Table A-1: Comparison of top-performing LLM approaches for Bayesian optimization across domains. Errors mark standard deviation. Bold numbers mark best performance by 95% confidence in separation of means.

(a) Molecular optimization using string-based representations

	TYK2, Poor start		**D2R, Random Start**	
Method	Recall@2% ↑	RMSE ↓ $\times 10^{-3}$	Recall@2% ↑	RMSE ↓ $\times 10^{-3}$
Best statistical method	0.32 ± 0.25	160 ± 47	0.80 ± 0.05	90 ± 5
Best Agent	$\mathbf{0.54 \pm 0.12}$	$\mathbf{93 \pm 7}$	0.78 ± 0.05	$\mathbf{86 \pm 6}$
Best Off-the-Shelf	0.19 ± 0.16	159 ± 84	0.61 ± 0.09	91 ± 5

(b) Four-residue Protein Optimization

	GB1, Budget 60		**TrpB, Budget 60**	
Method	Recall@0.5% ↑ $\times 10^{-3}$	Max Fitness ↑	Recall@0.5% ↑ $\times 10^{-3}$	Max Fitness ↑
Best statistical method	12 ± 7	0.63 ± 0.17	9 ± 10	0.51 ± 0.32
Best Agent	5 ± 6	0.50 ± 0.25	7 ± 7	0.57 ± 0.20
Best Off-the-Shelf	14 ± 7	0.56 ± 0.05	$\mathbf{16 \pm 9}$	0.58 ± 0.05

Method	TYK2 Bad Start		TYK2 Random Start		D2R Bad Start		D2R Random Start	
	Recall@2% ↑	MAX ↑	Recall@2% ↑	MAX ↑	Recall@2% ↑	MAX ↑	Recall@2% ↑	MAX ↑
GREEDY	0.16 ± 0.14	0.90 ± 0.07	0.62 ± 0.04	1.00 ± 0.00	0.47 ± 0.17	0.96 ± 0.02	0.52 ± 0.13	0.97 ± 0.03
UCB	0.32 ± 0.25	0.93 ± 0.07	0.69 ± 0.04	1.00 ± 0.00	0.68 ± 0.08	0.97 ± 0.02	0.80 ± 0.05	0.99 ± 0.01
TS	0.19 ± 0.15	0.91 ± 0.07	0.65 ± 0.04	1.00 ± 0.00	0.56 ± 0.15	0.96 ± 0.02	0.71 ± 0.07	0.97 ± 0.02
EI	0.45 ± 0.03	1.00 ± 0.00	0.45 ± 0.03	1.00 ± 0.00	0.65 ± 0.06	1.00 ± 0.01	0.62 ± 0.04	1.00 ± 0.01
ϵ-GREEDY	0.43 ± 0.16	0.97 ± 0.04	0.62 ± 0.04	1.00 ± 0.00	0.45 ± 0.15	0.95 ± 0.02	0.59 ± 0.10	0.97 ± 0.03
AGENT	0.47 ± 0.14	1.00 ± 0.01	0.62 ± 0.11	1.00 ± 0.00	0.70 ± 0.09	0.98 ± 0.02	0.65 ± 0.10	0.96 ± 0.02
SIMPLEAGENT	0.54 ± 0.12	1.00 ± 0.01	0.64 ± 0.06	1.00 ± 0.00	0.69 ± 0.08	0.99 ± 0.00	0.78 ± 0.05	0.99 ± 0.02
DeepSeek-R1	0.20 ± 0.11	0.95 ± 0.05	0.45 ± 0.14	1.00 ± 0.00	0.40 ± 0.14	0.98 ± 0.03	0.53 ± 0.11	0.96 ± 0.02
GPT-5	0.19 ± 0.16	0.95 ± 0.04	0.30 ± 0.18	0.95 ± 0.05	0.44 ± 0.17	0.95 ± 0.01	0.61 ± 0.09	0.96 ± 0.02
Qwen3	0.17 ± 0.17	0.92 ± 0.07	0.52 ± 0.06	1.00 ± 0.00	0.39 ± 0.17	0.95 ± 0.01	0.53 ± 0.10	0.97 ± 0.03

Table A-2: Performance comparison of acquisition methods in the molecular task by Recall@2% and max fitness achieved (**MAX**). Errors mark the standard deviation. N=5 for GPT-5 and Qwen3; N=10 for all others.

A Additional results

A.1 LLMs have trouble comprehending SMILES

An in-depth analysis of the off-the-shelf LLM responses revealed several failure modes. Firstly, the Qwen3 (and Llama-maverick-4, no reasoning, and Claude-Sonnet-3.7, not included in data) were greedy with respect to selecting SMILES strings. When prompted with a table sorted by predicted affinity, they exclusively chose from the very top of the table. When the table was shuffled, they sometimes appeared only to process subsets of the table. As seen in Response A.1, Llama-4-maverick has a very high probability of selecting a candidate index following a previously selected candidate index.

The reasoning was often advanced, but despite extensive prompt engineering efforts, LLMs struggled to implement their stated strategies in practice. Models would provide excellent rationales for decisions and develop sophisticated AL strategies during the reasoning phase, yet default to simple heuristics during the selection process. The reasoning around specific molecule structures was often extremely brief and related only to one or two substructures, often on the edges of the SMILES string (Figure A-1, right). The model frequently confused different substructures in multi-ring systems.

To further investigate SMILES comprehension, we asked Qwen3 to summarize different SMILES from our database (Figure A-1, left). The summaries often included several correctly named substructures and correct chemical properties, but the relation between the substructures was absent. Qwen3 also used SMILES substrings to design hypotheses and filters. This led to very dense summaries of the

Method	Budget 60, GB1		Budget 480, GB1		Budget 60, TrpB	
	Recall@0.5% ↑ $\times 10^3$	Max Fitness ↑	Recall@0.5% ↑ $\times 10^3$	Max Fitness ↑	Recall@0.5% ↑ $\times 10^3$	Max Fitness ↑
GREEDY	12 ± 7	0.63 ± 0.17	231 ± 60	0.89 ± 0.13	9 ± 10	0.51 ± 0.32
UCB	9 ± 7	0.56 ± 0.18	196 ± 53	0.90 ± 0.12	7 ± 8	0.50 ± 0.30
TS	12 ± 8	0.62 ± 0.17	216 ± 55	0.90 ± 0.12	9 ± 9	0.52 ± 0.29
EI	7 ± 7	0.53 ± 0.18	131 ± 44	0.90 ± 0.12	6 ± 7	0.46 ± 0.29
ϵ-GREEDY	11 ± 7	0.59 ± 0.18	205 ± 65	0.90 ± 0.12	8 ± 9	0.50 ± 0.30
AGENT	4 ± 4	0.45 ± 0.17	166 ± 71	0.90 ± 0.17	7 ± 8	0.45 ± 0.28
SIMPLEAGENT	5 ± 6	0.50 ± 0.25	194 ± 52	0.90 ± 0.08	7 ± 7	0.57 ± 0.20
DeepSeek	10 ± 8	0.51 ± 0.16	-	-	22 ± 12	0.65 ± 0.08
GPT-5	14 ± 7	0.56 ± 0.05	74 ± 31	0.71 ± 0.18	16 ± 9	0.58 ± 0.05
Qwen3	3 ± 3	0.46 ± 0.14	5 ± 4	0.54 ± 0.24	11 ± 8	0.61 ± 0.08
SFT	8 ± 8	0.52 ± 0.22	53 ± 24	0.75 ± 0.15	4 ± 5	0.35 ± 0.31
DeepSeek-BLIND	18 ± 16	0.55 ± 0.19	-	-	7 ± 9	0.46 ± 0.29
Qwen3-BLIND	3 ± 4	0.41 ± 0.19	-	-	3 ± 5	0.43 ± 0.32
GPT-5-BLIND	20 ± 11	0.60 ± 0.13	-	-	10 ± 15	0.45 ± 0.35

Table A-3: Performance comparison of acquisition methods in the protein task. Errors mark the standard deviation. N=50 for statistical methods, N=10 otherwise.

chemical space, essentially containing copy-pasted substrings (Response B.7.1). The analysis showed that the reasoning models were capable of utilizing substructure information from the SMILES, but they require significant token usage. When prompted with more SMILES, precision decreased drastically. Deepseek-R1 reasoned similarly. Qwen3 and Deepseek-R1 relied heavily on overly simple filters that only captured minimal information about a given molecule's performance. Other studies have similarly shown that reducing detailed numeric or conceptually complex data increases performance [Agarwal et al., 2025].

Interestingly, GPT5 used a completely different language. The summaries were more extensive (Response B.7.1) and detailed. It derived relations between substructures, for example *"Substructure filter: pyridyl-diamide with two secondary amides: ArL̂–NH–C(=O)–pyridyl–NH–C(=O)–ArR̂"*. GPT5's failure likely lay in its ability to apply such derived filters in practice. We observed several instances in which it incorrectly mapped even simple substructures, stating that it found sulfonyl groups in the indices representing molecules that did not contain a single sulfur atom. GPT5 has indeed been shown to perform better than other models on atom mapping, which may translate to better understanding relations between substructures, but not on SMILES to IUPAC [Runcie, 2025, Runcie et al., 2025]. We aim to investigate this further in subsequent work.

GPT actually recognized that the chemical space was constrained in the clustered, poor, starting configurations (Figure 2.b), but did nothing about it: `"The chemical space is tightly focused around a single core with well-behaved, monotonic SAR along three modular regions."`

Response 1: Llama-4-maverick response from processing chunk

```
...
Here are 60 selected indices, ensuring a mix across the entire range and
    diversity in chemical structures and predicted properties:

<selected_indices>
[38, 497, 244, 248, 293, 294, 999, 1000, 113, 116, 140, 599, 605, 758, 762,
    838, 839, 847, 851, 884, 885, 937, 940, 949, 950, 951, 952, 953, 954, 10,
     20, 31, 32, 33, 34, 39, 48, 52, 60, 62, 70, 106, 108, 115, 117, 118,
    119, 121, 122, 123, 124, 125, 130, 131, 132, 133, 134, 135, 136, 137,
    138, 139]
</selected_indices>
```

A.2 Comprehensive benchmark of statistical methods

In the main article, we compared the performance of LLM-enabled methods with two statistical acquisition functions: UCB and GREEDY. These were chosen as they performed best among statistical methods in the molecular and protein tasks, respectively. Here, we present a more comprehensive benchmark that encompasses five standard statistical acquisition strategies: upper confidence bound (UCB), greedy, Thompson sampling (TS), expected improvement (EI), and ϵ-greedy with $\epsilon = 0.05$. To provide an overview of the relative performances, we report the average rank of each method across tasks (1 being the best), along with confidence intervals estimated via bootstrapping over random seeds.

Performances of all statistical methods on the conditions evaluated in the main article, compared to SIMPLEAGENT and GPT-5, are in Figures A-7 and A-8.

On the molecular task, UCB ranked the best of the statistical methods across all tasks (4) and metrics (2) evaluated (UCB average rank 1.7 $[1.5, 1.875]$, EI: 1.96, TS: 2.88, ϵ-GREEDY: 3.04, GREEDY: 4.06 by recall@2% and max fitness achieved). EI discovered candidates with significantly higher fitness than other methods (EI average rank 1.07 $[1.0, 1.25]$, UCB: 1.88 by max fitness achieved), but performed poorly on the recall metric (EI average rank 2.86 $[2.5, 3.25]$, UCB: 1.5 by recall@2%). SIMPLEAGENT ranked higher on average against the statistical methods across tasks and metrics (SIMPLEAGENT average rank 1.78 $[1.375, 2.125]$, UCB: 2.2 $[2.0, 2.375]$, EI: 2.52, TS: 3.68, ϵ-GREEDY: 3.89, GREEDY: 4.92), but UCB performed better on tasks where the campaign was randomly initialized (UCB average rank 1.23 $[1.0, 1.5]$, SIMPLEAGENT: 2.05). All results available in Table A-2.

On the protein task, GREEDY ranked the best of the statistical methods across all tasks and metrics (GREEDY average rank 1.81 $[1.25, 2.38]$, TS: 2.25, UCB: 3.25, ϵ-GREEDY: 3.31, EI: 4.38, by recall@0.5% and max fitness achieved, across all conditions). GPT-5 ranked on average slightly better against the statistical methods by recall on short campaigns (GPT-5 average rank 2.3517 $[1.0, 5.0]$, GREEDY: 2.67) but otherwise underperformed. All results available in Table A-3.

It should be noted that the average rank provides only an overview of an acquisition function's performance and cannot be used to make stronger claims.

A.3 UCB sensitivity

The agentic workflows can employ UCB to select candidates. At each call to the UCB function, the agent sets the hyperparameter β, $score = \mu + \sigma\sqrt{\beta}$, which controls the trade-off between exploration and exploitation. Vanilla use of UCB requires the user to decide β in advance, raising the question of how sensitive UCB performance is to the choice of β. Figure A-9 reports UCB performance on the molecular task as β varies from 0 (greedy) to 25. With random initialization, UCB remained robust across a broad range ($\beta \in [2, 20]$), but performance deteriorated sharply under a poor start. The agent's ability to adaptively tune β therefore provides additional robustness.

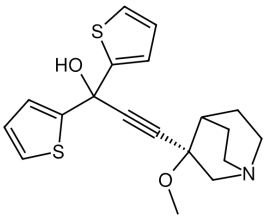
A.4 How informative is the starting point sampled by GPT-5?

GPT-5 performed well on the GB1 protein task. We investigated whether this advantage arose from the initial points logically selected by GPT-5 in the first batch, rather than from its subsequent acquisition strategy. Figure A-10 shows the performance of Greedy, UCB, and the **SIMPLEAGENT** workflow when initialized with GPT-5's first-batch selections (N=5). As statistical methods performed worse under this setup, GPT-5's superior performance cannot be entirely attributed to its choice of starting points, confirming the strength of its acquisition strategies.

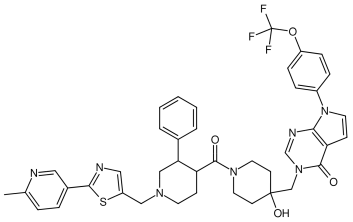
A.5 LLMs as Acquisition Functions and Generative Models

We elaborate on our decision to use off-the-shelf LLMs as acquisition functions in the molecular task and as generative models in the protein task, based on differences in search space structure and representation complexity of the two tasks.

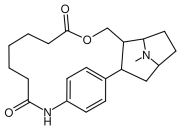
In the molecule task, the search space is characterized by (i) complex representations (long SMILES strings with strict syntactic constraints) and (ii) a sparse and constrained search space. To use LLMs as a generative model, either the LLM must already be aware of the valid subset of SMILES or



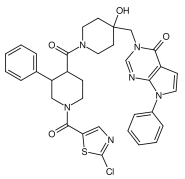
"Carbonyl (CO) ×, thiol groups (two -SCH2-) ×, hydroxyl (terminal -OH), alkyne (C#C). Five-membered heterocyclic ring (piperidine derivative) with chiral center. Alkyne is rigid; rest of molecule flexible. H-bond donors: hydroxyl; acceptors: carbonyl. Moderate size (∼15 heavy atoms) ×. Complex structure with multiple stereocenters and functionalgroups."



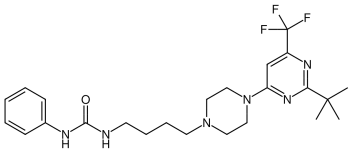
"...thiazole-piperidine with trifluoromethoxy group.." ✓



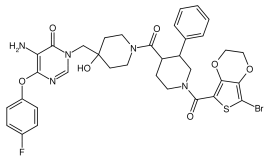
"Pyrrolidine ring (heterocyclic), benzene ring (aromatic), ester (C=O-O), amide (C=O-NH). Flexible backbone with multiple rotatable bonds (ester, amide, alkyl chains). H-bond donors: amide NH, pyrrolidine NH. Acceptors: amide carbonyl, ester carbonyl, ester O. Medium size, complex structure with 4 rings and 5 functional groups." ✓



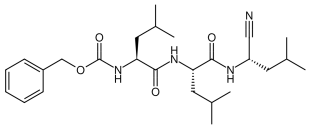
"...chlorothiazole variant with high confidence..."✓



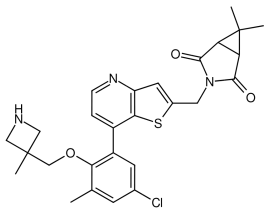
Key functional groups: isopropyl, piperidine (heterocyclic) ×, benzene (aromatic), carbonyl, amide, trifluoromethyl. Ring systems: six-membered piperidine, aromatic benzene. Flexibility: multiplerotatable bonds (amide, piperidine, side chains). H-bonding: amide NH (donor), carbonyl (acceptor), piperidine N (donor/acceptor). Size: ∼20 heavy atoms ×, moderate complexity. Cationic: piperidine N may protonate at physiological pH. Hydrophobic: trifluoromethyl, aromatic rings; hydrophilic: amide, piperidine.



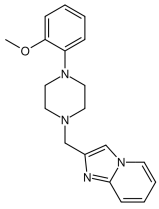
"...brominated benzodioxole, unique substitution pattern..."×



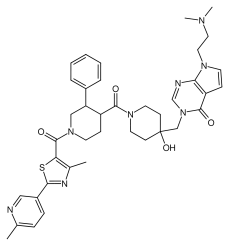
"Tripeptide with nitrile (C#N), isopropyl, and phenyl side chains. Aromatic benzene ring. Multiple rotatable bonds (peptide backbone, side chains). Amide NH donors (3) and carbonyl acceptors (3). Ester O (acceptor). Chiral centers (3). ∼18 heavy atoms ×. Cationic at physiological pH (nitrile, amides). Hydrophobic (isopropyl, phenyl) and hydrophilic (amide, ester) regions.



"...cyclopropylmethyl variant of successful scaffold..."×



"Aromatic benzene ring with methoxy group (OCH_3) at C1. Piperazine (six-membered, two N) and pyrrolidine (five-membered, one N) heterocycles. Flexibility from ring conformations and single bonds. H-bond donors: piperazine NH, pyrrolidine NH ×; acceptors: methoxy O. Moderate size ∼15 heavy atoms×), complex structure with multiplerings."



"...with a dimethylaminoethyl group..."✓

Figure A-1: **Left:** Qwen3 responses when asked to summerize given SMILES. **Right:** Qwen3 reasoning about given SMILES during selection or chemical space summary.

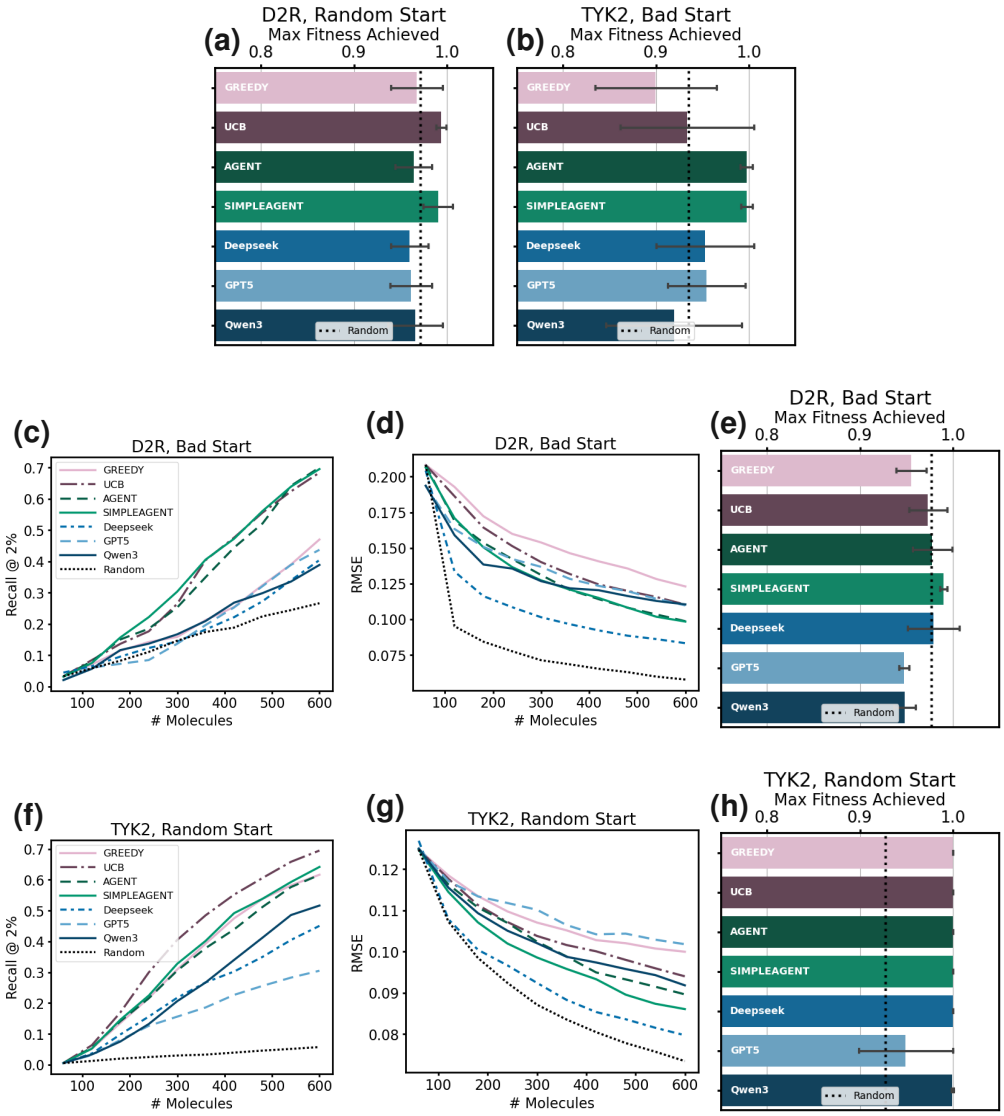


Figure A-2: **Top row:** Complement to Figure 2, max fitness achieved in D2R campaign with random start (**a**) and in TYK2 campaign with poor start (**b**). **Middle row:** Campaign results from D2R campaign with poor start. **Bottom row:** Campaign results from TYK2 with random start. Error bars mark standard deviation. The SIMPLEAGENT achieved comparable results to the best statistical methods (UCB) (**c, f**) while informing the predictive model more (**d, g**), and sometimes discovering higher affinity molecules within the budget.

there must exist a consistent projection of its outputs onto valid candidates. However, no compact structured rule exists to describe valid candidates more efficiently than simply enumerating them. Prior work also shows that LLMs struggle to generate valid SMILES for larger molecules [Narayanan et al., 2025], and the outputs from off-the-shelf models tend to bear little to no resemblance to any valid candidate. Thus, this task is better suited to in-context filtering, where the model selects from a curated candidate list, which is feasible because the number of valid molecules is small enough for the LLMs to practically process.

Conversely, the protein task is characterized by (i) simple representations (amino acid sequences) and (ii) a dense and highly structured search space. Off-the-shelf models are easily informed of the search space structure, can reliably produce 4-amino acid sequences, and (almost) all combinations of amino

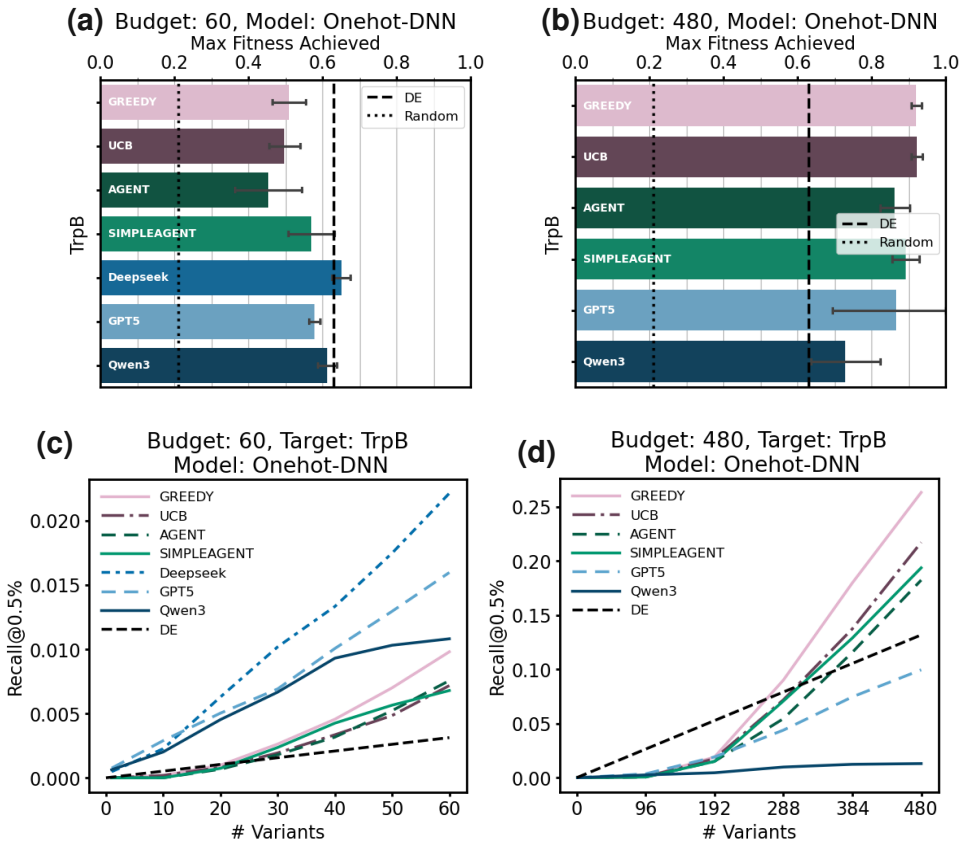


Figure A-3: Complement to Figure 4. Max fitness and recall for campaign targeting a different protein motif (TrpB) with a budget of 60 and a batch size of 10 motifs (**a,c**) and with a budget of 480 and a batch size of 96 motifs (**b,d**). For TrpB, sequences closer to the WT generally yielded higher fitness than for GB1 (Figure 4.a), leading to DE outperforming almost all fitness methods. All off-the-shelf models informed by the WT thereby performed well on this target. Qwen3 primarily produced informed DE-like behavior, whereas GPT-5 explored more advanced patterns. Again, the off-the-shelf models failed to scale to the longer campaigns, while the agents performed similarly or slightly worse than the statistical methods. DeepSeek-R1 failed to complete the longer campaign.

acids are biologically viable. Using in-context filtering in the protein task would be impractical, as that would require parsing all 20^4 sequences every cycle.

B Implementation Details

B.1 Bayesian Optimization Loop

Each Bayesian Optimization (BO) campaign was defined by three parameters: the `initial size`, i.e., the number of initial observations selected either randomly or from a cluster; the `batch size`, i.e., the number of samples chosen or generated per iteration; and the total `budget`, i.e., the number of samples drawn throughout the campaign. Each cycle proceeds as follows:

1. Train a `predictive model` on the accumulated observations.
2. Use the `predictive model` to predict the fitness of all candidates in the search space.
3. Select candidates for observation based on `Acquisition function` applied to the predictions and uncertainty estimations.
4. Assign true labels to selected candidates (mimicking observations from wet lab experiment) and add them to the set of accumulated observations.

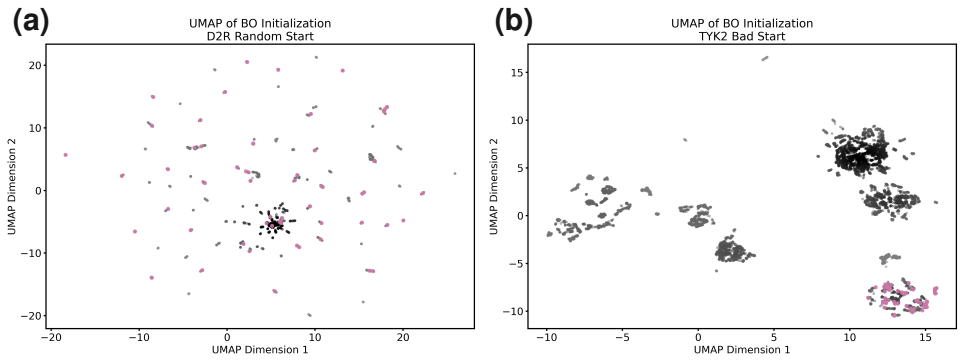


Figure A-4: UMAP projection of selected start-batch molecules (**Pink**) relative to all available molecules in the database (**Gray**). **a:** Random start on the D2R target. **b:** poor (suboptimal) start on the TYK2 target.

B.2 LLM-enabled approaches

We investigated the use of Large Language Models (LLMs) as either the predictive model or the acquisition function within the BO loop. Three strategies were explored:

- **Off-the-shelf models:** LLM models applied directly without fine-tuning or agents. For the molecular optimization task, the LLM replaced only the acquisition function and had access to the predictive model's predictions and uncertainty estimations. For the protein optimization task, the LLM replaced both the predictive model and the acquisition function, operating in a fully standalone manner.
- **Agentic workflows:** A system in which an LLM orchestrates tools and a multi-step reasoning process to sample the candidate space. Here, the agentic workflow replaced the acquisition function.
- **Fine-tuned models:** LLMs fine-tuned with synthetic datasets to perform optimization tasks. Fine-tuned models were evaluated only for the protein optimization task.

We employed several publicly available LLM families using their respective APIs: Llama-4-Maverick [AI, 2025], DeepSeek-R1 [DeepSeek-AI, 2025], and Qwen3-32B [QwenTeam, 2025] (via the Lambda API); GPT-5 [OpenAI, 2025] (via the OpenAI API); Claude models [Anthropic, 2024] (via the Anthropic API); and locally hosted Qwen models (via Hugging Face).

B.3 Molecular optimization task

We employed publicly available binding affinity datasets assembled by Gorantla et al. [2024]; **TYK2**: 9,997 congeneric molecules with aminopyrimidine core scaffold, derived from RBFE calculations [Thompson et al., 2022], **USP7**: 4,535 diverse scaffolds from ChEMBL, contains multiple assay minima, **D2R**: 2,502 molecule subset of ACNet dataset [Zhang et al., 2023], shows high activity cliff, **Mpro**: COVID Moonshot project data (665 compounds). Mpro was discarded due to insufficient data, and USP7 due to irregularities. UMAP of the chemical search space in Figure A-4 showed that TYK2 dataset consisted of few large molecular clusters, making the search "too easy" when randomly initialized. By contrast, the D2R dataset consisted of more diverse scaffolds. Accordingly, we used TYK2 for experiments with "poor starts" (initialization restricted to a single cluster), while using D2R for random starts. "Poor starts" were generated by clustering a UMAP projection of the chemical space into 10 clusters with K-means and selecting all initial observations from a single cluster. For completeness, we also report the results for TYK2 with random starts and D2R with poor starts for a subset of methods.

A Gaussian process regressor (GP) with a Tanimoto similarity kernel (radius=4, nbits=4096) trained for 500 epochs (`learning rate=0.001, Adam, gpytorch 1.14, rdkit 2025.3.3`) was used as a predictive model. The GP provided both predictions and predictive uncertainties. GREEDY acquisition sampled the top candidates by prediction. UCB sampled the top candidates from the score

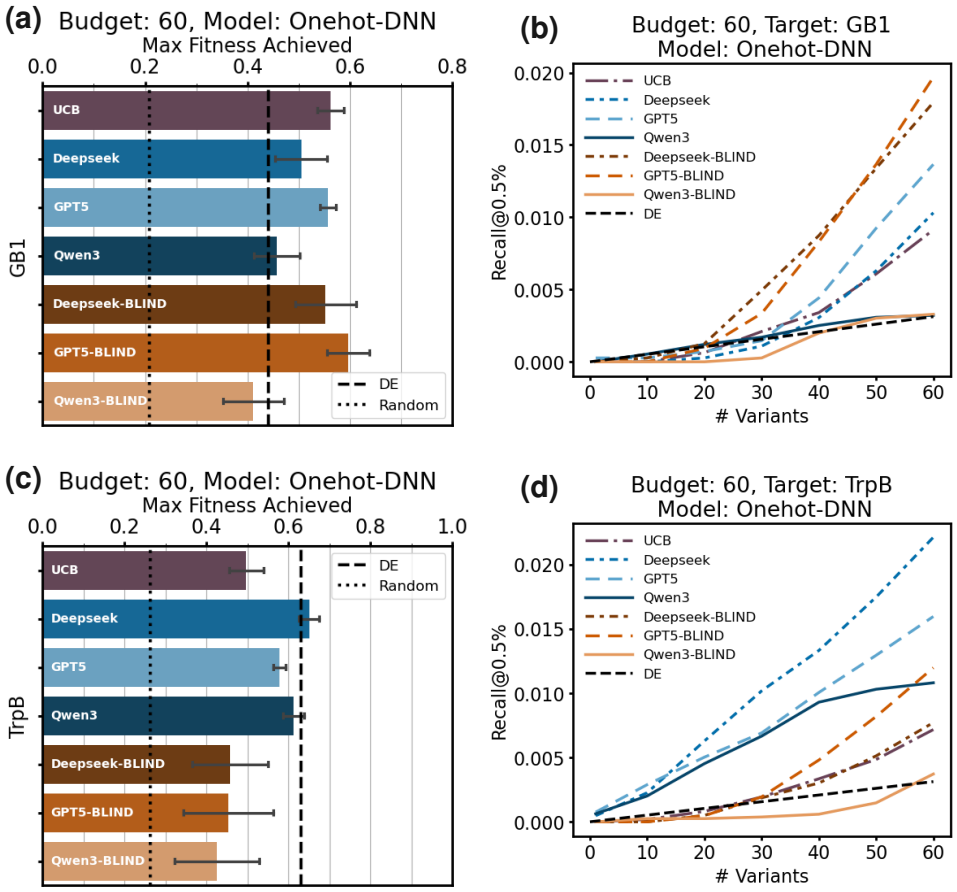


Figure A-5: Performance of off-the-shelf models initialized without any context to the target protein (**-BLIND**) on the GB1 task (**a-b**), and the TrpB task (**c-d**), compared to a subset of other models. The GPT-5 and DeepSeek-R1 blind models performed similarly or significantly better than all statistical methods on the recall metric. Notably, GPT-5-BLIND once found 30 sequences in the top 0.5 % of the search space within the 60 sequence budget; the highest number of sequences found by any statistical method (across 50 random seeds) was 23). The unexpected performance on the GB1 task was largely a coincidence. Both GPT-5 and DeepSeek-R1 often started the search with the likely deliberately conservative guess "AAAA" which happened to be highly successful in the GB1 task. The "AAAA" guess is less effective in the TrpB task. Qwen3 performed point-wise mutations, achieving performance similar to directed evolution (**DE**). Error bars mark standard deviation.

$s = \mu + \beta^{1/2}\sigma$, using $\beta = 4$. The Thompson sampler selected the top-k candidates from 100 posterior draws of the predictive model with low-rank approximation (rank 600). Expected improvement sampled the candidates that maximizes $\mathbb{E}[\max(0, f(x) - f^*)] = (\mu - f^\star)\,\Phi(Z) + \sigma\,\phi(Z), \quad Z = \frac{\mu - f^\star}{\sigma}$, where f^* is the largest candidate found so far, Φ and ϕ are the CDF and PDF respectively. For each sample, ϵ-GREEDY drew the top candidate by prediction, with a $\epsilon = 0.05$ probability of instead drawing from the uniform distribution.

Off-the-shelf LLMs were evaluated as replacements for the acquisition function. We tested Qwen3-32B (temperature 0.6), DeepSeek-R1, and GPT-5 (medium reasoning effort). For analysis of attention effects, we also employed Llama-maverick-4 without reasoning. At each cycle, the LLM was prompted to select candidates from a table containing index, SMILES, GP model prediction, and GP model confidence (uncertainty). For models with extended context windows (e.g., DeepSeek-R1), prompts additionally included all compounds and their labels from the accumulated observation (Prompt B.7.1). For models with more limited context (Qwen3, GPT-5), accumulated observations were compressed into a compact string summary (Prompt B.7.1).

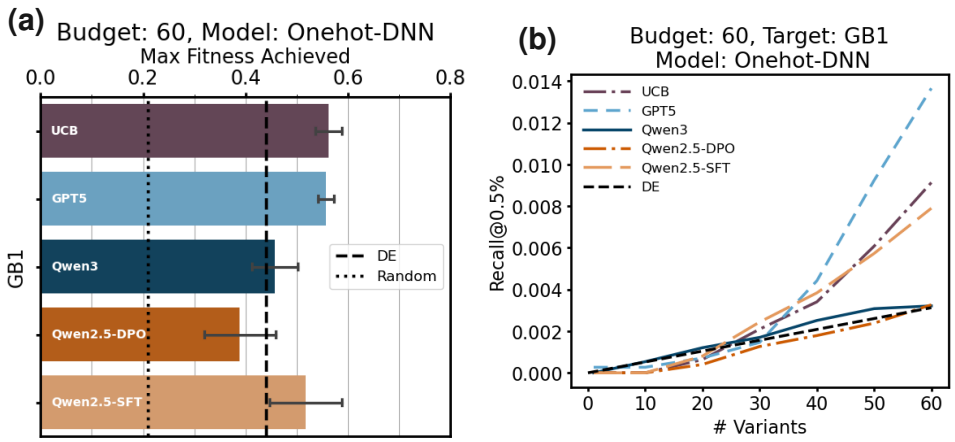


Figure A-6: **GB1**. SFT and DPO models compared to a subset of other methods. The SFT model significantly outperformed the reasoning model Qwen3, executing 30 times faster and achieving performance competitive with UCB on the recall metric. Error bars mark standard deviation.

Candidate sets were chunked for tractability: ∼400 candidates per chunk for Qwen3 and GPT-5, and ∼1,200 for DeepSeek-R1 and Llama-4. Each model was asked to select $\max(\text{batch size}, \text{chunk size}/n_{\text{chunks}})$ candidates from each chunk, with a final selection made from the pooled chunk outputs. Considerable prompt engineering and fallback strategies were required to make this approach both reliable and moderately efficient. Some strategies included providing in-context examples and providing examples of what not to do.

Agentic workflows were implemented in LangGraph 0.5.0, consisting of three main LLM components organized in the following structure:

Strategist (Qwen3-32B, temperature 0.6): A high-level planning node with reasoning enabled. It received detailed information about the current BO stage, prior cycles, and accumulated observations (Prompt B.7.1). It generated an arbitrary number of complementary filter strategies for candidate selection. (`generate_strategy` in Figure A-11). The only difference between **AGENT** and **SIMPLEAGENT** is that **AGENT** was provided with a SMILES table (or protein sequences in the protein task) appended to each cycle summary (see Prompt B.7.1 for an example) while the strategist of **SIMPLEAGENT** did not get explicit SMILES information.

Implementers (Claude Sonnet 3.5-20241022, temperature 0): A sequence of execution nodes without reasoning capability (`implement_strategy` in Figure A-11). Each implementer received one strategy from the strategist and applied database filtering using prediction thresholds, UCB weighting, SMARTS pattern matching (parsed into SQL-style boolean expressions with AND/OR/NOT and parentheses), and/or Tanimoto similarity to accumulated observations or the current batch. Candidates selected by an implementer were removed from the database to avoid duplication. If, after all strategies were applied, the batch size remained unfilled due to overly restrictive filters, a final implementer node was created to complete the batch using a less stringent strategy.

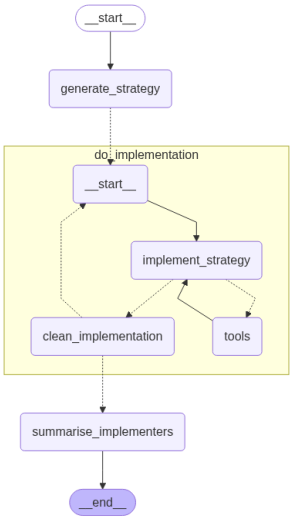


Figure A-11: LLM workflow structure showing strategist-implementer-summary node hierarchy with the tool integration loop.

Summary node (Claude Sonnet 3.5-20241022, temperature 0): A final summarization step that processed the outputs of all implementer nodes, generating a cycle-level report containing rationales, decisions, and identified issues. This report was then provided to the strategist in the following cycle, allowing it to refine its strategy formulation.

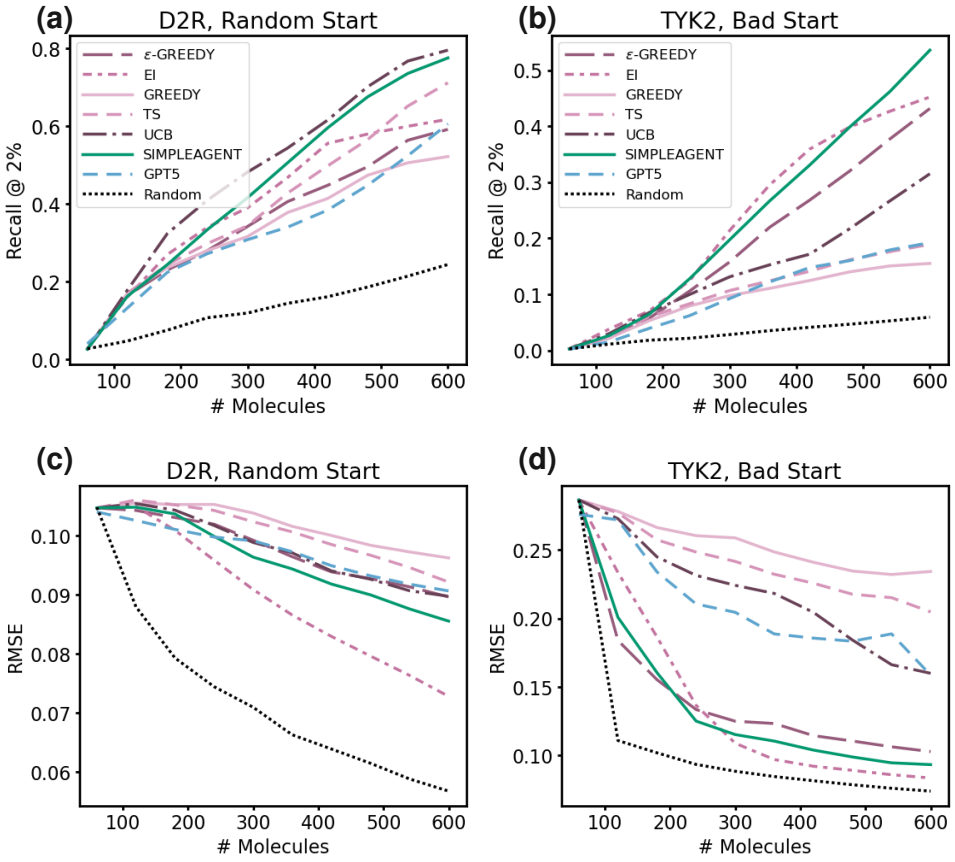


Figure A-7: Performance of statistical acquisition functions (**GREEDY**, **UCB** = upper confidence bound, **TS** = Thompson sampling, **EI** = Expected improvement, **ϵ-GREEDY**) on the molecular BO task. (**GPT-5**) and the best agentic workflow (**SIMPLEAGENT**) added for reference.

The yellow-highlighted region in Figure A-11 represents the core implementation loop where strategies were executed through available tools. To probe whether the strategist's performance was constrained by tool availability in the molecular domain, we tested a variant where all tool descriptions were removed and the system was instead told: "your strategies will be implemented by experienced chemists". The only commonly requested filtering operation not available to our agent was k-means clustering of the data in the dimensionality-reduced projections.

B.4 Protein optimization task

We built on the codebase and dataset introduced by Yang et al. [2025]. The dataset contains fitness scores for nearly all possible mutations across four-residue motifs in two proteins: GB1 and TrpB. In the original work, performance was benchmarked as the highest discovered fitness across four cycles (`batch size=96, starting size=96, budget=480`). We adopted the Deep Neural Network (DNN) ensemble predictive model with one-hot amino acid encoding, which was the setting optimized by Yang et al. [2025]. For comparison, we also tested a Gaussian Process (GP) regressor with ESM2 embeddings [Lin et al., 2023], previously shown to be the weakest-performing baseline. Acquisition functions (Greedy, Thompson Sampling, UCB with $\beta = 4$) were left identical to the original setup. For additional implementation details, see Yang et al. [2025]. EI and ϵ-GREEDY were implemented identically to the molecular task. We ran seven campaign variants: **Small-GB1** and **Small-TrpB** assessed performance on short campaigns using an optimized predictive model-encoding combination. **Small-Bad** evaluated the performance of acquisition functions when paired with a suboptimal predictive model-encoding combination, whereas **Big-*** assessed performance on the larger campaign with an optimal setting. Performance was measured both by the highest discovered

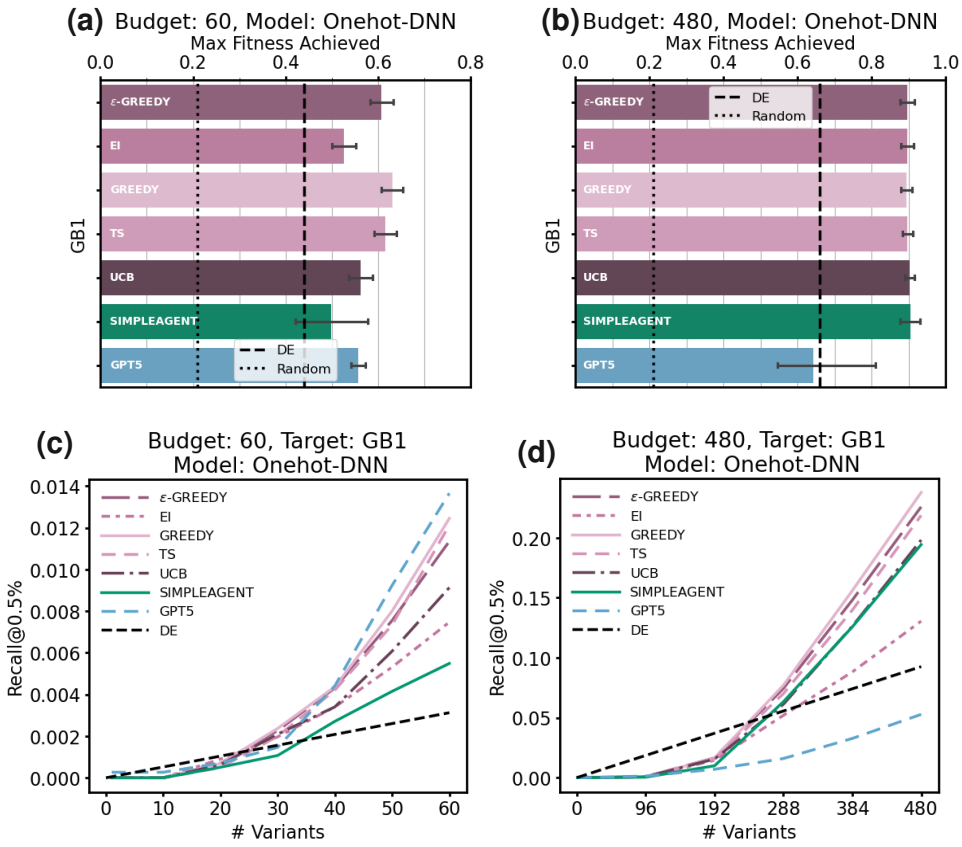


Figure A-8: Performance of statistical acquisition functions (**GREEDY**, **UCB** = upper confidence bound, **TS** = Thompson sampling, **EI** = Expected improvement, **ϵ-GREEDY**) on the protein BO task. (**GPT-5**) and the best agentic workflow (**SIMPLEAGENT**) added for reference.

	Target	Model	Initial size	Batch size	Budget
Small-GB1	GB1	DNN-one-hot	10	10	60
Small-Bad	GB1	GP-ESM2	10	10	60
Big-GB1	GB1	DNN-one-hot	96	96	480
Small-TrpB	TrpB	DNN-one-hot	10	10	60
Big-TrpB	TrpB	DNN-one-hot	96	96	480

Table A-4: BO scenarios

fitness and by recall at the 0.5% cutoff, which excludes WT sequences from the threshold. We avoided the 2% cutoff used in Domain 1, which saturates too quickly in this task and becomes less informative.

Off-the-shelf models were deployed as fully standalone generative models, independent of predictive models. Each campaign began with an introduction to the task, the WT sequence, and a short description of the target motif and protein (Prompt B.7.1). For example, the GB1 background was provided as:

"The target is a four-site epistatic region (wildtype: V39, D40, G41, V54, fitness 0.1) of the 56-residue protein G domain B1 (GB1), an immunoglobulin-binding domain from Streptococcal bacteria. These sites account for a majority of the most strongly epistatic interactions in GB1 and span a fitness landscape of 160,000 variants. Variants were assessed for IgG-Fc binding using mRNA display and high-throughput sequencing".

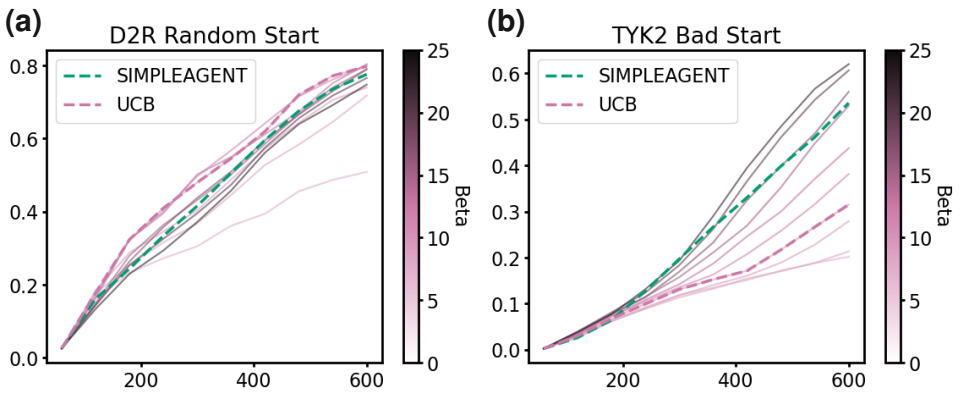


Figure A-9: Sensitivity analysis of UCB acquisition methods for different values of hyperparameter β, $score = \mu + \sigma\sqrt{\beta}$ in the D2R campaign with random start (**a**) and the TYK2 campaign with poor start (**b**). The dashed line (**UCB**) marks the setting used in the main text ($\beta = 4$). With poor initialization, performance was highly sensitive to β, whereas with random initialization, UCB remained robust. The SIMPLEAGENT adaptively selects β at each call.

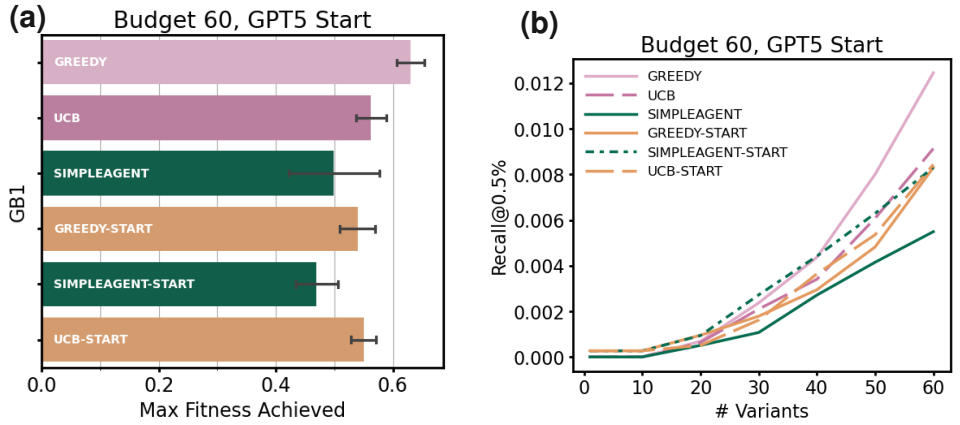


Figure A-10: Performance of two statistical methods (**GREEDY**, **UCB**) and an agentic workflow (**SIMPLEAGENT**) when initialized with either random starting points (N=10) or GPT-5-selected starting points (**-START**, N=5). GPT-5's chosen starting points are not consistently more informative than random sampling.

Notably, for the TrpB target, we replaced the protein name "TrpB" with the generic term "protein", since Qwen3 tended to overproduce tryptophan residues (Trp/W) when exposed to the word "TrpB", completely deteriorating the performance. This oversampling of Trp residues does not stem from an effort to improve fitness, as Trp occurs 584 times less frequently in the top 0.5 % sequences compared to the overall distribution.

After each cycle, the model was prompted with the validation results mimicking wet lab observations: "*The validation experiment in cycle {current_cycle} is finished. These are the results: {validated_results}.*" along with a system reminder about output format and the top 10 sequences discovered so far (Prompt B.7.1). Models were asked to generate 50% overhead per batch and rank outputs by importance, mitigating issues of overlapping generations or invalid/unlabeled sequences. Invalid generations were corrected via re-prompting (e.g., "the following were invalid for experiments") and asking to generate new sequences. To handle context limits and reduce formatting failures, the reasoning LLMs were wrapped in a LangGraph context, which validated outputs, prompted corrections, summarized conversations during longer runs, and extracted sequences from prompts (Figure A-12). **BLIND** models were implemented with the same setup but without any

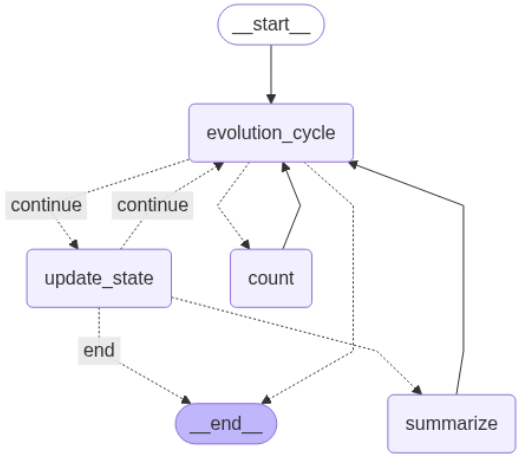


Figure A-12: Standalone LLM workflow: the **evolution node** prompted the LLM to analyze and generate new variations for the next cycle, the **count node** gave the LLM feedback if generations are duplicates or invalid, the **update state node** labeled the proposed variations, and a simple **summary node** triggered if the message stream approaches context limit. The loop ended after a set number of cycles.

background information of the protein target. A full example campaign using GPT-5 BLIND is shown in Prompt B.7.2

The agentic workflows were implemented identically to the workflows in the molecular task, but with a different set of tools. The tools included sorting by Hamming similarity, applying regex filters, and model prediction- and UCB-based selections. As in the molecular task, we tested both **AGENT** (receiving a full table of accumulated observations and their corresponding fitnesses) and **SIMPLEAGENT** (receiving only summarized data). Interestingly, **SIMPLEAGENT** greatly outperformed **AGENT**.

B.5 Fine-Tuning

We investigated whether a non-reasoning LLM could be fine-tuned to better perform in optimization tasks. Prior work [Wang et al., 2025] showed that training a non-reasoning LLM on acquisitions generated by a statistical method in an artificial setting can improve its Bayesian behavior. To this end, we constructed 14 synthetic datasets using fitness functions that take ESM2 embeddings of motifs as input. Synthetic fitness values were assigned using Algorithm 1, designed to generate diverse fitness distributions. Steps 9–13 (optional) were applied to increase dataset complexity. Although parameters were not extensively tuned, they were selected to induce variance across distributions while preserving similarity to biological fitness landscapes, in which the majority of samples exhibit zero fitness. For each dataset, we ran 50 campaigns (`batch size=10, initial size=10, budget=400`) using one-hot embeddings, a DNN ensemble predictive model, and TS as the acquisition function. From each trajectory, we sampled 15 time points t with probability $P(X = t) = t^{-1} \sum_{i=1}^{40} i^{-1}, t \in [1, 40]$. At each time point, we constructed a prompt consisting of the prior trajectory (sequences with assigned fitness), with the next batch used as the ground-truth optimal selection. To increase variance between prompts, sequence positions were permuted, and the order of sequences was shuffled in-context. This step was crucial for successful training.

We trained models using both Supervised Fine-Tuning (SFT), where the ground truth serves as a label, and Direct Preference Optimization (DPO) [Rafailov et al., 2024], where the ground truth is contrasted with a randomly generated string from the set of accumulated observations.

SFT was run with TRL [Von Werra et al., 2020] SFTTrainer (trl 0.19.1, transformers 4.54.1, flash-attn 2.7.4, vllm 0.10.0) with `per_device_batch_size=8`, `gradient_accumulation_steps=2`, `dtype=bf16`, `gradient_checkpointing=True`, `max_grad_norm=1`, `weight_decay=0.1`, `learning_rate=5e-6`, `warmup_ratio=0.05`, `lr_schedular_type=cosine` for three epochs, and with early-stopping based on the validation loss. Training was applied only to responses, not prompts.

DPO was run with `per_device_batch_size=1`, `gradient_accumulation_steps=16`, `dtype=bf16`, `gradient_checkpointing=True`, `max_grad_norm=1`, `weight_decay=0.1`, `learning_rate=5e-7`, `beta=0.9`, `lr_schedular_type=cosine` for two epochs. Strong regularization was crucial for maintaining stability during training, keeping it close to the reference model.

While the SFT-fine-tuned model performed strongly, the DPO-fine-tuned model performed poorly in this setting. It is likely that the positive and negative examples (mutated strings vs. random strings) were too similar, making discrimination difficult. Sequential training (SFT followed by DPO) also deteriorated performance compared to SFT alone.

Algorithm 1 Synthetic Fitness Generation

Input: Sequences $\mathcal{S}$
Tokenize: $\mathbf{E} \leftarrow \mathbf{ESM2}(\mathcal{S})$,$\mathbf{E} \in \mathbb{R}^{n \times d}$
Sample binary mask: $\mathbf{m} \sim \text{Bernoulli}(p = 0.2)^d$
Sample log-weights: $\log \mathbf{w} \sim \mathcal{N}(\mu = -6.5,\, \sigma^2 = 1.5)$
Apply mask: $\mathbf{w} \leftarrow \mathbf{m} \odot \exp(\log \mathbf{w})$
Compute fitness: $\mathbf{f} \leftarrow \mathbf{E} \cdot \mathbf{w}$
Add noise: $\log \mathbf{f} \leftarrow \mathcal{N}(\log \mathbf{f},\, 0.01^2)$
Normalize: $\mathbf{f} \leftarrow (\mathbf{f} - \min f)/(\max f - \min f)$
Set 10% randomly to zero: $f_i \leftarrow 0$ for $i \in \mathcal{I} \subset \{1, \ldots, n\}$
Set near-zero values to zero: $f_i \leftarrow 0$ if $f_i < 0.001$
while $\text{Quantile}_{0.995}(\mathbf{f}) > 0.5$ **do**
    $\mathbf{f}_i \leftarrow \mathbf{f}_i^{1.5}$
end while
return Dataset $\{(s_i, f_i)\}_{i=1}^n$

We used the Qwen2.5-7B-Instruct model from HuggingFace. The off-the-shelf model underperformed random selection, likely due to formatting issues and its greedy behavior. Training on four NVIDIA A100-SXM4-80GB GPUs with Deepspeed [Rasley et al., 2020] required 12 min for 0.5B models and 2 h for 7B models.

The fine-tuned model was configured to generate 10 sequences per batch and run at inference with a temperature of 0.7. If invalid sequences were produced, the in-context data was reshuffled and the model re-prompted; if duplicates occurred, the temperature was incremented by 0.05 for that cycle. Running a small campaign (`batch size=10, starting size=10, budget=60`) required 11.9 s without any optimization, compared to 15.6 s for one-hot + GP + Thompson (Proper posterior sampling of posterior approximated with 1000 Fourier features) on a single NVIDIA A100-SXM4-40GB.

B.6 Statistical Analysis

Statistical intervals in figures are standard deviations in mean. Intervals in text refer to the 95% confidence interval of the population mean, computed using bootstrapping for each comparison.

Bootstrapping. Significance as assessed at the 95% confidence level, defined as non-overlapping confidence intervals. The threshold was defined as 95% confidence in separation between (1) a statistical method and all LLM-enabled methods, or (2) an LLM-enabled method and all statistical methods.

For agent–statistical and off-the-shelf-statistical comparisons in the molecular task, bootstrapping was performed using shared random seeds to account for the strong covariance between trajectories starting from the same point. Sample sizes were N=10 for agent–statistical comparisons, N=10 for DeepSeek-statistical comparisons, and N=5 for Qwen3 and GPT-5–statistical comparisons. In each bootstrap replicate, N random seeds were drawn with replacement, and the means of the corresponding trajectories were compared.

For off-the-shelf-statistical comparison in the protein task, off-the-shelf models are independent of starting points; therefore, bootstrapping was performed directly on the trajectories. For each replicate,

N=50 trajectories were sampled with replacement from the statistical methods and N=10 trajectories from the off-the-shelf LLMs. Group means were then compared between methods.

All bootstrap iterations were repeated 10,000 times.

Average ranking. The average rank of a model on a given set of conditions and metrics compared to a set of other models was estimated by bootstrapping and directly averaging across all samples. Ties were solved by assigning all models with the same score the lower rank (e.g., scores$\uparrow$ $[10, 11, 11, 13] \rightarrow$ ranks $[4, 2, 2, 1]$).

B.7 Prompts and Responses

B.7.1 Prompts

Prompt 1: In-context acquisition prompt. "Oracle" here refers to the predictive model

```
You are selecting ligands for validation in an active learning campaign for
    protein {protein}.

**OVERALL CAMPAIGN OBJECTIVE:** Maximize the total number of top high-
    affinity ligands in the training data at the end of the campaign. A top
    high-affinity ligand is in the top 2% of all ligand candidates.

**Campaign Status:**
- Cycle: {cycle}/{total_cycles}

**Historical Data:**
Used to train the Gaussian Process Regression oracle, sorted by RBFE:
<validated_ligands>
{self._compact_df(labeled_data, index=False)}
</validated_ligands>

3. **Candidates** [SMILES, predicted RBFE, std]:
Randomly ordered.
<candidates>
{self._compact_df(chunk)}
</candidates>

**Your task:**
- Reason about the chemical space, the candidates, the reliability of the
    oracle model, and the overall campaign goal. Describe the rationale
    behind your selection.
- Select exactly {budget} candidates to move on with (validate and add to
    training data for next cycle). Output format:
<selected_indices>
[index1, index2, ...]
</selected_indices>
```

Prompt 2: Prompt to summarize chemical space

```
You are a chemoinformatic expert working in a team of chemoinformatic experts
     selecting ligands for validation in an active learning campaign protein
    {protein}.
You have access to a set of validated ligands, and your junior colleague has
    access to a set of thousands of candidates to validate.
**Your task:**
- Create a concise summary of the validated chemical space that preserves all
     structural features, patterns, and structure-activity relationships,
- Your junior colleague will use the summary to select candidates from his
    large list. Include any crucial instructions for the selection process.
    Your junior collegue will not have access to the training data.
- Consider number of validated ligands, and number of cycles left. What are
    the most important features to preserve?

**Campaign Status:**
- Current cycle: {cycle}
- Total cycles: {total_cycles}
- Oracle model: {oracle_name}
- Overall Campaign Goal: Maximize number of validated high-affinity ligands
    in the training data at the end of the campaign
```

```
**Validated Training Data:**
{len(training_df)} ligands with measured RBFE values:
<validated_ligands>
{self._compact_df(training_df)}
</validated_ligands>

**Required output:**
<training_data_description>
Comprehensive but concise description of the training data to be used by the
    junior colleague that fully encompasses the chemical space, and the
    active learning context.
</training_data_description>

**Output Format:**
<training_data_description>
[Training data description that will be passed to the junior colleague]
</training_data_description>
Include exactly one instance of <training_data_description> tag in your
    response
```

Response 2: Example of Qwen3 summary of chemical space

```
<training_data_description>
The validated ligand set (120 molecules) centers on **bipartite scaffolds**
    linking aromatic/heteroaromatic rings via **amide/urea linkers**, with
    key features:
1. **Core Architecture**: Two aromatic rings (often phenyl/pyridine) bridged
    by amide/urea (-CONH-/-NHC(O)N-) linkers. Common substructures include *
    ccnc(c)*-*C(=O)N*-*cccccX* (X = Cl/F/Br/O).
2. **Electron-Withdrawing Groups**: Para/meta halogens (Cl, F, Br) dominate
    on terminal aromatics (e.g., *c(c(Cl)F)*, *c(c(Cl)Cl)*). Dual
    fluorination (*c(c(F)F)*) improves RBFE (>9.5 kcal/mol).
3. **Hydrogen Bonding Motifs**: Urea/urethane linkers (*N=C(O)N*) enhance
    activity (top 10\% RBFE >= 9.5 kcal/mol). Terminal nitroso/amino groups
    (*-N(H)C=O*, *-NH2*) reduce activity (<8.5 kcal/mol).
4. **Aliphatic Chains**: N-linked methyl/ethyl groups (*-CH3*, *-CH2CH2-* in
    *N(C)C(=O)N*) stabilize topologies. Longer chains (*-CH2CH2CH2-*)
    correlate with reduced affinity (\Delta RBFE $\simeq$ -0.3-0.5 kcal/mol
    per CH2).
5. **Steric Constraints**: Bulky substituents (e.g., *Br*, *N(C)(C)C*)
    tolerate modestly (RBFE >9 kcal/mol if paired with strong ÏĂ-stacking
    partners); avoid *I* or *NO2* due to poor performance (median RBFE ~7
    kcal/mol).
6. **Orthogonal Features**: Ether linkers (*-O-CH2-*) improve water
    solubility without sacrificing activity (best performers retain Cl/F on
    aromatic rings).

**Selection Priorities for Next Cycles**:
- Fill gaps in **meta-fluorinated aromatics** and **mixed X/Y halogen pairs**
     (e.g., *c(c(Cl)F)* vs *c(c(F)F)*).
- Test **shortened/cyclic spacers** (replace *CH2CH2* with *CH2* or *CH2OCH2
    *) to optimize linker stiffness.
- Explore **triazine/triazole cores** as rigid alternatives to *ccnc(c)*
    scaffolds.
- Avoid redundant modifications (e.g., additional Br/I substitutions or
    longer aliphatic chains).
- Validate **heteroaryl replacements** (thienyl, furyl) adjacent to amide
    linkers.
```

```
Candidates must align with existing SAR while introducing minimal scaffold
    novelty. Favor analogs expanding **fluorination patterns** and **
    alkylation balance** across the bipartite framework.
</training_data_description>
```

Response 3: Example of GPT5 summary of chemical space

```
<training_data_description>
Context and goal
- You will down-select candidates (no access to training data) for TYK2
    across 2 remaining cycles (currently 7/9). Oracle = Gaussian Process
    regressor (uncertainty-aware). Objective = maximize the number of
    validated high-affinity ligands in the training set by end of cycle 9.

What has worked (validated chemical space; 418 ligands; top RBFE up to ~12.2)
- Two privileged chemotypes dominate. Keep these intact unless explicitly
    exploring:
  1) "Triad" anilide series (majority, highest RBFE):
     Left: 6-membered diazine/pyridine core, most often a 4-aminopyrimidine/
    pyrimidinyl-pyridine, typically 5-alkyl (Me >> Et ~= iPr) and para-amine
    bearing a small basic side chain (see "Left-side chains" below).
     Middle: 2-aminopyridine (or 2-aminopyridinyl) most often 5-fluoro-
    substituted.
     Linker: secondary anilide/benzamide (--NH--C(=O)--Ar) with the amide NH
    on the "middle ring" side (do not reverse the amide).
     Right (distal aryl): dihalo phenyl; the recurring best pattern is ortho-
    chloro + meta/para-fluoro; di-F is also strong; o-Cl/o-Cl is acceptable;
    o-Br variants can be good. Occasional phenol tolerated but generally not
    top.
     Summary motif (abstract): [5-Me-(amino)pyrimidine/pyridyl]--NH--(5-F-2-
    aminopyridyl)--NH--C(=O)--[o-Cl, m/p-F phenyl].
  2) "Morpholine-tail heteroaryl amide" series (secondary cluster; many
    9.8--11.7):
     Acyl aryl similar to above (often F/Cl patterns).
     Hinge-facing heteroaryl is more N-rich (e.g., Nc--cnn(c)--) bearing a
    pendant N-morpholine or N-(2-oxa-5-azabicyclic) tertiary amine (--
    N4CCOCC4). Keep the tertiary amine and the heteroaryl arrangement
    together; they are synergistic.

Left-side chains (key to high RBFE, in order of priority)
- Small, conformationally constrained cations:
  A) Azetidine (--N1CCC1--) on the left ring amine: repeatedly among the top
    (~11--12.2).
  B) Cyclopropyl-bearing secondary amines (--NC1CC1--) and small hydroxyalkyl
     secondaries (--N--CH(CH3)--CH2OH or --N--CH2--CH2OH) are strong.
  C) 4-hydroxyazetidine and other compact "N,O" motifs are good compromises.
  D) Bulkier or more flexible amines (long chains, multiple heteroatoms) and
    heavily N-methylated cations tend to underperform vs A--C.

Substituent SAR you can apply directly
- Distal aryl (acyl side):
  Priority 1: o-Cl + m- or p-F (best-in-class recurring motif).
  Priority 2: o-Cl + p-Cl or o-Cl + p-Br (slightly lower on average, but
    still strong).
  Priority 3: di-F (good); di-Cl (acceptable; generally slightly lower).
  Deprioritize: strong donors (p-OMe/phenoxy), dense polar patterns, or
    removal of ortho-halogen.
- Middle ring:
  2-aminopyridine with 5-F consistently outperforms unsubstituted analogs;
    removal of the 5-F is generally a drop.
```

```
- Left heteroaryl core:
  4-aminopyrimidine/related diazines with a 5-Me substituent are most common
    in top tier. Me > Et ~= iPr; larger alkyls drop off.
- Amide orientation:
  Preserve the "middle ring --NH--C(=O)-- distal aryl" orientation. Amide
    reversal/urea/sulfonamide replacements are not supported by top data.
- Halogen count:
  Retain at least one ortho-halogen on the distal aryl (Cl >> F for ortho);
    removing it usually costs 0.5--1.0 RBFE.
- H-bond pattern:
  Keep exactly one amide NH donor and two ring nitrogens across the triad;
    adding extra strong donors/acceptors generally hurts.

What to avoid (seen repeatedly lower)
- Reversing the amide, swapping the amide for urea/sulfonamide, or breaking
    the triad topology.
- Overly flexible or bulky cationic tails; long aliphatic chains; multiple
    extra heteroatoms on the left side chain.
- Distal aryl without ortho-halogen; heavy electron-donation (e.g., para-OMe)
     unless paired with the optimal ortho-halogen pattern.
- Extra ring nitrogens that overpolarize the middle ring (e.g., turning the 2-
    aminopyridine into a more N-rich diazine) typically drops RBFE.
- Nitrile decorations and phenols appear in mid-tier but rarely at the top;
    deprioritize unless needed for diversity.

Concrete selection rules for your candidate list
1) Hard filters (must pass)
   - Contains the triad or morpholine chemotype:
     a) Triad: left 6-membered N-containing ring (prefer 4-aminopyrimidine)
    linked to 2-aminopyridine (prefer 5-F), linked via --NH--C(=O)-- to a
    distal aryl.
     b) Morpholine series: aryl amide paired with Nc--cnn(c)--(N-morpholine)
    heteroaryl tail.
   - Distal aryl includes an ortho-halogen (prefer Cl; Br acceptable; F only
    if paired with another F/Cl).
   - Exactly one amide linker (secondary amide; not reversed).
2) Prioritization scoring (apply additively; pick highest-scoring per cluster
    )
   +3: Left side chain is azetidine (--N1CCC1--).
   +2: Left side chain is cyclopropyl secondary amine or small hydroxyalkyl
    secondary amine.
   +2: Middle ring = 2-aminopyridine with 5-F.
   +2: Distal aryl = o-Cl + p-F (or o-Cl + m-F).
   +1: Distal aryl = o-Cl + p-Cl (or o-Cl + p-Br) or di-F.
   +1: Left ring = 4-aminopyrimidine with 5-Me.
   -2: Reversed amide, missing ortho-halogen, or bulky/flexible tertiary
    amines on the left.
   -1: Extra strong donors (phenols, guanidines) or >1 extra heteroatom
    beyond the patterns above.
3) Diversity protocol (to hedge with GP uncertainty; 2 cycles left)
   - Cycle 7->8 (this round): 70% exploitation + 30% exploration
     Exploitation: Pick close analogs within the triad keeping the features
    above; systematically vary ONE position at a time:
       R-distal: swap p-F <-> p-Cl <-> p-Br; move F m<->p; di-F vs o-Cl/p-F.
       R-middle: 5-F -> 5-H (limited set) to quantify the drop; 2-
    aminopyridine <-> 2-aminopyrimidine (very small set).
       R-left chain: azetidine <-> hydroxyethyl secondary amine <->
    cyclopropyl; \pmsingle N-methyl; \pmone OH.
     Exploration: 3--4 small clusters outside the densest neighborhood:
       - Replace distal phenyl with 2-pyridyl retaining o-Cl (on 3-position
    relative to amide) and a para-F where possible.
       - Swap 5-Me on the left ring to Et or iPr (single change) to probe
    steric tolerance.
```

```
      - Morpholine-series pickups that match the same distal aryl patterns (
    o-Cl with F).
   - Cycle 8->9 (final): 85--90% exploitation
     - Lock in the top 2--3 R-group patterns discovered; make nearest-
    neighbor analogs (1 change only; keep ortho-Cl and 5-F).
     - Reserve 10--15% for "safety" variants: same distal aryl but two best
    left-chain options; same left chain but two best distal patterns.

Quick triage checklist (yes/no)
- Amide present and correctly oriented? Yes -> proceed; No -> drop.
- Distal aryl has ortho-Cl (or Br)? Yes -> proceed; No -> only keep if di-F
    and everything else optimal.
- Middle ring is 2-aminopyridine with 5-F? Yes -> strong; No -> keep a few
    for diversity only.
- Left chain is compact cation (azetidine/cyclopropyl/hydroxyethyl)? Yes ->
    strong; No -> deprioritize.
- Left heteroaryl is 4-aminopyrimidine (5-Me)? Yes -> strong; No -> keep
    minority for exploration.

If you must choose one "safe" pattern to maximize hits now
- Pick candidates matching: 4-aminopyrimidine (5-Me) --NH-- 5-F-2-
    aminopyridine --NH--C(=O)-- 2,4-dihalo phenyl (o-Cl + p-F), with a left
    azetidine side chain. Then explore \pm(p-F->p-Cl), \pm(azetidine->
    cyclopropyl/hydroxyethyl), \pm(o-Cl/p-F->o-Cl/m-F).

What not to change this late
- Do not remove the ortho-halogen on the distal aryl or the 5-F on the middle
     ring across the bulk of your picks.
- Do not invert the amide or replace it with urea/sulfonamide.
- Do not add bulky solubilizing tails; prefer small, rigid cations.

Submission format suggestion (to help us review quickly)
- Group selected candidates into small R-group matrices:
  Matrix 1 (triad, fixed left = azetidine; vary distal p-substituent): {o-Cl/
    p-F, o-Cl/p-Cl, o-Cl/p-Br, o-Cl/m-F}.
  Matrix 2 (triad, fixed distal = o-Cl/p-F; vary left chain): {azetidine,
    cyclopropyl secondary amine, hydroxyethyl secondary amine, 4-
    hydroxyazetidine}.
  Matrix 3 (morpholine series, fixed distal = o-Cl/p-F; vary heteroaryl N
    pattern minimally).
- Within each matrix, avoid >1 simultaneous change relative to the
    prototypical pattern.

Rationale link to RBFE observations
- The highest RBFE ligands co-occur with: compact left cations (especially
    azetidine), 5-F on the central 2-aminopyridine, and ortho-Cl plus F on
    the distal phenyl; Br is tolerated. When these features are removed,
    drops of ~0.3--1.0 RBFE are common; when combined, RBFE >=10.5--12 is
    frequent. The morpholine-tail chemotype is a reliable secondary path with
     the same distal aryl patterns.

Use this guide to pre-filter and prioritize; then, if you have a model to
    score uncertainty, allocate the exploration quota to the edges defined
    above.
</training_data_description>
```

Prompt 3: Strategist prompt. "Oracle" here refers to the predictive model

```
You are a senior medicinal chemist and expert in active learning performing
    an active learning drug development campaign. Your task is to evaluate
```

```
    the current state of the campaign and delegate candidate selection to
    junior colleagues who have strong computational skills but limited
    chemical intuition. They need clear, specific instructions to filter the
    molecular database effectively.

**CONTEXT AND DATA EXPLANATION:**
- SMILES: Chemical structure representation of molecules
- affinity: Validated binding affinity from experiments (RBFE score, higher
    is better)
- predictions: Oracle model's predicted affinity for unlabeled candidates
- std: Oracle model's standard deviation in predictions
- max_per_cycle: Maximum affinity achieved in each cycle's selected compounds
- mean_per_cycle: Average affinity of selected compounds in each cycle
- oracle RMSE: Root Mean Square Error between oracle predictions and
    validated affinities for selected compounds at a given cycle (measurement
     of the oracle's performance on selected compounds, not nessecairily the
    oracle's performance on the entire database. If we heavily favor
    exploitation one cycle, RMSE will appear lower because we are selecting
    candidates that the model is confident about, but not necessarily the
    best candidates.)

- Completed cycles: {cycles_completed}
- Total cycles: {total_cycles}
- Budget this cycle: {batch_size}
- Oracle model: Gaussian Process Regression
- Measure of affinity: RBFE (higher is better)
**Overall Campaign Goal**
From our ligand database, discover as many compounds as possible that would
    rank in the top 2% by binding affinity. Success is measured by recall: of
     all the truly high-affinity ligands in the database, how many can you
    find within your experimental budget?

**Previous cycles performance**
- Max affinity in selected candidates (starting from cycle 0, should increase
     over time): {max_per_cycle}
- Mean performance in selected candidates (starting from cycle 0, should
    increase over time): {mean_per_cycle}
- Oracle RMSE on selected candidates a given cycle (starting from cycle 0,
    not included in training data that cycle, should decrease over time): {
    oracle_rmse_per_cycle}


**Previous Cycles Summaries:**
All Validated Candidates Used for Training or Current Cycle Oracle. Here
    follows strategies implemented at previous cycles, and their respective
    performance. Use this to inform your new strategies in the context of the
     current cycle and campaign state, dont copy them.
<past_cycles_data>
{past_cycles_data}
</past_cycles_data>

**ANALYSIS:**
Analyze the current state of the campaign.

**CHEMICAL ANALYSIS REQUIREMENTS:**
Identify chemical regions for exploitation and exploration:
- Be specific: use chemical knowledge and terminology to describe patterns
    and potential binding motifs. Look for both simple and complex (multiple
    substructures) patterns.
- Similar targets: [Structural patterns in the training data that are
    promising]
- Potential targets: [knowledge about protein {protein} that could be used to
     guide the selection, hypothesis testing motifs]
```

```
- Coverage gap: "Training data covers a [small/medium/large] portion of the
    relevant feature space"
- What targets were explored in previous cycles? How did they perform?

**Active Learning Status:**
Identify if exploration or exploitation is more important:
- How are we doing towards the overall campaign goal?
- Passed progress: [is the current training data promising or is the oracle
    plateauing? Are we stuck in a local optimum?]
- Exploration vs exploitation: [can we afford to explore more or should we
    exploit more?]
- Is the oracle better or worse that our intuition?

**TASK:**
Design the optimal selection strategy for this cycle that your junior
    colleagues can execute independently. Your junior colleagues don't know
    about each other's work. This may be a single protocol or multiple
    complementary protocols, depending on what's most appropriate for the
    current campaign state. Design hypotheses for testing and informing
    coming cycles when suitable.
Provide only actionable filtering instructions - no explanations or chemical
    rationale needed.

YOUR JUNIOR COLLEAGUES' CAPABILITIES
They can filter candidates using:
- Computational approaches: Predictions
- Upper confidence bound (UCB): Predictions + beta * std, given beta
- Chemical approaches: Substructures (SMARTS or substructure names) and
    similarity metrics
- Diversity approaches: Tanimoto similarity metrics (pairwise between
    selected candidates or to training data)
- Hybrid approaches: Combining the above

EACH PROTOCOL SHOULD SPECIFY:
Exact number of candidates to select
Precise filtering criteria with numerical thresholds
Clear chemical and/or computational constraints

Ensure total candidate count across all protocols equals {batch_size}. Output
     your analysis and strategies as soon as you are confident in your
    selection.
```

Response 4: Example of cycle summary. "Oracle" here refers to the predictive model

```
Cycle 1:
# AL Campaign Acquisition Summary

## Implementation
All selection strategies were successfully implemented, yielding 60 total
    compounds:
- 25 compounds balancing exploitation/exploration (pred >7.5, UCB beta=1.5)
- 20 exploration-focused compounds (UCB beta=3.0, high uncertainty)
- 10 structurally diverse compounds (Tanimoto <0.3)
- 5 compounds exploiting known SAR (fluorinated pyrimidines with halogens)

No substructure filters failed or needed to be released. Database contained
    all requested chemical patterns.

## Campaign Impact
```

```
The selections maintain a strategic 42/58 split between exploitation and
    exploration:
- Exploitation (30 compounds): Targeting high-affinity regions through
    predicted values >7.5 and known SAR patterns of fluorinated pyrimidines
- Exploration (30 compounds): Probing uncertain regions through high UCB
    scores and ensuring structural diversity

This balanced approach is appropriate for the early campaign stage with
    limited training data (60 compounds). The selections will expand chemical
     space coverage while still leveraging emerging SAR patterns around
    halogenated aromatics. The varying uncertainty thresholds (beta=1.5 vs
    beta=3.0) provide a gradient of exploration intensity, helping to
    systematically improve model reliability.

Strategy 1: Select 25 candidates with predictions >7.5 and UCB (beta=1.5) to
    balance high-affinity exploitation with moderate uncertainty
Number final selections: 25
RMSE: 0.76
Mean: 9.60
Max: 10.63

Strategy 2: Select 20 candidates with UCB (beta=3.0) to prioritize
    exploration of uncertain regions (std >0.45)
Number final selections: 20
RMSE: 1.14
Mean: 8.92
Max: 10.87

Strategy 3: Select 10 candidates with Tanimoto similarity <0.3 to the
    training set to enforce diversity
Number final selections: 10
RMSE: 1.56
Mean: 8.29
Max: 9.57

Strategy 4: Select 5 candidates containing fluorinated pyrimidine core [n]1
    cccc([F])c1 and [Cl] or [Br] substitutions
Number final selections: 5
RMSE: 0.75
Mean: 10.04
Max: 10.33
Selected candidates from cycle 1 with oracle predictions and std:
SMILES|affinity|oracle_prediction|oracle_std
c1cc(c(cc1N)C(=O)Nc2cc(ncc2F)NC(=O)C3CC3)Cl|8.96|9.84|0.173
CNC(=O)Nc1cc(ccn1)NC(=O)c2c(cccc2Cl)F|9.95|9.68|0.252
c1cc(c(cc1N)Cl)C(=O)Nc2cc(ncc2F)NC(=O)C3CC3F|10.5|9.7|0.178
...
```

Prompt 4: Standalone Model Start Prompt

```
You are an expert protein engineer with deep chemical intuition leading a
    directed evolution campaign. Apply rigorous chemical principles to
    discover high-fitness variants within your experimental budget.

**CAMPAIGN OVERVIEW:**
- Target: Four-site region of {self.protein}
- Background: {background}
- Total Budget: {self.total_budget} experimental validations
- Batch Size: {self.batch_size} sequences per round
- Number of cycles: {self.max_cycles}
```

```
- Sequence Length: 4 amino acids

This is the START of your campaign. You have no prior data.

**STRATEGIC APPROACH:**
At every cycle, you'll receive previous results. Select {3*self.batch_size //
     2} untested sequences using chemical logic and hypothesis testing.

**OUTPUT REQUIREMENTS:**
1. **Chemical Reasoning**: Evaluate the past cycles and explain your
    mechanistic hypotheses and chemical logic for the next cycle. Add
    reasoning within <reasoning> tags.
3. **Priority Ranking**: Sort by priority with chemical justification
4. **Final List**: End with {3*self.batch_size//2} ranked sequences for next
    cycle validation for buffer, {self.batch_size} of which will be validated
     in the next cycle.

Design sequences that test well-reasoned chemical hypotheses. Balance
    thorough exploration with chemical principles to maximize discovery
    potential through systematic, chemistry-guided experimentation.
```

Prompt 5: Standalone Model System Message

```
You are an expert protein engineer with designing and controlling a directed
    evolution campaign. You must strategically select protein sequences to
    maximize fitness discovery within your experimental budget at the end of
    the campaign.

Current cycle {state["current_cycle"]} of {state["max_cycles"]}.
Select {3*self.batch_size//2} sequences, optimal for the campaign goal, to
    validate in next cycle.
Of these, {self.batch_size} will be validated in the next cycle, depending on
     experimental feasibility.
Sort you selection by priority.

Highest performing sequences so far:
{self.validated_results.sort_values(by='fitness', ascending=False).head(10).
    round(3).to_string(index=False)}

**Output format**: End your analysis with a list of {3*self.batch_size//2}
    sequences (STRICTLY 4-letter codes like 'AMHG', 'QPEI').
```

B.7.2 Response examples

Prompt 6: GPT5 TrpB BLIND full campaign example.

```
You are an expert protein engineer with deep chemical intuition leading a
    directed evolution campaign. Apply rigorous chemical principles to
    discover high-fitness variants within your experimental budget.

**CAMPAIGN OVERVIEW:**-
- Target: Four-site region of unknown
- Background: No background available, we enter the campaign blind.
- Total Budget: 60 experimental validations
- Batch Size: 10 sequences per round
- Number of cycles: 6
- Sequence Length: 4 amino acids
```

```
This is the START of your campaign. You have no prior data.

**STRATEGIC APPROACH:**
At every cycle, you'll receive previous results. Select 15 untested sequences
     using chemical logic and hypothesis testing.

**OUTPUT REQUIREMENTS:**
1. **Chemical Reasoning**: Evaluate the past cycles and explain your
    mechanistic hypotheses and chemical logic for the next cycle.
3. **Priority Ranking**: Sort by priority with chemical justification
4. **Final List**: End with 15 ranked sequences for next cycle validation for
     buffer, 10 of which will be validated in the next cycle.
4. **Output format**: List 15 sequences (STRICTLY 4-letter codes like 'AMHG',
     'QPEI') at the end of your response.

Design sequences that test well-reasoned chemical hypotheses. Balance
    thorough exploration with chemical principles to maximize discovery
    potential through systematic, chemistry-guided experimentation.
```

Response 5: GPT5 response

```
Chemical reasoning and strategy for Cycle 1 (blind start)
- With no prior data, we should maximize information gain per experiment by
    spanning orthogonal physicochemical axes: net charge (positive vs
    negative), hydrophobicity (aliphatic vs aromatic), polar H-bonding (Ser/
    Thr vs Asn/Gln), conformational rigidity/flexibility (Pro/Gly), redox/
    metal coordination (Cys/His, D-x-D), and secondary-structure propensities
     (Ala/Leu/Ile/Val).
- The set below implements a fractional-factorial-like sweep across these
    axes, so early results can reveal main effects and simple interactions by
     comparing closely related contrasts (e.g., cationic vs anionic; aromatic
     vs aliphatic; HXH vs D-x-D; rigid PxxP vs flexible PGPG).
- Prioritization favors maximal diversity first, then close contrasts to
    refine mechanistic hypotheses.

Priority ranking with brief chemical justification
1) KRRR - Strongly cationic, Arg/Lys-rich for electrostatic binding to acidic
    /negatively charged targets (e.g., nucleic acids or acidic protein
    patches).
2) DEED - Strongly anionic, acidic cluster probe; contrasts directly with
    KRRR to test electrostatic preference.
3) WFWY - Aromatic-rich, tests pi-pi/pi-cation interactions and hydrophobic
    packing using mixed F/W/Y.
4) LVIL - Aliphatic hydrophobic core motif; contrasts WFWY to separate
    aromatic-specific effects from generic hydrophobics.
5) STST - Ser/Thr H-bond network and turn/helix edge propensity; probes polar
     OH-mediated recognition.
6) HEGH - HExH-like histidine motif; tests metal coordination and pH-
    sensitive binding via His.
7) DADE - D-x-D acidic motif; canonical divalent metal-binding/chelation
    contrast to HEGH.
8) CAGC - Cys pair (CxxC) enables redox/disulfide or soft-metal coordination;
     probes thiol chemistry and structural constraint.
9) PGPG - Alternating Pro/Gly for high turn propensity and conformational
    flexibility; tests if tight turns are favored.
10) PAPP - PxxP motif to enforce rigidity and SH3-like motif geometry;
    contrasts PGPG for flexibility vs rigidity.
11) RKRK - Alternating cationic pattern to test charge spacing vs clustered
    cationic (KRRR).
```

```
12) QNQQ - Amide-rich polar H-bonding without charge; probes polar
    recognition distinct from Ser/Thr chemistry.
13) FWLW - Alternating aromatic/aliphatic to test interleaved hydrophobic/
    aromatic patterns vs clustered aromatics (WFWY).
14) AAAA - Helix-prone neutral baseline; a clean reference for main-effect
    comparisons across positions.
15) LYKS - Mixed motif (hydrophobic-aromatic-basic-polar) to sample
    heterogeneous interfaces and potential pi-cation (Y/K) interplay.

Final list of 15 sequences for next-cycle validation (top 10 prioritized
    above):
KRRR, DEED, WFWY, LVIL, STST, HEGH, DADE, CAGC, PGPG, PAPP, RKRK, QNQQ, FWLW,
     AAAA, LYKS
```

```
The validation experiment in cycle 0 is finished. These are the results:
sequence  fitness
    AAAA    0.074
    KRRR    0.045
    LVIL    0.020
    DADE    0.016
    WFWY    0.013
    STST    0.013
    PAPP    0.010
    RKRK    0.002
    DEED    0.000
    HEGH    0.000
```

```
Cycle 1 analysis and hypotheses from cycle 0 data
- Small/neutral wins: AAAA is top (0.074), suggesting minimal side-chain bulk
     and strong helix propensity (Ala) are beneficial. This points toward a
    preference for compact, low-steric side chains and possibly an alpha-
    helical segment.
- Cationic helps, especially Arg-rich clusters: KRRR is second-best (0.045),
    while alternating cationic RKRK is poor (0.002). This implies clustered
    or context-dependent cationic interactions are beneficial; Arg appears
    superior to Lys (guanidinium vs ammonium).
- Hydrophobic tolerated but suboptimal: LVIL (0.020) and WFWY (0.013)
    indicate hydrophobic packing/aromatics help but are inferior to A-rich.
- Acidic clusters and His motifs are disfavored: DEED (0.000), HEGH (0.000);
    DADE (0.016) shows that isolated/alternating acidity with small residues
    can be tolerated, but excess negative charge is detrimental.
- Polar OH acceptable but not optimal: STST (0.013) suggests limited benefit
    from Ser/Thr networks.

Working hypotheses for next designs
1) Ala-rich background is optimal; adding limited positive charge (preferably
     Arg) may boost fitness if positioned correctly. Test single-position Arg
     and Lys scans on AAAA.
2) Arg likely outperforms Lys; quantify this across positions.
3) Two Arg in an Ala background may be better than one but worse than fully
    charged; test spacing effects (AARR).
4) Small/polar (Ser) or smallest (Gly) substitutions might be position-
    sensitive; edge vs internal effects may differ.
5) Keep exploration limited for acidic/aromatic features; include one
    âĂIJsentinelâĂİ acidic variant to measure penalty of an isolated negative
    charge in an Ala carpet.
```

```
Priority ranking with justification
1) RAAA - Single Arg at position 1 on AAAA; probes N-terminal placement of
    cationic charge with minimal steric penalty.
2) ARAA - Single Arg at position 2; tests internal Arg placement effect.
[...]
```

```
The validation experiment in cycle 1 is finished. These are the results:
sequence  fitness
    AAAG    0.279
    AAAS    0.221
[...]
```

```
Cycle 2 analysis and mechanistic hypotheses
- Strong terminal preferences emerged:
  - Position 4: Gly and Ser are highly beneficial (AAAG 0.279 > AAAS 0.221 >>
     AAAA 0.074). This is signature of a C-terminal cap preference for small/
    flexible or H-bonding residues that relieve steric clash and/or stabilize
     a local turn/helix cap (G > S; next likely T, then Asn; bulkier polar
    like Gln expected weaker).
  - Position 1: Gly improves fitness (GAAA 0.119 > AAAA 0.074). N-terminal
    flexibility or reduced steric bulk appears favorable.
- Charges are generally detrimental, especially Arg at any position (RAAA,
    ARAA, AAAR all ~0-0.01). Lys is tolerated only internally (AAKA âĽĹ AAAA)
    , but not beneficial; Lys at pos1 is harmful (KAAA 0.018). Acidic
    residues are strongly disfavored (AADA 0.000). Hydrophobics/aromatics
    also underperform.
- Therefore, the productive design space is: small/non-bulky residues,
    especially G/S at termini; possibly T/N as alternative C-caps; internal
    positions likely prefer A/S/G with caution about excessive flexibility
    from multiple Gly.

Design strategy for Cycle 2
- Test additivity/epistasis by combining the best terminal features: G/S at
    pos4 with G (and S) at pos1.
- Expand C-terminal cap chemistry: Thr and Asn; include Gln as a bulkier
    polar comparator.
- Probe whether added flexibility internally helps or hurts: single Gly scans
     at positions 2 and 3; single Ser scans at positions 2 and 3.
- Limit risky motifs (charge, aromatics, strong hydrophobics), reserving
    budget for the productive small/polar space.

Priority ranking with chemical justification
1) GAAG - Combines the two strongest single effects (G1 + G4); tests near-
    maximal flexibility at termini; highest expected fitness.
2) GAAS - G1 + S4; probes if H-bond cap at 4 with flexible N-terminus rivals
    G1/G4.
[...]
```

```
<O>The validation experiment in cycle 2 is finished. These are the results:
sequence  fitness
    GAAS    0.315
```

```
    SAAG    0.238
[...]
```

```
Cycle 3 analysis and mechanistic hypotheses
- Clear SAR emerged: termini prefer very small/polar residues, centers prefer
     Ala.
  - C-terminal cap: S4 and G4 are strongly beneficial (AAAS 0.221; AAAG
    0.279). Thr4 is highly deleterious (AAAT 0.008). Asn4 is roughly neutral/
    slightly worse than baseline (AAAN 0.075).
  - N-terminal: G1 improves baseline (GAAA 0.119). S1 is promising when
    combined with favorable C-caps (SAAG 0.238; SAAS 0.186), but S1 alone is
    unmeasured.
  - Best family identified: X A A Y with X âĹĹ {G,S}, Y âĹĹ {G,S}; strongest
    observed GAAS (0.315) > SAAG (0.238) > GAAG (0.202) > SAAS (0.186),
    indicating terminal epistasis (S4 best with G1; G4 best with S1).
  - Internal positions penalize flexibility/polarity: introducing Gly/Ser at
    2 or 3 reduces fitness (AGAA 0.071; AAGA 0.023; AASA 0.036), though a
    strong C-cap can partially rescue (AAGS 0.107). Centers prefer Ala.

Working hypotheses to guide next designs
1) The global maximum likely resides near XAA Y with X,Y âĹĹ {G,S}; we will
    probe whether modest, chemically plausible deviations at positions 2-3 (
    single Ser/Gly) can fine-tune the optimum under favorable terminal pairs.
2) N-capping alternatives: Pro or Asn at position 1 might sometimes stabilize
     local turns/caps; test them in the context of strong C-caps (S4/G4).
3) Cys as a small polarizable C-cap surrogate for Ser/Gly: test C4 with/
    without G1 to map tolerance. Include one Gln4 sentinel to bound amide-cap
     performance.
4) Complete missing single-mutant baselines: SAAA (S1 alone) and ASAA (S at
    pos2) for positional sensitivity.

Priority ranking with justification
[...]
```

```
The validation experiment in cycle 3 is finished. These are the results:
sequence  fitness
    GPAS    0.341
[...]
```

```
Cycle 4 strategy and chemical reasoning
- Confirmed terminal epistasis: Position 4 strongly prefers S or G but with
    context. G1 pairs best with S4 (GAAS 0.315 > GAAG 0.202); S1 pairs best
    with G4 (SGAG 0.277 > SAAS 0.186). Amide caps (N/Q at 4) and Thr4 are
    poor; Cys4 is mediocre. Keep 4 âĹĹ {S,G}.
- Internal positions favor Ala; introducing Gly or Ser at 2 or 3 depresses
    fitness even under favorable termini (e.g., GGAS 0.100, GAGS 0.093, GSAS
    0.085). Key exception: Proline at position 2 gives a real boost on the G1
    /S4 ridge (GPAS 0.341 > GAAS 0.315), likely via local turn/helix capping
    and preorganization without over-flexibilizing the core.
- N-terminus: G1 and S1 are both good in the right context; S1-G4 and G1-S4
    look like complementary optima. Nonpolar/hydrophobic or charged
    substitutions were uniformly harmful earlier.
```

```
Hypotheses to test now
1) Proline-at-2 as a transferable enhancer: Move P2 to the S1/G4 ridge to
    create SPAG; also test if P2 improves the weaker G1/G4 ridge (GPAG) and
    the weaker S1/S4 ridge (SPAS). Expect SPAG to be competitive with the
    current best.
2) Is P best kept at 2? Shift P to position 3 under strong terminals to
    quantify penalty vs benefit (GAPS, GAPG, SAPG), anticipating lower but
    mapping robustness.
3) N-cap alternatives: Thr1 can N-cap in helices; test TPAG/TPAS as
    conservative variants against S1 with both C-termini.
4) How much turn bias is too much? Introduce P2-G3 tight turn (GPGS, SPGS)
    and double-Pro core (GPPS, GPPG) as stress tests; expect reduced but
    informative.
5) Is G1 essential for P2 benefit? Compare APAS vs GPAS baseline (already
    known) to quantify the necessity of the G1 N-cap in the P2 motif.
6) Internal polarity under P2: Test GPSS to confirm S3 remains penalizing
    even with P2.

Priority ranking with justification
1) SPAG - Aligns with S1/G4 synergy; adds P2 which helped on G1/S4. Highest
    likelihood to meet or exceed current best.
[...]
```

```
The validation experiment in cycle 4 is finished. These are the results:
sequence  fitness
    TPAG    0.292
[...]
```

```
Cycle 5 chemical reasoning and goals
- Clear terminal epistasis remains: position 4 prefers S or G, but the
    preference flips with N-terminus and with the presence of Pro at position
     2.
  - With G1 and P2 present, S4 is superior (GPAS 0.341 > GPAG 0.225).
  - With T1 and P2, G4 is far superior (TPAG 0.292 >> TPAS 0.113).
  - With S1 and P2, evidence from G3 sentinels suggests G4 > S4 (SPGG 0.172 >
     SPGS 0.071); we now test this in the optimal A3 background.
- P2 is a key enhancer when used with optimal terminals; P3 is deleterious.
    Internal polar/flexible residues at 2-3 are generally harmful unless they
     are precisely Pro2 or Gly2 in the S1/G4 context (SGAG 0.277 > SAAG
    0.238).
- A3 appears optimal; G3/S3 consistently reduce fitness, but we will probe
    whether a small hydrophobe (V3) can be tolerated/improve preorganization
    under strong termini/P2.
- N-terminal identity strongly modulates the best C-cap when P2 is present;
    we will systematically map X1 in {S,A,N} across Y4 in {G,S} with P2 fixed
     to locate the global maximum near the GPAS/SPAG/TPAG ridge.

Priority ranking with justification
1) SPAG - Tests S1/P2 with G4; predicted to be high due to S1-G4 synergy and
    P2 enhancement (cf. SGAG 0.277; P2 likely boosts further).
[...]
```

```
The validation experiment in cycle 5 is finished. These are the results:
sequence  fitness
    GPVS    0.495
[...]
```