability and thermostability:** The mutant must pass the pLDDT gate from Protenix-v1 and increase the unfolding free energy ΔG_{unfold} predicted from MGnify stability predictor. We define $\Delta\Delta G_{\text{unfold}} = \Delta G^{\text{mut}}_{\text{unfold}} - \Delta G^{\text{WT}}_{\text{unfold}}$; thus, $\Delta\Delta G_{\text{unfold}} > 0$ indicates increased stability.
- **Binding affinity:** The mutant must pass the pLDDT gate and have a lower $\log_{10}(\text{IC}_{50}/\mu\text{M})$ than WT for the same ligand from the Boltz-2 prediction. Thus, a lower value indicates stronger predicted binding.
- **Pocket expansion:** The mutant must pass the pLDDT gate from Protenix-v1 and increase or preserve the active-site pocket volume measured by pyKVFinder: $\text{V}^{\text{pk}}_{\text{mutant}} \geq \text{V}^{\text{pk}}_{\text{WT}}$. The active-site cavity is selected using ligand coordinates and projected pocket residues when available.

Rationale alignment. For queries that include defective targets M, we additionally assess whether each mutated target follows the intended repair rule.

- **Solubility:** Surface-hydrophobic targets must be changed to an allowed charged or polar residue, excluding cysteine. Free-surface cysteine targets receive credit for Cys→Ser.
- **Fold stability:** Mutations must satisfy the pattern-specific repair rule: larger hydrophobics for under-packed core sites, hydrophobic replacements for buried unsatisfied polar residues, helix-compatible substitutions for helix defects, Pro for Pro-compatible loops, neutral nonpolar replacements for buried unpaired charges, or the consensus/same-class residue for back-to-consensus targets.
- **Thermostability:** Mutations must be thermophile-directed: the mutant residue must match the suggested thermophile-enriched residue, belong to the acceptable thermophile-dominant amino-acid set, or satisfy an explicitly specified acceptable class.
- **Binding affinity:** Mutations must improve residue–substrate complementarity by matching the ideal residue character for the detected mismatch, such as polar, hydrophobic, or opposite charge.
- **Pocket expansion:** Pocket-lining targets must be replaced by an accepted smaller residue, or by any amino acid whose side-chain volume satisfies $\text{V}^{\text{sc}}_{\text{mutant}} \leq \text{V}^{\text{sc}}_{\text{WT}} - 10\,\text{Å}^3$.

Search terms for literature-based filtering. We first built task-specific literature pools from PubMed and Semantic Scholar using broad engineering queries, and then filtered out a candidate if

a returned title or abstract contained the protein name together with at least one design term and at least one task-context term.

- **Solubility and fold stability.** We used broad protein-engineering queries covering directed evolution, de novo/computational/rational design, mutation, variants, and engineering. Design terms included protein engineering/editing, computational or rational design, mutation, and solubility/stability/thermostability improvement; context terms included protein engineering, solubility, stability, and thermostability.
- **Activity.** We used enzyme-focused queries covering enzyme engineering, catalytic activity, mutation, k_{cat}, and directed evolution. Design terms included enzyme/protein editing or mutation, catalytic improvement, affinity enhancement, interaction engineering, and enzyme–substrate or enzyme–inhibitor binding; context terms included enzyme, protein engineering, and activity.
- **Thermostability.** We used temperature-focused queries covering thermostability improvement, thermal stability, thermophile/mesophile contrasts, protein mutation, and melting temperature. Design terms included mesophile/thermophile engineering or mutation, protein engineering/editing, computational or rational design, and thermostability improvement; context terms included thermostability, thermal stability, Tm, heat resistance, thermophile/mesophile, thermotolerance, thermal denaturation, thermolability, and optimal growth temperature.

E Generation Task Details

E.1 Answer evaluation

We evaluate generated protein sequences along several axes. We first check sequence validity, requiring a non-empty sequence composed only of canonical amino acid letters. For miniprotein-conditioned generation, the sequence must additionally fall within the 40–80 amino acid length range.

For coarse-grained functional generation, we evaluate structural plausibility and functional consistency. Structural plausibility is assessed from predicted structure confidence, requiring pLDDT > 70 and pTM > 0.5. Functional consistency is assessed with an ESM-C-based GO-reference matching rule (Candido et al., 2026).

For target-specific binder generation, we evaluate structural plausibility and target engagement. We require a valid sequence, pLDDT > 70, pTM > 0.5, and ipTM > 0.8 (Pacesa et al., 2024; Nori et al., 2025). Under the binding-site-conditioned setting, we further require the binder to contact at least half of the specified target-site residues within a 5 Å heavy atom distance cutoff.

We additionally report sequence novelty across both GO-conditioned and binder-design queries. For each scorable GO-conditioned output, novelty is defined as 100 times one minus its maximum ungapped sequence identity to experimentally annotated Swiss-Prot proteins satisfying all queried GO terms. For each scorable binder, novelty is defined as 100 times one minus its maximum aligned sequence identity among significant MMseqs2 hits (E-value at most 1) in UniRef50 release (2026-03); a completed search with no hit receives a score of 100. We report the query-weighted mean over all scorable GO and binder outputs, rather than an equal-weight average of the two cohort means.

The overall generation score is a strict pass: a query is counted as correct only when the sequence passes every required check for its subtask.

E.2 Source dataset curation and contamination mitigation

We curate three categories of source data for the generation stage, namely functional GO terms, protein targets, and small-molecule targets. The integrity of the binder track relies on a strict temporal holdout (2025-09-01) for curating protein binding targets from UniProt and small-molecule targets from PubChem entries. This ensures that the structural or chemical targets are "novel" to the model.

- **GO functional terms:** We curate GO terms in the way that balances functional rarity. Specifically, we compute the information content (IC) (Gligorijević et al., 2021) for each GO term combination and partition the candidate pool into equal-mass quantile bins. Taking MF–BP paired conditioning as an example, the IC of each pair $(\mathrm{MF}_i, \mathrm{BP}_j)$ with MF term i and BP term j is $\mathrm{IC}(\mathrm{MF}_i, \mathrm{BP}_j) =$

Table 19: Comparison of VIBEPROTEINBENCH with existing protein-design benchmarks. *Rec.*: recognition; *Eng.*: engineering; *Gen.*: generation. *Lang. interface*: natural-language queries vs. rigid input schemas. *Mech. rationale*: expose expert mechanistic rationales during sequence design. *Struct. quality*: folded-structure validation beyond sequence-level metrics. *Contam. control*: explicit mitigation via temporal holdout, structure/sequence split, or literature filtering.

Benchmark	Lang. interface	Rec. stage	Eng. stage	Gen. stage	Mech. rationale	Struct. quality	Contam. control
PDB-Struct (Wang et al., 2023)	✗	✗	✗	✓	✗	✓	✓
ProteinBench (Ye et al., 2024)	✗	✗	✗	✓	✗	✓	✓
ProteinLMBench (Shen et al., 2024)	✓	✓	✗	✗	✗	✗	✗
LiveProteinBench (Rong et al., 2025)	✓	✓	✗	✗	✗	✗	✓
PDFBench (Kuang et al., 2025)	✓	✗	✗	✓	✗	✓	✓
InstructProBench (Song et al., 2025)	✓	✗	✗	✓	✗	✓	✓
Mol-Instructions (Fang et al., 2023)	✓	✓	✗	✓	✗	✗	✗
SciReasoner (Wang et al., 2025c)	✓	✓	✗	✓	✗	✗	✗
VIBEPROTEINBENCH (Ours)	✓	✓	✓	✓	✓	✓	✓

$-\log_2 \frac{|\mathcal{P}(\mathrm{MF}_i)\cap\mathcal{P}(\mathrm{BP}_j)|}{|\mathcal{P}|}$, where $\mathcal{P}(\cdot)$ denotes the set of Swiss-Prot proteins annotated with given GO term and $\mathcal{P}$ is the full Swiss-Prot collection.

- **Protein targets:** We curate high-resolution (≤ 3.0 Å) dimeric structures from the RCSB PDB (Burley et al., 2021). Targets are restricted to those where at least one chain was registered after the September 1, 2025 cutoff, ensuring they were not present in the training sets of current LLMs. We exclude enzymes and DNA/RNA-binding complexes to maintain track-specific focus. For site-specific subtasks, we identify the top-10 residues on the target chain with the highest density of heavy-atom contacts (≤ 5 Å) to the partner chain.
- **Small-molecule targets:** Ligands are sourced from PubChem (Kim et al., 2023) using a creation-date cutoff of September 1, 2025. We apply a stringent artifact filter based on the Plinder badlist to remove buffers, reagents, and other non-biological molecules. SMILES strings are canonicalized using RDKit, and targets with extreme charges or long hydrocarbon linkers are discarded to ensure chemical relevance.

F EXTENDED COMPARISON WITH EXISTING BENCHMARKS

We expand on the comparison with existing language-interfaced protein design benchmarks. Below, we revisit each benchmark in detail along the three limitations: coverage of mechanistic protein design competencies, rigor of in silico validation, and breadth of design intents under evaluation. Then we also provide a comparison against language-interfaced benchmarks focusing on protein recognition or retrieval. We summarize the distinctions between VIBEPROTEINBENCH and prior benchmarks in Table 19.

Coverage of mechanistic protein design competencies. PDFBench (Kuang et al., 2025) and InstructProBench (Song et al., 2025) restrict evaluation to function-conditioned sequence generation, leaving protein understanding and rationale-guided engineering unexamined. Mol-Instructions (Fang et al., 2023) and the evaluation sets released with SciReasoner (Wang et al., 2025c) cover a wider set of biomolecular tasks, including protein property understanding and functional generation, but do not incorporate an engineering task. More importantly, these benchmarks do not probe whether LLMs can internalize and apply mechanistic logic as human practitioners do in real-world workflows. In practice, protein design is guided by mechanistic principles grounded in sequence–structure–function relationships, where practitioners examine structural and biochemical signals and apply targeted modifications accordingly. Consequently, they cover only a partial slice of the competencies involved in vibe protein design, and do not unify these competencies into a coherent evaluation of design as a workflow that connects understanding, rationale-guided modification, and generation.

Rigor of in silico validation. Many benchmarks rely on metrics that do not systematically verify the quality of designed proteins through in silico validation. SciReasoner (Wang et al., 2025c) adopts surface-level assessment such as sequence alignment and valid amino-acid composition.

Mol-Instructions (Fang et al., 2023) aligns each generated sequence against reference proteins in the corresponding functional regions, which conflates functional consistency with similarity to known proteins. These benchmarks do not comprehensively evaluate whether generated sequences are structurally plausible or functionally consistent at the level of folded protein outcomes.

Breadth of design intents. Finally, existing benchmarks predominantly express functional intent through semantic language-based descriptions such as Gene Ontology terms or textual annotations (Kuang et al., 2025; Fang et al., 2023; Wang et al., 2025c), and rarely supply novel constraints such as newly characterized binding partners. They also omit practical laboratory constraints, including miniprotein length regimes and binding-site specifications, that frequently arise in therapeutic and diagnostic design. Taken together, these gaps leave open whether current LLMs can perform protein design as a coherent workflow grounded in mechanistic reasoning, verifiable biological outcomes, and realistic design constraints.

Language-interfaced benchmark for protein understanding and retrieval. A complementary line of work evaluates protein QA, protein-text retrieval, and leakage-aware protein understanding benchmarks (Carrami & Sharifzadeh, 2024; Shen et al., 2024; Jararweh et al., 2025; Xu et al., 2023; Wu et al., 2025; Rong et al., 2025). These benchmarks assess whether models can answer questions about proteins or align protein sequences with textual descriptions. However, their scope is largely confined to understanding or retrieval, whereas VIBEPROTEINBENCH evaluates targeted sequence modification and novel protein generation under specified functional constraints.

G HUMAN EXPERT REVIEW

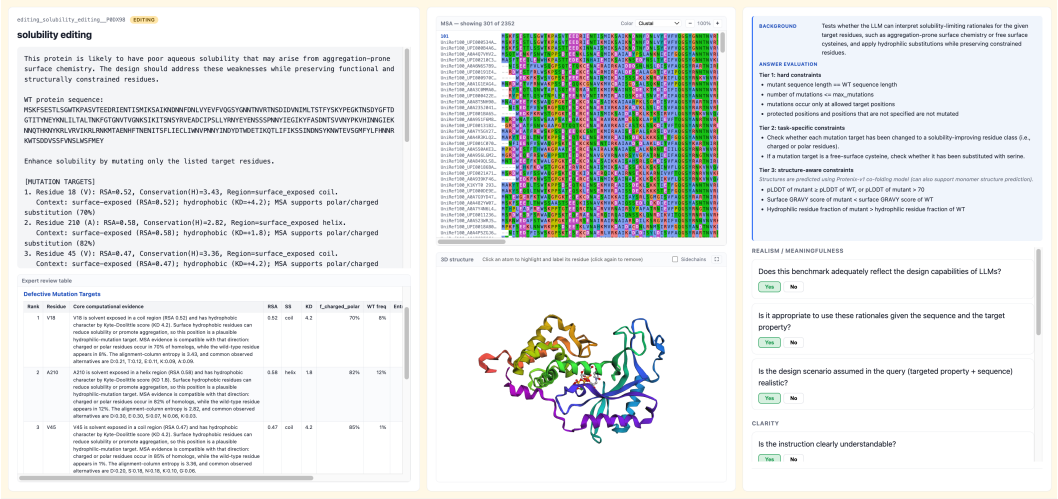

Figure 8: Snapshot of expert review interface for solubility engineering query.

We further validate the task construction and evaluation criteria of VIBEPROTEINBENCH through review by human domain experts. Each subtask is presented to the experts together with its background, including evolutionary and structural context, the computational tools used to construct it, and the answer evaluation method. The experts then score each query against three binary rubrics: realism and meaningfulness, clarity, and in silico verifiability. The snapshot of review interface and full rubric definitions are provided in Figure 8 and Table 20, respectively.

H DATASET STATISTICS

We report detailed benchmark statistics in Table 21. The engineering stage is smaller because its per-query content has to be validated, whereas recognition queries of the same type share one template. Although both recognition and engineering use established computational tools, the resulting features serve different purposes. In recognition, a computed property or curated annotation directly defines

Table 20: Human expert review rubrics for recognition and engineering queries. Each subtask and query is scored against three binary criteria: *realism/meaningfulness*, *clarity*, and *in silico verifiability*. The criterion is highlighted in **bold** and the guiding questions presented to the experts are listed below.

Human Expert Review Rubrics	
Recognition	**Engineering**
Realism/meaningfulness • Does this benchmark adequately reflect the design capabilities of LLMs? • Does possessing this knowledge meaningfully contribute to performing design tasks?	**Realism/meaningfulness** • Does this benchmark adequately reflect the design capabilities of LLMs? • Is it appropriate to use these rationales given the sequence and the target property? • Is the design scenario assumed in the query (targeted property and sequence) realistic?
Clarity • Is the instruction clearly understandable? • Can domain experts solve this task with their own knowledge or with computational tools?	**Clarity** • Is the instruction clearly understandable? • Can domain experts solve this task with their own knowledge or with computational tools?
In silico verifiability • Is the ground truth unambiguous given a well-defined computational method?	**In silico verifiability** • Does the evaluation method allow for reasonably verifying the answer to the query?

the reference answer under a shared query template, so expert review focuses on whether the task and its answer definition are appropriate. In engineering, these features are used to infer which residues should be modified or protected. The rationale defining them is authored by domain experts, but its instantiation on a given protein is automatic, computed from structural features, ligand geometry, and Henikoff-weighted MSA statistics. We therefore present each query to an expert together with its structure rendering, MSA, and per-residue features, and ask whether the selected residues actually satisfy the rationale (Appendix G). This check cannot be amortized across the benchmark, and a misassigned residue can change the evaluation result for that query. We therefore prioritize careful, expert-driven construction over sheer volume, as comparable expert-curated benchmarks do; for example, CMT-Benchmark (Pan et al., 2026) contains 50 condensed-matter-theory problems and InfiBench (Li et al., 2024) contains 234 coding questions.

Table 21: Dataset statistics for VIBEPROTEINBENCH.

Subtasks	# Queries
Recognition – Sequence	
Motif detection	74
Protein charge state classification	57
Protein hydropathy classification	57
Protein aromaticity prediction	57
Residue charge identification	57
Residue hydropathy identification	57
Subtotal	*359*
Recognition – Structure	
Secondary structure identification	25
Burial classification	51
Disulfide bond identification	5
Subtotal	*81*
Recognition – Function	
Family classification	66
Superfamily classification	64
GO term identification (MF)	11
GO term identification (BP)	7
GO term identification (CC)	3
Fold recognition	36
Subtotal	*187*
Recognition total	**627**

Subtask	# Queries
Engineering	
Solubility engineering	20
Stability (fold stability) engineering	10
Stability (thermostability) engineering	10
Activity (binding affinity) engineering	9
Activity (pocket extension) engineering	11
Subtotal	*60*
Engineering – constraint-free	
Solubility engineering	20
Stability (fold stability) engineering	10
Stability (thermostability) engineering	10
Activity (binding affinity) engineering	9
Activity (pocket extension) engineering	11
Subtotal	*60*
Engineering total	**120**
Generation – Coarse-grained	
GO-MF term conditioned	30
GO-MF & GO-CC term conditioned	30
GO-MF & GO-BP term conditioned	30
Subtotal	*90*
Generation – Target-specific binder	
Small-molecule target	30
Protein target	30
Protein target & miniprotein	30
Protein target & binding-site conditioned	30
Subtotal	*120*
Generation total	**210**
Total	**957**