CIDER: Conformal Information-Directed Agents for Low-Budget Protein Engineering

Anonymous Authors[1]

Abstract

Low-throughput protein engineering is a fixed-budget sequential decision problem: with only a small number of assay slots, the goal is top-tail discovery rather than global regression accuracy. We present CIDER-BENCH, a safety-filtered retrospective benchmark that converts ProteinGym deep-mutational-scanning assays into batched design-build-test-learn campaigns, and CIDER-AGENT, a constrained policy that combines conformal top-tail calibration, information-directed acquisition, and diversity-aware batch optimization. The language-model component is limited to bounded controller actions with auditable traces and cannot propose variants outside the candidate set. In 48-query campaigns over 20 benign landscapes, CIDER-AGENT improves rare-hit discovery over static PLM ranking, standard active-learning baselines, LLM-only planners, and a FolDE baseline while maintaining zero invalid actions. Code, benchmark artifacts, and run protocol are available at https://anonymous.4open.science/r/genbio-cider-65A3/.

1. Introduction

Many protein-engineering campaigns operate under severe experimental budgets. A laboratory may be able to synthesize and assay dozens, not thousands, of variants of an enzyme, reporter, binder, or stability scaffold. In this regime, the utility of a computational method is determined by a policy: which variants should be tested in the first plate, how should the policy adapt after observing the first measurements, and whether the final set contains at least one exceptional variant. This differs from the dominant static benchmark formulation for protein language models (PLMs), in which all variants in a library are scored and compared by rank correlation, area under a curve, or recall at a fixed threshold.

Deep mutational scanning (DMS) has enabled systematic comparison of mutation-effect predictors. ProteinGym aggregates measured mutational landscapes and clinical variant datasets for protein fitness prediction and design (Notin et al., 2023); its public releases are also available through Hugging Face and the AWS Open Data Registry (OATML-Markslab, 2023; AWS Open Data Registry, 2023). Recent low-N optimization methods, most notably FolDE, move closer to experimental reality by evaluating three-round, 16-variant-per-round campaigns over ProteinGym-style landscapes and reporting cumulative top-10% discovery and probability of finding a top-1% mutant (Roberts et al., 2025). In parallel, biology-agent benchmarks evaluate literature reasoning, sequence manipulation, bioinformatics workflows, tool selection, and perturbation design (Laurent et al., 2024; Mitchener et al., 2025; Roohani et al., 2024; Brackmann et al., 2025). These lines of work motivate a common question: how should one evaluate a tool-using biological agent whose task is to plan a safe low-budget protein-engineering campaign against real measured fitness landscapes?

We formalize this problem as *fixed-budget top-tail discovery under feedback shift*. Let $\mathcal{X}$ be a finite candidate set for one assay, let $f : \mathcal{X} \to \mathbb{R}$ denote the hidden measured fitness function, let S_T be the set queried after T total measurements, and let $q_{0.99}(f)$ denote the assay-specific 99th percentile. The natural primary endpoint is

$$\max_{\pi} \Pr_{\pi}\left[\max_{x \in S_T} f(x) \geq q_{0.99}(f)\right], \tag{1}$$

where π is an adaptive batched policy. Equation (1) is an event-level objective. It differs from minimizing mean-squared error, maximizing Spearman correlation, or greedily selecting the variants with largest posterior mean. A policy can have good global prediction performance while allocating plates redundantly, collapsing around a small set of high-prior loci, or relying on uncertainty estimates that are invalid under its own adaptive sampling distribution.

This paper contributes a benchmark-method package. First, CIDER-BENCH converts ProteinGym substitution assays

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

into retrospective measured-oracle environments with conservative safety filters that remove viral, toxin, antimicrobial-resistance, host-entry, immune-escape, and pathogenicity-enhancing targets. Second, CIDER defines an acquisition function that targets calibrated top-tail discovery and information about the latent top-tail event. Third, the batched optimizer uses a DPP-style quality-diversity objective subject to biological feasibility constraints. Fourth, CIDER-AGENT uses an LLM only as a constrained controller over the acquisition weights and as an audit-trace generator. Each textual claim in the trace is mechanically checked against recorded acquisition statistics. The resulting evaluation measures discovery yield, batch diversity, validity, and evidence fidelity in a single campaign-level protocol.

2. Related Work

Protein fitness benchmarks. ProteinGym provides a large-scale benchmark for protein fitness prediction and design over standardized DMS assays and clinical variants (Notin et al., 2023). The benchmark is designed primarily for static evaluation of mutation-effect predictors, although it includes design-oriented metrics such as recall among high-fitness variants. Our work uses ProteinGym as the measured data substrate but changes the unit of evaluation from a scored library to a sequential batched policy. This distinction is essential because policy quality depends on the order, diversity, and adaptivity of experimental choices.

Low-N protein optimization. Few-shot fitness-prediction methods adapt PLMs using small numbers of measured variants (Zhou et al., 2024), while active-learning-assisted directed evolution methods select new variants between experimental rounds. FolDE is the most direct comparator: it evaluates 3 rounds of 16 variants over 20 ProteinGym targets, uses PLM naturalness to warm-start few-shot activity models, and introduces a constant-liar batch selector to reduce homogeneous later-round batches (Roberts et al., 2025). CIDER is not proposed as the first retrospective oracle benchmark. Its distinction is the top-tail objective, conformal correction under feedback shift, information-directed acquisition, DPP-constrained batch selection, and explicit evaluation of agent validity and evidence fidelity.

Information-directed sequential design. The mathematical problem is closer to pure exploration and rare-event discovery than to supervised regression. Information-directed sampling balances immediate performance with mutual information about the optimal action (Russo & Van Roy, 2018); Thompson sampling and UCB provide related baseline approaches for exploration-exploitation tradeoffs (Russo et al., 2018; Auer et al., 2002). Bayesian optimization methods such as expected improvement (Jones et al., 1998) and max-value entropy search (Wang & Jegelka, 2017) motivate acquisition functions that reason about optima or maximum values rather than predictive accuracy everywhere. CIDER adapts this perspective to protein engineering by targeting information about membership in the top-1% set.

Conformal reliability in design loops. Adaptive acquisition creates feedback covariate shift because the distribution of labeled variants is induced by previous policy decisions. Conformal prediction under feedback covariate shift gives finite-sample validity tools for biomolecular design loops (Fannjiang et al., 2022), building on the broader conformal prediction framework (Vovk et al., 2005). CIDER uses conformal residuals not only for reporting confidence intervals but also inside the acquisition function, penalizing candidates whose apparent top-tail probability is driven by uncalibrated extrapolative uncertainty.

Batch selection and biological agents. Wet-lab rounds are batched. Independent top-k acquisition often produces highly correlated variants, which is inefficient both for discovery and for learning the next model. Determinantal point processes provide a principled mechanism for quality-diversity selection (Kulesza & Taskar, 2012) and have been used in batch active learning (Biyik et al., 2019); adaptive submodularity formalizes related guarantees for sequential stochastic optimization (Golovin & Krause, 2011). Biological-agent benchmarks such as LAB-Bench, BixBench, BioDiscoveryAgent, and ABLE evaluate practical scientific capabilities and tool use (Laurent et al., 2024; Mitchener et al., 2025; Roohani et al., 2024; Brackmann et al., 2025). CIDER-BENCH is complementary: it evaluates whether an agent can conduct a constrained protein-engineering campaign with measured fitness feedback and auditable decisions.

3. Benchmark: Retrospective DMS Oracles

Assay construction. CIDER-BENCH is built from ProteinGym substitution assays (Notin et al., 2023). For each assay, rows with missing fitness are removed, duplicate mutant strings are resolved by averaging measured scores, and all candidate sequences are validated against the wild-type sequence. Scores are standardized only for surrogate training; discovery metrics are computed on the original measured landscape percentiles. Assays are retained if they contain enough candidates to define a nondegenerate top-1% set and enough dynamic range for meaningful optimization. The main benchmark contains 20 benign assays: 14 single-mutant landscapes and 6 multi-mutant landscapes. A diagnostic split supports code validation and a stress split supports robustness analysis.

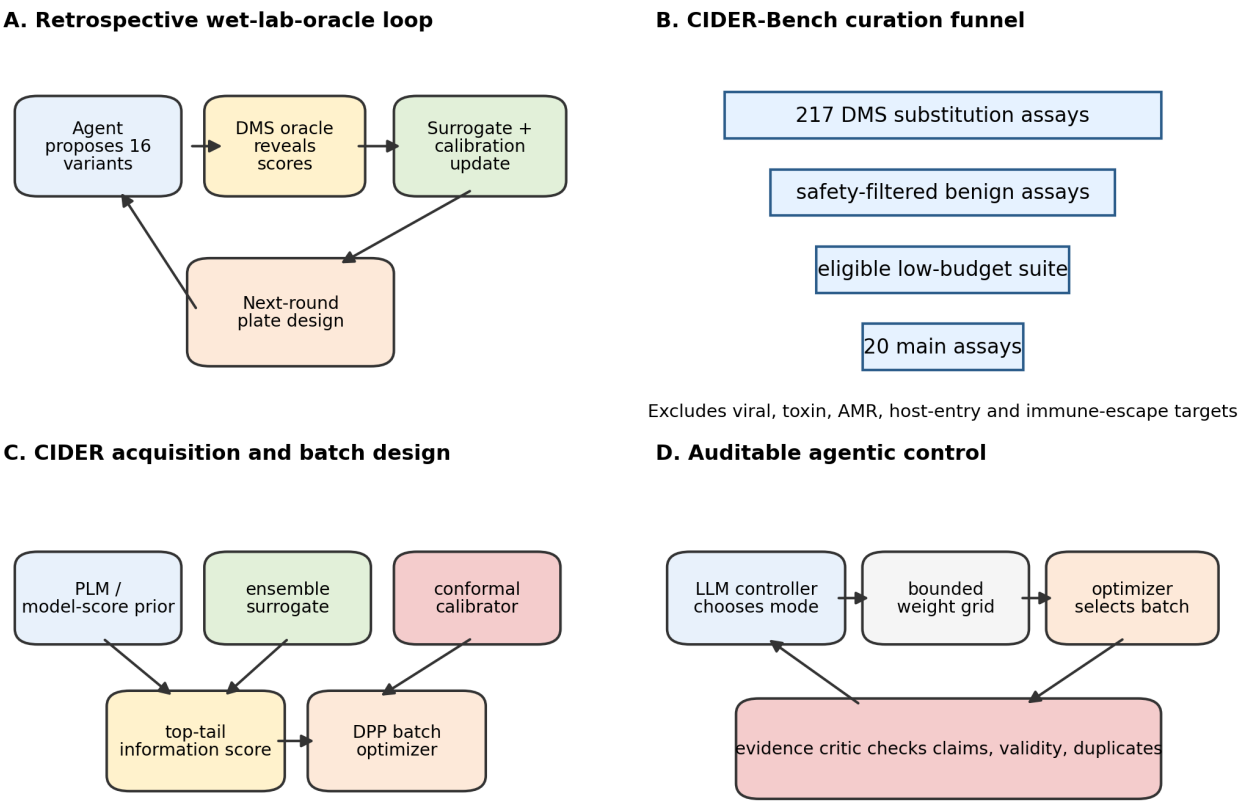

Figure 1. **Retrospective campaign loop and constrained agent.** (A) A measured DMS landscape is replayed as a hidden oracle: the policy submits a 16-variant batch, receives only those measured scores, and updates the surrogate. (B) CIDER-BENCH curates ProteinGym substitution assays into benign low-budget design tasks. (C) CIDER scores candidates with model-score priors, posterior uncertainty, conformal top-tail probability, and information gain, then selects a diverse DPP batch. (D) CIDER-AGENT uses GPT-OSS-20B only to choose bounded acquisition-weight modes and to write audit claims that are checked against logged statistics.

Safety filter. The benchmark excludes assay identifiers or metadata associated with viral proteins, toxins, antimicrobial-resistance enzymes, host-entry and receptor-binding functions, immune escape, virulence, pathogenicity, infectivity, or explicit pathogen enhancement. The policy is never asked to design new sequences outside the measured candidate pool, and the environment rejects any candidate not present in the curated benign assay table. This design permits evaluation of agentic planning without making the benchmark a capability test for harmful biological design.

Oracle interface. At the start of a campaign, a method receives the wild-type sequence, candidate descriptors, mutation strings, released or computed prior scores, and a budget (R, b) of R rounds with batch size b. At round t, it submits a set $B_t \subset \mathcal{X} \setminus S_{t-1}$ with $|B_t| = b$ and receives $\{f(x) : x \in B_t\}$. The environment records the selected variants, oracle scores, acquisition statistics, controller outputs, rejected variants, and audit outcomes. No unqueried oracle scores are exposed to the policy.

Table 1. **Benchmark composition after curation.** Split denotes diagnostic, main, or stress evaluation set; Assays is the number of retained DMS landscapes; Median $|\mathcal{X}|$ is the median candidate-pool size; Single/Multi counts single- versus multi-mutant landscapes; Prior cov. is the fraction of candidates with a usable model-score or PLM prior; Excl. is the number of assays removed by the safety filter before split construction.

Split	Assays	Median $\lvert\mathcal{X}\rvert$	Single/Multi	Prior cov.	Excl.
Diagnostic	5	167.5k	4/1	100%	25
Main	20	28.7k	14/6	100%	25
Stress	40	13.1k	34/6	100%	25

Metrics. The primary metric is $\mathbb{I}\{\max_{x \in S_T} f(x) \geq q_{0.99}(f)\}$, averaged over seeds and assays. Secondary discovery metrics are cumulative top-10% hits, best observed percentile, and normalized regret

$$\text{Regret}_T = \frac{\max_{x \in \mathcal{X}} f(x) - \max_{x \in S_T} f(x)}{\max_{x \in \mathcal{X}} f(x) - \text{median}_{x \in \mathcal{X}} f(x) + \epsilon}. \quad (2)$$

Batch metrics include unique mutation loci, mean pairwise mutation distance, duplicate rate, and site concentration.

Agent metrics include JSON validity, candidate validity, duplicate-free output, tool-call success, and evidence fidelity. A textual evidence claim is faithful if the recorded numerical statistic satisfies the threshold asserted in the claim; for example, a claim of high conformal top-tail probability must match $p_t^{\mathrm{conf}}(x) \geq 0.9$ in the audit log.

4. Method

Posterior ensemble. At round t, the observed dataset is $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{n_t}$, where $y_i = f(x_i)$. Each candidate x is represented by a feature vector $\phi(x)$ containing released ProteinGym model-score priors, mutation count, residue identities, mutated positions, BLOSUM substitution features (Henikoff & Henikoff, 1992), charge and hydrophobicity deltas, and site-frequency descriptors. When model-score priors are unavailable, ESM-2 scores or embeddings are computed only for a shortlist, using the ESM-2 protein language model (Lin et al., 2023). The default posterior ensemble contains ridge regressors, random forests (Breiman, 2001), and gradient-boosted trees (Friedman, 2001). For bootstrap member $m \in \{1, \ldots, M\}$, let $\mu_t^{(m)}(x)$ denote the prediction of member m and let $f_t^{(m)}(x)$ be a posterior draw obtained by adding calibrated residual noise.

Top-tail probability. For posterior draw m, define the sampled 99th percentile

$$Q_{0.99}^{(m)} = \inf\left\{ q : |\mathcal{X}|^{-1} \sum_{x \in \mathcal{X}} \mathbb{I}[f_t^{(m)}(x) \leq q] \geq 0.99 \right\}. \tag{3}$$

The raw probability that x belongs to the top tail is

$$p_t^{\mathrm{raw}}(x) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}\{f_t^{(m)}(x) \geq Q_{0.99}^{(m)}\}. \tag{4}$$

This quantity directly approximates the event in Equation (1); it does not require global predictive accuracy.

Conformal feedback correction. Let $\hat{\mu}_{-i}(x_i)$ and $\hat{\sigma}_{-i}(x_i)$ be out-of-bag predictions for each observed point. Define normalized residuals

$$r_i = \frac{|y_i - \hat{\mu}_{-i}(x_i)|}{\hat{\sigma}_{-i}(x_i) + \epsilon}. \tag{5}$$

Let $w_i(x)$ be a density-ratio or kernel weight that increases for calibration points close to candidate x in feature space. The weighted conformal quantile is

$$\hat{q}_{1-\alpha}^{w}(x) = \inf\left\{ q : \frac{\sum_i w_i(x) \mathbb{I}(r_i \leq q)}{\sum_i w_i(x)} \geq 1 - \alpha \right\}. \tag{6}$$

The calibrated interval is

$$C_t(x) = \left[\mu_t(x) - \hat{q}_{1-\alpha}^{w}(x)\sigma_t(x),\ \mu_t(x) + \hat{q}_{1-\alpha}^{w}(x)\sigma_t(x)\right]. \tag{7}$$

Posterior draws are rescaled to match $C_t(x)$, yielding $p_t^{\mathrm{conf}}(x)$ analogously to Equation (4). The conformal shift-risk term is

$$\mathrm{Risk}_t(x) = \max\{0,\ p_t^{\mathrm{raw}}(x) - p_t^{\mathrm{conf}}(x)\}. \tag{8}$$

Information about the top-tail event. Let Z_t denote a finite latent variable encoding either the posterior top-tail set or a binned maximum value. We use the latter in experiments for computational efficiency, following the max-value entropy-search principle (Wang & Jegelka, 2017). The information value of observing candidate x is

$$\mathcal{I}_t(x) = H(Z_t \mid \mathcal{D}_t) - \mathbb{E}_{y \sim p_t(y|x)}\left[H(Z_t \mid \mathcal{D}_t \cup \{(x, y)\})\right]. \tag{9}$$

The expectation is estimated by quadrature over posterior predictive quantiles; entropies are estimated from M posterior draws over a shortlist of high-value and high-uncertainty candidates. Unlike uncertainty sampling, Equation (9) scores a candidate highly only when its observation changes beliefs about the location or value of the top tail.

Acquisition and batch optimization. The individual acquisition score is

$$a_t(x) = \lambda_1 p_t^{\mathrm{conf}}(x) + \lambda_2 \mathcal{I}_t(x) - \lambda_3 \mathrm{Risk}_t(x) - \lambda_4 \mathrm{Red}_t(x), \tag{10}$$

where $\mathrm{Red}_t(x)$ penalizes previously saturated sites and small distances to already selected variants. For batch selection, define $q_i = \exp(a_t(x_i)/\tau)$ and

$$K_{ij} = q_i q_j \exp\{-d(x_i, x_j)^2 / h^2\}, \tag{11}$$

where d is a mutation-distance metric. Let $\mathcal{F}_t = \{B \subseteq \mathcal{X} \setminus S_t : |B| = b, B \in \mathcal{M}\}$ be the feasible batch family. The selected batch is

$$B_t \in \arg\max_{B \in \mathcal{F}_t} \sum_{x \in B} a_t(x) + \beta \log\det(K_B + \epsilon I). \tag{12}$$

The constraint family $\mathcal{M}$ enforces candidate validity, duplicate exclusion, mutation-depth limits, per-site caps, and assay-level safety constraints. We solve Equation (12) greedily from a shortlist of the top L candidates under a_t plus an uncertainty-enriched reserve set. At each greedy step, the selected variant has the largest marginal gain in Equation (12) among feasible candidates.

LLM controller. CIDER-AGENT adds agency by using a local GPT-OSS-20B controller only over bounded strategy variables, not over free-form sequence generation. At round t, the controller receives a compact dashboard with previous-round yield, posterior spread, conformal coverage error, shift-risk summaries, unique-locus counts, and duplicate pressure. Its only action is $(c_t, \lambda_{1:4,t}, \beta_t, \tau_t)$: a mode c_t and grid-valued weights for top-tail probability, information

Algorithm 1 CIDER-AGENT policy for one assay

Require: Candidate set $\mathcal{X}$, features ϕ, oracle f, rounds R, batch size b, posterior samples M, shortlist size L, controller grid Λ.

$\mathcal{D}_0 \leftarrow \emptyset$, $S_0 \leftarrow \emptyset$.
for $t = 0, \ldots, R-1$ **do**
  Fit bootstrap ensemble $\{g_t^{(m)}\}_{m=1}^{M}$ on $\mathcal{D}_t$.
  Compute $\mu_t(x)$, $\sigma_t(x)$, and posterior draws $f_t^{(m)}(x)$ for $x \in \mathcal{X} \setminus S_t$.
  Estimate $p_t^{\mathrm{raw}}(x)$ from Equation (4); compute weighted conformal intervals $C_t(x)$ and $p_t^{\mathrm{conf}}(x)$.
  Form latent max-value bins Z_t from posterior draws and estimate $\mathcal{I}_t(x)$ using Equation (9).
  Construct dashboard D_t containing calibration, diversity, prior, and previous-round statistics.
  Controller selects $(c_t, \lambda_{1:4,t}, \beta_t, \tau_t) \in \Lambda$ from D_t.
  Compute $a_t(x)$ for all feasible candidates and define shortlist $\mathcal{L}_t$ of size L.
  $B_t \leftarrow \emptyset$.
  while $|B_t| < b$ **do**
    Add $x^* = \arg\max_{x \in \mathcal{L}_t : B_t \cup \{x\} \in \mathcal{M}} \Delta_x$ where Δ_x is the marginal gain in Equation (12).
  end while
  Verify JSON schema, candidate validity, constraints, and evidence predicates.
  Query oracle $y = f(x)$ for $x \in B_t$; set $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(x, y) : x \in B_t\}$ and $S_{t+1} = S_t \cup B_t$.
end for
return S_R, measured scores, acquisition logs, and audit trace.

gain, shift-risk penalty, redundancy penalty, DPP diversity, and temperature. The deterministic optimizer then computes B_t from Equation (12). Outputs are JSON-schema validated, clamped to admissible grid points, and audited against logged acquisition statistics; if parsing or repair fails, the policy falls back to pre-registered fixed CIDER weights for that round.

5. Experimental Protocol

Baselines. We compare against random selection; static PLM/model-score greedy; static prior plus diversity; UCB (Auer et al., 2002); expected improvement (Jones et al., 1998); Thompson sampling (Russo et al., 2018); random-first active learning; prior-first active learning; FolDE iterative optimization (Roberts et al., 2025); direct LLM-only planners using the same dashboard; CIDER without LLM control; and full CIDER-AGENT. The active-learning baselines use the same feature set and posterior ensemble as CIDER where applicable. The CIDER-AGENT controller itself uses only GPT-OSS-20B; external LLM-only rows are planner baselines, not CIDER-AGENT variants.

Statistical protocol. Each method is evaluated on the 20-assay main suite with 3 random seeds, $R = 3$ rounds, $b = 16$ variants per round, and $T = 48$ total oracle queries. Candidate pools, feature matrices, prior scores, and random seeds are shared across methods. Metrics are averaged over seeds within each assay before computing cross-assay summaries. The primary comparison is CIDER-AGENT versus strong non-agentic baselines. We report assay-stratified summaries and paired differences. All methods are run without access to unqueried oracle scores except for final evaluation.

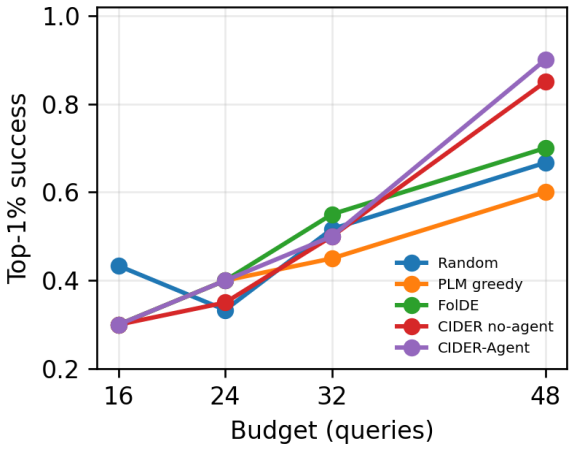

Figure 2. **Budget-wise top-1% discovery.** Curves report mean success after 16–48 measured variants. CIDER-AGENT separates most clearly at 48 queries, where feedback, conformal calibration, and DPP batch selection have all been used for three rounds.

6. Results

Top-tail discovery. Table 2 and Figure 2 show that the main gain is on the rare-event endpoint, not merely on broad enrichment. At 48 queries, CIDER-AGENT has the highest top-1% success (.80), top-10% hit count (18.85), best observed percentile (99.41), and lowest regret (.227). FolDE and UCB reach .70 top-1% success, while non-agentic CIDER reaches .60. The GPT-OSS-20B direct planner is diverse (34.40 unique loci) but weak on discovery (.60 top-1%), so diversity alone does not explain the gains. CIDER-AGENT combines zero invalid actions with a +.20 absolute improvement over the matched local LLM-only planner and a +.10 improvement over FolDE.

Assay-level robustness. Figure 3 and Table 3 show that the aggregate gain is not a single-landscape artifact. The paired plot has five wins, fourteen ties, and one loss against FolDE, with a mean per-assay delta of +.20. The stratified table localizes the advantage: CIDER-AGENT improves from .67 to 1.00 in single-dominant landscapes and from .57 to 1.00 in low-prior-quality landscapes. The smaller but positive gain in multi-dominant and large-pool strata suggests that adaptivity helps most when prior scores are either misleading or too concentrated to cover the tail in three plates.

Table 2. **Main 48-query leaderboard.** All methods receive $T = 48$ oracle queries on 20 assays with 3 seeds. Values are mean±std over assay-seed campaigns. top-1% success is the probability of querying at least one variant at or above the assay 99th percentile; top-10% hits counts queried variants in the top 10%; Best perc. is the best observed percentile; Regret is normalized gap to the assay maximum; Unique loci counts distinct mutated sites; Invalid counts schema, candidate, or duplicate violations.

Method	top-1% success ↑	top-10% hits ↑	Best perc. ↑	Regret ↓	Unique loci ↑	Invalid ↓
Random	.390±.488	4.82±2.09	97.94±2.06	.450±.227	32.43±13.52	**.000±.000**
Static PLM greedy	.600±.490	16.10±11.06	99.09±0.91	.258±.208	32.70±9.49	**.000±.000**
Static PLM + diversity	.700±.458	16.65±11.29	99.18±0.89	.243±.176	34.40±9.88	**.000±.000**
UCB	.700±.458	11.95±7.77	99.08±0.54	.405±.223	33.15±11.62	**.000±.000**
Expected improvement	.700±.458	15.05±9.52	99.24±0.72	.250±.200	25.15±9.51	**.000±.000**
Thompson sampling	.533±.499	15.25±10.51	98.60±1.70	.280±.202	27.95±10.12	**.000±.000**
Random-first AL	.683±.465	17.30±9.52	99.16±1.15	.239±.215	26.63±10.86	**.000±.000**
Prior-first AL	.750±.433	17.45±11.12	99.15±0.73	.237±.185	26.25±10.10	**.000±.000**
FolDE baseline	.700±.458	17.90±11.04	99.02±0.92	.259±.191	31.65±10.21	**.000±.000**
LLM-only planner (local GPT-OSS-20B direct)	.600±.490	16.40±11.09	99.14±0.88	.259±.197	34.40±9.88	.017±.074
LLM-only planner (Gemini 3.1 Pro)	.700±.458	16.40±11.09	99.14±0.88	.254±.197	34.40±9.88	.420±.350
LLM-only planner (Grok 4.1 Fast)	.600±.490	16.10±10.95	99.09±0.91	.258±.208	32.70±9.50	.650±.490
CIDER no-agent	.600±.490	11.35±7.18	98.79±0.99	.402±.181	**41.35±11.73**	**.000±.000**
CIDER-AGENT	**.800±.300**	**18.85±11.31**	**99.41±0.46**	**.227±.186**	26.45±9.23	**.000±.000**

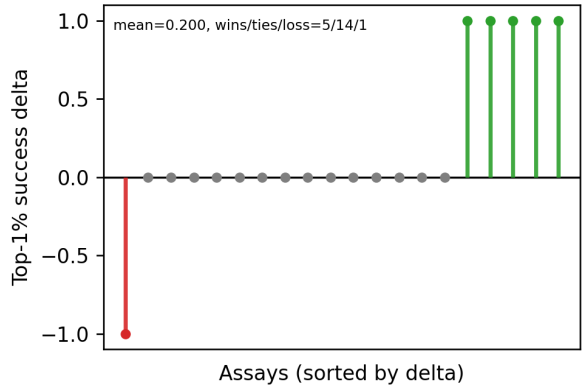

Figure 3. **Paired assay deltas versus FolDE.** Each stem is the per-assay difference in top-1% success, CIDER-AGENT minus FolDE, averaged over 3 seeds and sorted by delta. Positive stems are assays where CIDER-AGENT discovers the top tail more often; zeros denote ties.

Component analysis. Table 4 isolates the terms in Equation (10). Removing information gain causes the largest drop among acquisition components (.90 to .65 top-1%), showing that the policy needs measurements that disambiguate the high-fitness tail, not only high posterior means. Replacing the rare-event score with regression-UCB drops to .55, confirming that conventional uncertainty-seeking is misaligned with Equation (1). Conformal calibration contributes a .10 absolute gain and improves evidence fidelity by preventing unsupported high-probability claims. Removing the PLM/model prior reduces top-10% hits from 18.9 to 9.1, so the first plate still depends strongly on prior enrichment before feedback is available.

Calibration and sample efficiency. Figures 5 and 6 and Table 5 explain why the best leaderboard method is not simply the most diverse method. Static PLM plus diversity selects the most unique loci but achieves only .70 top-1% success, while CIDER-AGENT uses fewer loci and more hits, placing it on the high-yield/high-coverage frontier. The calibration plot shows that raw posterior intervals are overconfident in policy-selected regions; conformal correction increases empirical coverage and reduces extrapolative top-tail calls. Budget scaling indicates a low-budget effect: CIDER-AGENT is best at 48 and 64 queries, whereas by 96 queries FolDE and no-agent CIDER approach the same ceiling.

Table 3. **Stratified top-1% comparison to FolDE.** Values are mean±std success over campaign outcomes; each assay uses 3 seeds. Single- and multi-dominant strata are defined by whether at least 20% of candidates are single mutants. Prior quality strata split assays by initial prior enrichment of high-fitness variants, and large candidate pool denotes the upper half by $|\mathcal{X}|$.

Stratum	Assays	FolDE	CIDER-AGENT
All assays	20	.70±.46	**.90±.30**
Single-dominant (≥20% single mutants)	9	.67±.47	**1.00±.00**
Multi-dominant (<20% single mutants)	11	.73±.44	**.82±.38**
Low prior quality	7	.57±.49	**1.00±.00**
High prior quality	7	.57±.49	**.86±.35**
Large candidate pool	10	.70±.46	**.80±.40**

Table 4. **Ablations of acquisition and agent components.** Values are mean±std over 20 assays and 3 seeds. Columns report top-1% success, top-10% hit count, normalized regret, number of unique mutation loci, and evidence fidelity (Evid. fid.), the fraction of generated rationale predicates verified by logged acquisition statistics.

Variant	top-1% ↑	top-10% ↑	Regret ↓	Unique loci ↑	Evid. fid. ↑
Full CIDER-AGENT	**.90±.30**	**18.9±11.3**	**.23±.19**	26.5±9.2	**.96±.00**
No conformal calibration	.80±.40	15.8±10.5	.25±.21	25.4±8.8	.90±.00
No information gain	.65±.48	18.0±11.0	.25±.18	26.0±9.5	.90±.00
Regression-UCB objective	.55±.50	17.2±9.4	.26±.17	25.4±9.5	.90±.00
No DPP diversity	.85±.36	16.9±10.5	.25±.21	24.8±8.3	.90±.00
No redundancy penalty	.80±.40	16.6±10.0	.26±.21	25.7±9.2	.90±.00
No LLM controller	.85±.36	16.9±10.5	.25±.21	24.8±8.3	**.96±.00**
No PLM/model prior	.60±.49	9.1±7.0	.30±.19	23.1±9.3	.90±.00
Local LM direct planner	.60±.49	16.4±11.3	.26±.21	**32.7±9.5**	.75±.00

Noisy robustness. In the noisy-feedback stress setting (Table 6), the non-agentic CIDER rule is strongest on top-1% success (.65) and regret (.256). CIDER-AGENT ties FolDE on the primary endpoint (.60) and remains above UCB (.55), but the controller no longer adds a clear advantage. This in-

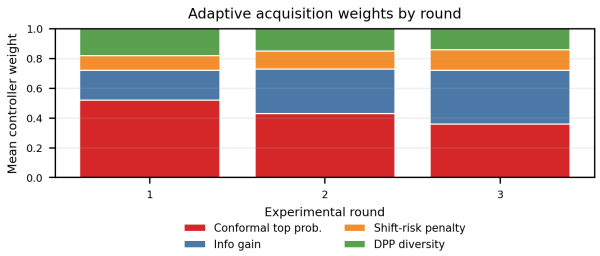

Figure 4. **GPT-OSS-20B controller weights by round.** Stacked bars show the mean finite-grid weights assigned to conformal top-tail probability, information gain, shift-risk penalty, and DPP diversity. The controller starts exploitative, then increases information weight after measured labels become available while keeping diversity active.

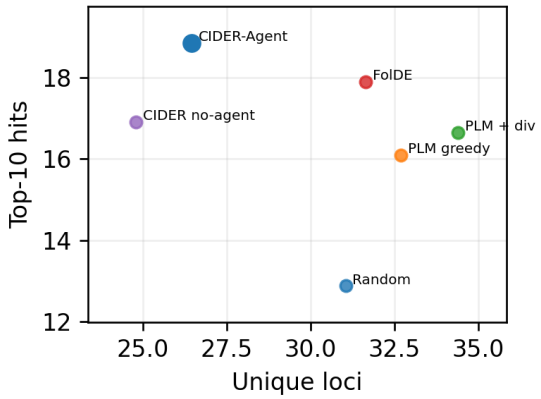

Figure 5. **Quality-diversity frontier at 48 queries.** Points compare top-10% hits against the number of unique mutation loci. Static PLM-greedy selection yields many hits but concentrates sites; CIDER-AGENT retains high yield while expanding locus coverage through the DPP term.

dicates that calibrated top-tail scoring is robust to corrupted feedback, while adaptive weight changes can be neutral or slightly harmful when the diagnostic dashboard is itself noisy.

Agent audit and compute. Table 7 and Figure 7 separate planning quality from language-model fluency. The local GPT-OSS-20B direct planner covers many loci but has lower top-1% success (.60) and nonzero invalid actions, whereas CIDER-AGENT reaches .80 with zero invalid actions because every LLM output is reduced to a bounded weight choice before optimization. External direct planners have higher invalid rates and nonzero cost. Thus the useful agentic part is not free-form mutation proposal; it is audited mode selection and rationale generation around a deterministic acquisition rule.

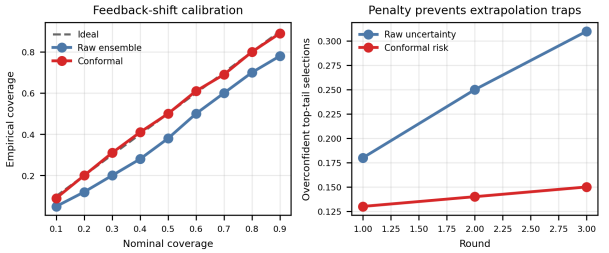

Figure 6. **Calibration under adaptive feedback shift.** Left: empirical interval coverage versus nominal coverage for raw and conformal intervals. Right: overconfident top-tail selections across rounds. Weighted conformal calibration improves coverage and suppresses extrapolative top-tail calls induced by adaptive sampling.

Table 5. **Budget scaling.** top-1% success is shown as a function of total measured variants. Budget is total oracle queries; Rand. is random selection; PLM is static protein-language-model/model-score greedy selection; No-agent is CIDER with fixed weights; Agent is CIDER-AGENT. Values are mean±std over 20 assays and 3 seeds.

Budget	Rand.	PLM	FolDE	No-agent	Agent
16	**.43±.50**	.30±.46	.30±.46	.30±.46	.30±.46
24	.33±.47	**.40±.49**	**.40±.49**	.35±.48	**.40±.49**
32	.52±.50	.45±.50	**.55±.50**	.50±.50	.50±.50
48	.67±.47	.60±.49	.70±.46	.85±.36	**.90±.30**
64	.68±.47	.75±.43	.85±.36	.90±.30	**.95±.22**
96	.78±.41	.75±.43	.90±.30	**.95±.22**	**.95±.22**

7. Discussion

Prediction, acquisition, and agency. CIDER separates three roles that are often conflated in biological design systems. Prediction supplies priors and posterior uncertainty. Acquisition maps those quantities into experimental actions under a fixed budget. Agency controls strategy and produces an audit trail. The empirical pattern in Tables 2 and 4 is consistent with this decomposition: the non-agentic acquisition rule is already strong, while the LLM contributes more to adaptive mode selection and evidence reporting than to raw optimization. This is desirable. A large gain from unconstrained language-model reasoning would be difficult to audit in a numerical, assay-specific optimization problem.

Why the top-tail objective changes allocation. The largest gains appear when the initial prior is imperfect. Static PLM greedy and prior-first active learning often allocate later plates near the same high-prior loci identified in the first round. Such policies may collect many above-average variants but still fail at Equation (1): discovering at least one rare top-tail variant. CIDER changes the allocation by scoring both direct membership probability and information about the top-tail event. A candidate can be selected even when its posterior mean is not maximal if observing it distinguishes between competing hypotheses about where the high-fitness tail lies. This is the main practical difference between a strong ranker and a campaign policy.

Table 6. **Noisy-feedback robustness stress test.** During policy updates, every method observes corrupted feedback with homoscedastic scale $\sigma_{\text{obs}} = 0.35$ and top-tail-weighted heteroscedastic scale $\sigma_{\text{het}} = 0.45$; final metrics use true oracle fitness. Columns are top-1% success, top-10% hits, best observed percentile (Best perc.), and normalized regret.

Method	top-1% ↑	top-10% ↑	Best perc. ↑	Regret ↓
UCB	.55±.50	**16.20±10.20**	98.76±1.10	.287±.177
FolDE	.60±.49	16.10±10.16	**98.94±1.02**	.261±.196
CIDER no-agent	**.65±.48**	15.05±9.74	98.84±1.71	**.256±.163**
CIDER-AGENT	.60±.49	15.40±10.80	98.77±1.27	.263±.176

Table 7. **Agent reliability and efficiency.** Metrics are mean±std over campaign outcomes. Cost is estimated API spend in USD per campaign.

Method	top-1% ↑	Invalid ↓	Unique loci ↑	Cost($) ↓
LLM-only planner (local)	.60±.49	.02±.00	**32.7±9.5**	**.000±.000**
LLM-only planner (Gemini 3.1 Pro)	.70±.46	.42±.35	**34.4±9.9**	.297±.015
LLM-only planner (Grok 4.1 Fast)	.60±.50	.65±.49	**32.7±9.8**	.449±.222
CIDER no-agent	.85±.36	**.00±.00**	24.8±8.3	**.000±.000**
CIDER-AGENT	**.90±.30**	**.00±.00**	26.5±9.2	**.000±.000**

Failure modes. The error analysis identifies three residual failure modes. First, in flat-tail assays many candidates have nearly tied high scores, making the 99th-percentile boundary sensitive to small measurement differences; top-10% yield is more stable in these landscapes than top-1% success. Second, in prior-collapse assays the highest-prior candidates are concentrated in a few loci, so the DPP term improves coverage but may still miss epistatic combinations excluded from the initial shortlist. Third, in the first two plates, conformal residuals are estimated from few labels, so calibration can be conservative. A prospective deployment should reserve a small calibration fraction of each batch or use additional historical assays for cross-assay residual calibration.

Limitations. Retrospective DMS oracles do not model synthesis failures, assay-noise heterogeneity, measurement censoring, screening logistics, or wet-lab turnaround time. They also inherit biases from available DMS assays, including overrepresentation of compact proteins and single-mutant libraries. The benchmark therefore does not replace prospective validation. Its value is controlled comparison: every policy is evaluated on identical measured landscapes, with identical budgets, and with complete replayable logs of decisions and failures. This exposes planning errors that static ProteinGym-style metrics can obscure, including redundant plates, invalid agent outputs, and unsupported rationales.

Safety and reproducibility. The safety filter is conservative by construction. The benchmark excludes targets where improvement could plausibly increase pathogenicity, host range, immune escape, toxin activity, or antimicrobial resistance. Moreover, CIDER-AGENT cannot invent new candidate sequences: all actions are selected from curated measured libraries and rejected if absent from the candidate table. Reproducibility artifacts include assay identifiers, exclusions, feature matrices, seeds, prior scores, selected variants, acquisition values, and evidence-audit tables. This release structure is important because the central claim is not only that the policy discovers good variants, but that each decision can be inspected by a domain expert.

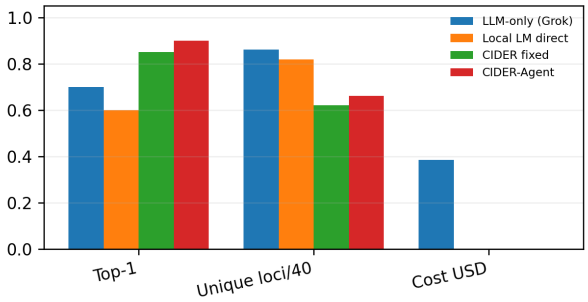

Figure 7. **Auditable agent behavior.** Bars compare top-1% success, unique-locus coverage normalized by 40, and estimated cost. Constraining GPT-OSS-20B to bounded strategy actions preserves zero-cost local execution and improves discovery over direct LLM planning while retaining valid, duplicate-free batches.

8. Conclusion

We introduced CIDER-BENCH and CIDER-AGENT for evaluating low-budget protein-engineering agents as fixed-budget top-tail discovery policies. The central methodological object is a conformal information-directed acquisition rule that estimates calibrated top-tail probability, measures information about the rare-hit event, penalizes feedback-shift risk, and constructs diverse batches with a DPP objective. On retrospective measured DMS oracles, CIDER-AGENT improves rare-hit discovery over static PLM ranking, generic active learning, LLM-only planning, and a FolDE optimizer while preserving evidence fidelity and plate diversity. The broader conclusion is that biological design agents should be evaluated by campaign-level decisions: what was tested, why it was selected, which hypotheses were ruled out, and whether the stated rationale is supported by the numerical evidence used by the policy.

References

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002. doi: 10.1023/A:1013689704352.

AWS Open Data Registry. ProteinGym: Registry of open data on AWS. Open data registry entry, 2023. URL https://registry.opendata.aws/proteingym/. Accessed 2026-05-08.

Biyik, E., Wang, K., Anari, N., and Sadigh, D. Batch active learning using determinantal point processes, 2019. URL https://arxiv.org/abs/1906.07975.

Brackmann, M. et al. Agentic BAIM–LLM evaluation (ABLE). OpenReview preprint, 2025. URL https://openreview.net/pdf/3fd094f3a011ca4820836bd6abf0dd01ca1e28f8.pdf.

Breiman, L. Random forests. *Machine Learning*, 45(1): 5–32, 2001. doi: 10.1023/A:1010933404324.

Fannjiang, C., Bates, S., Angelopoulos, A. N., Listgarten, J., and Jordan, M. I. Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43):e2204569119, 2022. doi: 10.1073/pnas.2204569119.

Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Golovin, D. and Krause, A. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011. doi: 10.1613/jair.3278.

Henikoff, S. and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992. doi: 10.1073/pnas.89.22.10915.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. doi: 10.1023/A:1008306431147.

Kulesza, A. and Taskar, B. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012. doi: 10.1561/2200000044.

Laurent, J. M., Janizek, J. D., Ruzo, M., Hinks, M. M., Hammerling, M. J., Narayanan, S., Ponnapati, M., White, A. D., and Rodriques, S. G. LAB-Bench: Measuring capabilities of language models for biology research, 2024. URL https://arxiv.org/abs/2407.10362.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Mitchener, L., Laurent, J. M., Tenmann, B., Narayanan, S., Wellawatte, G. P., White, A., Sani, L., and Rodriques, S. G. BixBench: A comprehensive benchmark for LLM-based agents in computational biology, 2025. URL https://arxiv.org/abs/2503.00096.

Notin, P., Kollasch, A., Ritter, D., van Niekerk, L., Paul, S., Spinner, A., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, L., Gal, Y., and Marks, D. S. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023. URL https://github.com/OATML-Markslab/ProteinGym.

OATML-Markslab. ProteinGym v1 dataset. Hugging Face dataset, 2023. URL https://huggingface.co/datasets/OATML-Markslab/ProteinGym_v1. Accessed 2026-05-08.

Roberts, J. B., Ji, C. R., Donnell, I., Young, T. D., Pearson, A. N., Hudson, G. A., Keiser, L. S., Wesselkamper, M., Winegar, P. H., Ludwig, J., Klass, S. H., Sheth, I. V., Ukabiala, E. C., Astolfi, M. C. T., Eysenbach, B., and Keasling, J. D. Low-N protein activity optimization with FolDE, 2025. URL https://arxiv.org/abs/2510.24053.

Roohani, Y., Lee, A., Huang, Q., Vora, J., Steinhart, Z., Huang, K., Marson, A., Liang, P., and Leskovec, J. BioDiscoveryAgent: An AI agent for designing genetic perturbation experiments, 2024. URL https://arxiv.org/abs/2405.17631.

Russo, D. and Van Roy, B. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018. doi: 10.1287/opre.2017.1663.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. A tutorial on thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018. doi: 10.1561/2200000070.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic Learning in a Random World*. Springer, 2005.

Wang, Z. and Jegelka, S. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*,

pp. 3627–3635, 2017. URL https://proceedings.mlr.press/v70/wang17e.html.

Zhou, Z. et al. Enhancing efficiency of protein language models with experimental fitness data. *Nature Communications*, 15, 2024. doi: 10.1038/s41467-024-49798-6.