BIOLOGICAL CONTEXT DOES NOT RELIABLY IMPROVE LLM-GUIDED EXPERIMENTAL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous labs increasingly deploy large language models (LLMs) as closed-loop biology optimizers that propose a design, read the outcome, then propose again. This is partly because of an assumption that biological knowledge helps an LLM choose better designs than Bayesian optimization, which has no access to biological context. We test that assumption on 45 biology tasks constructed from 2,485 experimental rows across 341 published papers, scored by oracles fitted to those rows, each run over 30 iterations under two prompt conditions. The domain-aware condition provides the published parameter names, units, and labels, and the domain-agnostic renames them so they are not biologically meaningful, with search space and feedback identical. Across 19 LLMs, 14 sit at or below a GP-UCB reference in *both* conditions, and biological context costs further ground, domain-agnostic outperforming domain-aware on 58.7% of tasks by trajectory and 61.9% by endpoint. The failure is specific to biology, reversing on education tasks built the same way. Domain-aware prompting tends to help models propose better initial designs, while making them respond worse to feedback. Part of the reason is anchoring, with models keeping the conventional categorical choice even when the oracle scores a different (published best) value higher. Post-training Llama 3.1 8B to frontier-level raw optimization does not transfer to biology-aware optimization improvements. Biology-aware optimization therefore looks like a separate capability that autonomous laboratory loops should target directly.

1 INTRODUCTION

Many foundational tasks in biology are optimizations. Expressing a protein, engineering a cell line, tuning an assay, and measuring a molecular interaction all reduce to finding the conditions under which a measured quantity is highest, and design-of-experiments methods have been standard for them for decades (Mandenius & Brundin, 2008; Walsh et al., 2025). They are hard because a single round is expensive, taking days to weeks of bench time, reagents and a technician, and using only the feedback the previous round returned. This is the design-build-test-learn cycle of synthetic biology and metabolic engineering (Carbonell et al., 2018), where the scarce resource is rounds of feedback rather than wells per round. Traditionally, Bayesian optimization has often been used as the optimizer in this loop (Srinivas et al., 2010; Shahriari et al., 2016).

LLMs are increasingly being explored as the optimizer in this autonomous-laboratory loop. Such LLM-driven systems are already running, and closed-loop autonomous experimentation has become an active research program (Dvivedi et al., 2026; Boiko et al., 2023). GPT-5 cut cell-free protein synthesis component cost by 40% in Ginkgo's automated laboratory (Smith et al., 2026), and OpenAI and Red Queen Bio report a 79× improvement to a cloning protocol (OpenAI, 2025). In these settings, the LLM proposes values for a fixed set of parameters, the assay returns one number, and the LLM proposes again. The Red Queen Bio evaluation is run this way, on fixed prompts that isolate the model's own contribution (OpenAI, 2025).

At least three properties are assumed to give an LLM an advantage over Bayesian optimization:

1. LLMs have access to biological context and prior scientific knowledge (Roohani et al., 2025; MacKnight et al., 2025).

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

2. LLMs may be stronger raw optimizers, able to use non-scalar feedback that classical optimizers cannot represent (Yang et al., 2024; Kobalczyk et al., 2026).
3. LLMs can call tools, including biology-specific ones, to build a richer understanding of the space agentically (Roohani et al., 2025; Smith et al., 2026; Boiko et al., 2023).

Evaluating the three assumptions separately is worthwhile, since weaknesses in biological knowledge, raw optimization, and tool use each lead to different training targets. An overall gain also does not rule out one aspect making performance worse. If biological context does not help, the case for an LLM in these loops rests on (2) and (3) instead.

We focus explicitly on assumption (1), which has not been tested in isolation on biology tasks. Prior work either gives the model biological context, tools, and retrieval at once, leaves the domain content in the prompt, or isolates context outside biology, where it helps. None of these separates what biological context contributes (§2).

Our scope is to isolate this first dimension, not to evaluate a full autonomous lab. Our domain-aware condition provides the published parameter names, units, and categorical labels, as deployed autonomous-laboratory loops do, and the model proposes a design from what it knows about the biology, receives one scalar score, and proposes again. *Parameterized* optimization fixes the design space in advance, so the action space is identical across the two conditions and the prompt is the only thing that varies.

Section 3 sets out the tasks, the two conditions, and the metrics, Section 4 reports the outcome, and Section 5 analyzes the failure.

Contributions.

1. **A controlled test of whether biological context helps LLM-guided biology optimization.** To our knowledge this is the first isolated test of that assumption on biology tasks. Across 45 biology tasks and 19 LLMs, biological context improves neither the trajectory nor the endpoint, and neither condition beats Bayesian optimization (§4).
2. **An analysis of why biological context does not improve LLM-guided biology optimization.** Domain-aware prompting proposes the best published parameter value more often at iteration 1 and less often by iteration 30, and it sticks with the most common published value after feedback favors another (§5.2, §5.3).
3. **Post-training evidence that biology-aware optimization is a separate capability from raw optimization.** Training Llama 3.1 8B to Bayesian-optimization parity under domain-agnostic prompts transfers none of the improvement to domain-aware ones (§6).

2 Related work

Biological knowledge. Many papers assume, implicitly or explicitly, that an LLM's biological knowledge helps, without testing it in isolation. BioDiscoveryAgent credits it for outperforming Bayesian optimization by 21% (Roohani et al., 2025), and MacKnight et al. (2025) credit pretrained knowledge for matching Bayesian optimization on six reaction datasets. The same assumption underlies autonomous-laboratory systems (Boiko et al., 2023; Dvivedi et al., 2026; Smith et al., 2026) and in-context molecular optimization (Nguyen & Grover, 2025). Two papers isolate context-awareness directly, both outside biology. LLAMBO removes the textual descriptions from an LLM-driven Bayesian optimizer and finds its predictive accuracy and calibration degrade (Liu et al., 2024), confirmed independently (Rychert et al., 2025). SemanticOpt finds richer context improves performance on chemistry, recipe, and hyperparameter benchmarks (Meindl et al., 2025). Both report that context helps, and neither varies the biological content of the prompt with the task, the oracle and the budget held fixed.

Raw optimization. Comparisons against classical methods keep the domain content in the prompt, so they measure an LLM with its priors intact rather than raw optimization. Gupta et al. (2025) find classical methods ahead of LLM agents on gene perturbation and molecular property tasks, and report that permuting the observed outcomes leaves LLM performance unchanged. Huang et al.

(2024) find LLMs underperform on numerical problems. The domain-agnostic arm of this paper is a more faithful test of that condition.

Tool use. LAB-Bench scores models with no tools (Laurent et al., 2024). Given web search and code execution, agents exceed human performance on two of its tasks, and a trained Llama 3.1 8B agent reaches frontier accuracy at lower cost (Narayanan et al., 2024), so tools help. The deployments bundle tool use with biological knowledge (Smith et al., 2026; Roohani et al., 2025; Boiko et al., 2023), so the gains reported there cannot be assigned to either.

3 Benchmark and Experimental Setup

We isolate biological context with two prompt conditions over identical tasks, and add GP-UCB as a reference that receives no context at all. The tasks are 45 biology optimization problems constructed from published experiments, each with an oracle fitted to the extracted rows, and each run over 30 iterations in which the model proposes a design and receives one scalar score in return. An oracle is only a useful stand-in where it is aligned with the biology, so we ground our analysis in the experimental literature.

3.1 Domain-aware prompts, domain-agnostic prompts, and the GP-UCB reference

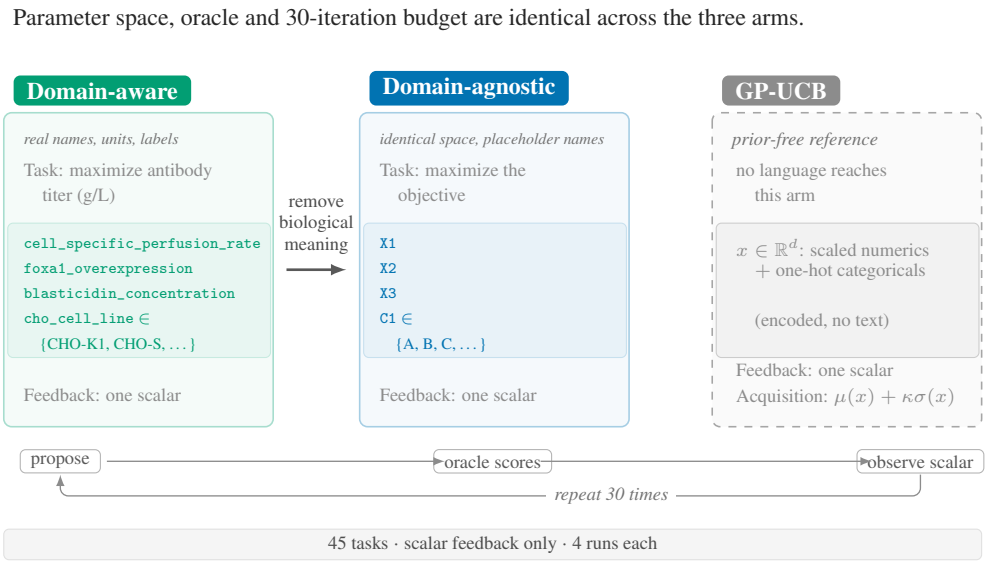

Figure 1: **The two prompt conditions differ only in whether the parameter names are biologically meaningful.** One held-out task, CHO antibody titer, is rendered three ways. Domain-aware (teal) shows the published parameter names, categorical labels, and units. Domain-agnostic (cobalt) replaces them with `X1`/`C1` placeholders over the identical space. GP-UCB receives no biological context at all, only a scaled and one-hot encoded vector. The parameter space, oracle, budget, and feedback channel are identical across the three, and every task-condition pair is run four times. Prompt templates in §A.2.

Figure 1 shows how we isolate biological context. The *domain-aware* arm receives the published parameter names, units, and categorical labels. The *domain-agnostic* arm is the same task with the same bounds and oracle, names replaced by `X1`/`C1` placeholders, removing the semantic channel and nothing else. GP-UCB (Srinivas et al., 2010) receives no biological context at all and shows what performance is achievable with the semantic channel removed, and strong modern baselines (Shahriari et al., 2016; Eriksson et al., 2019; Cowen-Rivers et al., 2022) make that a demanding bar. Every arm is run four times per task, GP-UCB gets 20 seeds, and web search is disabled throughout except where stated (§5.3). The domain-agnostic arm has nothing to look up, so enabling tools would

confound retrieval with naming. Because the two LLM arms differ only in whether the parameter names are biologically meaningful, any difference between them comes from the prompt's semantic content. We score proposals rather than stated rationales, since a model's stated reasoning need not name the feature that actually drove its choice (Turpin et al., 2023). Appendix A.2 lists both prompt templates in full.

3.2 Constructing 45 tasks and their oracles from published experiments

We construct 45 parameterized design spaces from published wet-lab experiments, each pooling conditions across several papers, for 2,485 rows across 341 papers, a median of 7 papers per task (range 3–21). Tasks include CHO antibody titer, HEK293 prime-editing efficiency, lipid-nanoparticle size, *C. glutamicum* lysine titer, and microalgae biomass, so the panel spans cell-line engineering, gene editing, formulation, and fermentation. Each mixes numeric with categorical parameters.

A deterministic *oracle*, a model fitted to the extracted rows, answers in place of the assay. We fit one per task, choosing among gradient boosting, random forest, and a stacking ensemble by held-out fit. The panel median leave-one-out R^2 is 0.888.

Evaluating optimization methods against a simulator rather than a wet lab is established practice (van Lent et al., 2023). For the head-to-head comparison the oracle does not have to be an accurate model of the biology. It scores all three arms with the same function, so a consistent ranking of parameter choices makes that comparison fair. Section 5 needs more than a consistent ranking, since it depends on which designs each arm proposes rather than on which arm scores higher, and therefore on the oracle peaking where the biology does.

3.3 Grounding tasks and mechanism claims in the source papers

The *literature optimum* for a parameter is the value the best-performing experiment in the literature used, and the *literature convention* is the modal value across those same papers. Both are recorded from the extracted rows before any model runs, and both come from the source papers rather than from the oracle, so oracle artifacts cannot drive the conclusions.

For each task we also score the published-best design with the oracle and see where it lands in the range the oracle assigns across that task's rows. The published-best design falls in the top 5% on **18 of the 45 tasks**, which we call the *oracle-aligned* tasks. Appendix A.4 re-reports the main contrast on them. The mechanism analysis of Section 5.3 is likewise defined by the literature rather than by the oracle. A task-categorical pair is *divergent* when its literature convention is not its literature optimum.

3.4 Metrics: trajectory and endpoint over 30 iterations

Because rounds of feedback are the scarce resource, we fix the number of iterations to 30. We track the best score found so far, which can only rise. The **trajectory** is the area under that curve (bsf-AUC@30), so it rewards reaching a good design early. The **endpoint** is where the curve ends (bsf-Outcome@30), so it rewards only the final result. The two answer different questions (Mohr & van Rijn, 2024) and agree on 78.8% of task-model pairs ($\phi = 0.56$, $n = 712$), so a run can lead on one and trail on the other.

Where a score is reported as GP-UCB-normalized, it is the signed difference from the GP-UCB reference on that task divided by the reference itself, so 0 is parity and -0.18 is short of the reference by 18% of its own score. The scale is not a percentage of any maximum and is not bounded above, since a run can exceed the reference.

Win rates take the *model* as the unit, tested against 50% with a two-sided Wilcoxon signed-rank test and a bootstrap interval over models. Paired contrasts across tasks are mean differences with 95% bootstrap intervals and the count of tasks favoring each arm.

4 BIOLOGICAL CONTEXT IMPROVES NEITHER OPTIMIZATION TRAJECTORY NOR ENDPOINT

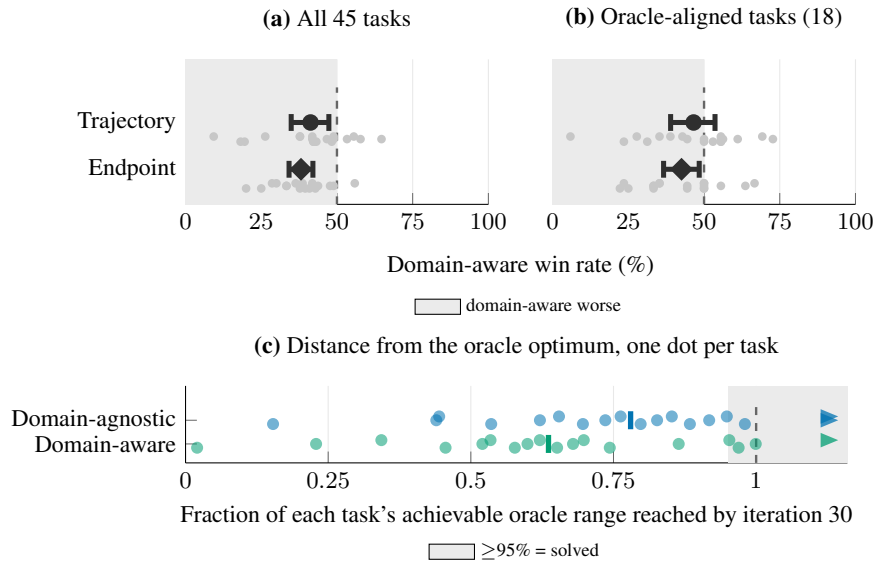

Figure 2: **Biological context improves neither the trajectory nor the endpoint, and the effect holds on the tasks whose optimum we verified against the source papers.** *(a)* All 45 biology tasks and *(b)* the 18 tasks whose oracle optimum reproduces the literature optimum. In both, each model contributes one win rate, plotted as the mean over models with a 95% bootstrap interval, with circles for trajectory, diamonds for endpoint, and gray dots for individual models. *(c)* A different quantity, the fraction of achievable oracle range reached by iteration 30 on those same 18 tasks, one dot per task per condition, with the median as a vertical rule and triangles for two tasks past the axis.

Biological context, supplied by domain-aware prompting, improves neither the trajectory nor the endpoint metric (Fig. 2a). Taking each model as one observation, the mean domain-aware win rate is 41.3% [34.9, 47.3] on trajectory ($p = 0.013$, 15 of 19 models below 50%) and 38.1% [34.2, 42.1] on endpoint ($p = 0.0003$, 17 of the 18 models scored on endpoint).

The result holds on the tasks where the oracle agrees with the literature optimum (Fig. 2b), so it is not an artifact of a mis-specified oracle. On the 18 oracle-aligned tasks the domain-aware endpoint gap holds, 42.6% [36.6, 48.4] with 10 of those same 18 models below 50% ($p = 0.023$). Trajectory on that subset is a null (46.6%, $p = 0.49$); with 18 tasks, that estimate remains imprecise (Appendix A.4). As expected from prior comparisons (Gupta et al., 2025; Huang et al., 2024; Rychert et al., 2025), no model outperforms GP-UCB by a meaningful margin in either condition, with 14 of the 19 at or below it in both (Fig. 6, Appendix A.5). The eight-model HEBO comparison likewise finds no statistically established majority-task superiority in either condition (Appendix A.3). The domain-aware deficit therefore adds to a starting position that is already no better than the optimizer these loops would replace.

The domain-awareness gap is not an artifact of both conditions saturating the oracle (Fig. 2c). On the 18 oracle-aligned tasks the fraction of achievable oracle range reached is grounded in an optimum the source papers report. Only 5 of 18 reach 95% in either condition, and domain-agnostic finishes higher on 12 of the 18 ($+0.088$ [-0.050, $+0.227$], $p = 0.071$), a difference consistent in direction but modest in magnitude. Runs reach or exceed the oracle's own optimum on the 22 tasks whose optimum we could not verify, exceeding it on 9 of them. That indicates the oracle extrapolates past every literature row, not that those tasks are solved (Appendix A.4).

5 WHY DOMAIN-AWARE PROMPTING UNDERPERFORMS

Biology-aware optimization here means optimizing a design when the parameter names, units, and labels are biologically meaningful. It does not mean optimizing over sequences, structures, or protocols, which this benchmark does not present.

Education tasks built the same way as the biology ones test whether the domain-awareness gap is specific to biology or a property of descriptive prompts (§5.1). Match rates at iteration 1 and at iteration 30 locate where in the loop the two conditions diverge (§5.2), and the six tasks whose literature convention is not their literature optimum show which values domain-aware prompting proposes instead (§5.3).

5.1 DO DOMAIN-AWARE PROMPTS GENERALLY HURT?

Domain-aware prompts are longer and more descriptive than domain-agnostic ones, so the domain-awareness gap could come from the form of the prompt rather than from what the model knows. A prompt-form effect would show up in any domain, not only in biology. We therefore built 10 education optimization tasks the same way we built the biology ones, from published learning-science experiments and public education datasets, among them Project STAR class size (Krueger, 1999), Open University engagement logs (Kuzilek et al., 2017), and ASSISTments randomized trials (Prihar et al., 2022). Each has the shape of a biology task, with a fixed set of numeric and categorical parameters, an oracle fitted to the extracted rows, the same 30 iterations, and the same scalar feedback, and only what the parameters describe changes. They are deposited with the biology tasks (Reproducibility statement).

On these tasks the impact of domain-aware prompting reverses. Domain-aware prompting outperforms domain-agnostic on 68.5% of task-model pairs on trajectory and 68.1% on endpoint ($p = 2 \times 10^{-6}$ and $p = 0.0035$), compared with 46.3% and 41.0% on biology. The domain-by-condition contrast is $+22.1\%$ on trajectory and $+27.1\%$ on endpoint ($p = 2 \times 10^{-7}$ and $p = 2 \times 10^{-5}$, Fig. 8, Appendix A.5). The reversal indicates that domain-aware prompts do not generally hurt LLM optimization, and that the effect is specific to biology.

5.2 WHERE IN THE OPTIMIZATION TRAJECTORY DOES THE DOMAIN-AWARE GAP OCCUR?

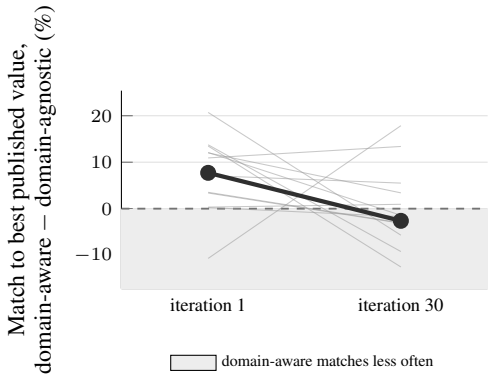

Figure 3: **Models given biological context match the literature optimum more often at iteration 1 and less often by iteration 30.** Each line shows domain-aware minus domain-agnostic in the rate of proposing the literature optimum, at iteration 1 and at iteration 30. The bold line is the estimate pooled over task-model cells and the faint lines are the individual models, of which one, Claude Opus 4.7, moves the other way on the fewest cells of any model in the panel (§A.1).

Returning to biology, we locate where in the 30 iterations the gap appears, narrowing to the 18 oracle-aligned tasks of Section 3.3, whose oracle optimum reproduces the literature optimum, so that what we measure reflects the biology rather than a poorly fitted oracle. The design a model proposes at iteration 1 reflects its prior alone, since no score has arrived yet. How much its match rate rises

over the iterations that follow is how far it responds to the oracle's feedback. Across those 18 tasks, models given domain-aware prompts propose the literature-optimum parameter value +7.7% more often than under domain-agnostic prompts at iteration 1. By iteration 30 domain-agnostic matches it more often, by 2.6%. The rise from iteration 1 to iteration 30 is +3.8% under domain-aware prompting and +14.1% under domain-agnostic, a difference of −10.3% ($p < 0.0001$, Fig. 3). Biological knowledge therefore improves the design a model starts from and reduces how far it responds to the feedback that follows.

5.3 Do LLMs with domain-aware prompts tend toward conventional choices rather than the optimum?

Why does biological context reduce how far a model responds to the oracle's feedback? One possibility is that the model stays anchored to the conventional parameter value in the literature rather than updating on the feedback it receives. If so, the gap should be largest where literature convention and the literature optimum disagree. We test that on six *divergent* tasks, picked by a rule applied to the extracted paper rows before any model runs: the tasks of the 18 whose most oracle-influential categorical parameter has a literature convention that is not its literature optimum. On those six, following convention and finding the optimum pull in opposite directions.

On those six tasks, models given domain-aware prompts name the conventional value on 24.9% of proposals, twice the 12.4% under domain-agnostic prompts, and keep naming it after the oracle has scored the literature optimum higher. Models given domain-aware prompts name the literature optimum on 21.6% of proposals, compared with 38.0%, higher under domain-agnostic on *all six*. GP-UCB, which receives no biological context at all, lies between the two conditions at 29.2%. Nested inside three wider subsets of tasks, ordered by how closely convention tracks the optimum, the gap appears only in these six (Fig. 4a). Three biology experts, blinded to condition, agreed with our anchored classification on 10 of 12 trajectory judgments (Appendix B.4). Fig. 4b shows one instance in full and Table 3 lists the six pairs.

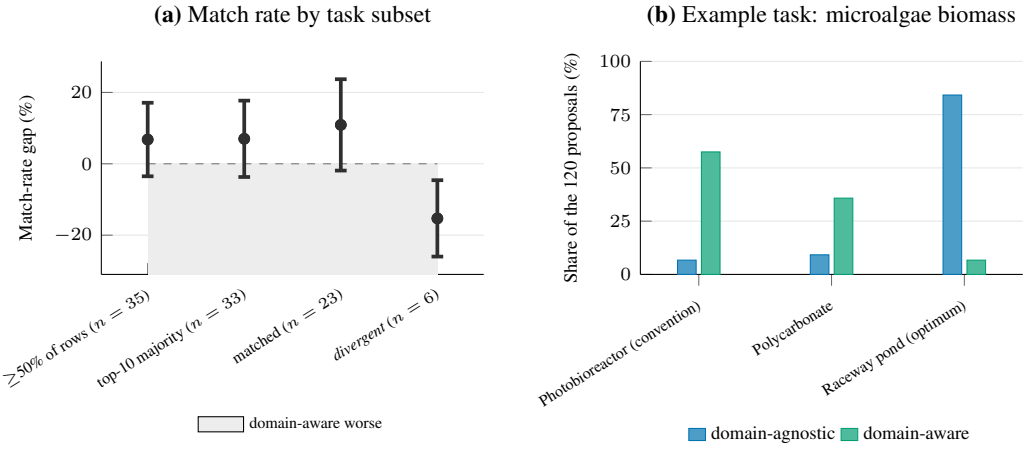

Figure 4: **The cost of biological context appears only where literature convention disagrees with the literature optimum. (a)** Match-rate gap for the value the best published experiment used, over four subsets of tasks nested by how closely literature convention tracks that value. Only the innermost subset, the six where the two disagree, clears zero, and it clears it downward. **(b)** One instance in full, Gemini 3.1 Pro on microalgae biomass, share of its 120 proposals naming each reactor type over four runs of 30 iterations. Raceway pond is the literature optimum at 1.13 g/L (Yang et al., 2018), photobioreactor the conventional value. Both panels show 95% bootstrap intervals. Exploration breadth on these six tasks is in Appendix B.2.

Narrow exploration does not explain the anchoring. On the six divergent tasks models with domain-aware prompts explore more of the space than under domain-agnostic prompts, wider on 52 of the 78 task-model pairs ($p = 0.011$), with CHO antibody titer the one task that reverses. Forcing web search at every iteration for both arms of Claude Opus 4.7 leaves domain-aware behind domain-agnostic by 20% on average (Appendix B.3). On five of six tasks the source paper predates every

model's release by over a year, though this does not establish training-set inclusion. On the pairs with a 12-month margin domain-aware still names the conventional value 24.0% compared with 15.5% (39 pairs, one-sided Fisher $p = 0.052$).

6 POST-TRAINING SHOWS BIOLOGY-AWARE OPTIMIZATION IS A SEPARATE CAPABILITY

Does the domain-awareness gap come from weak optimization? If it does, improving a model's raw optimization should improve its biology-aware optimization too. If it does not, biology-aware optimization is a capability that has to be trained for on its own. We train Llama 3.1 8B on raw optimization with no biology in the prompt, then give it the biology and check whether the improvement transfers.

We post-train Llama 3.1 8B with supervised fine-tuning (SFT) first, on 2,607 proposals from three frontier teachers, which the base model requires because its optimization is far below the GP-UCB reference. It sits further below that reference than any other model in the panel (Fig. 5a). Direct preference optimization (DPO) follows, on 5,197 preference pairs whose *chosen* side is the design GP-UCB would have proposed given that history and whose *rejected* side is the design actually proposed there. Each pair starts from a real rollout history, taken from the base model and three frontier models, so the pairs match the states the model meets at test time, and is kept only where the oracle confirms GP-UCB's design scored higher. All training data is rendered domain-agnostically. We train four independent DPO runs from the same fine-tuned adapter, one per seed, and evaluate each on the 16 held-out tasks with four runs per task (Appendix C).

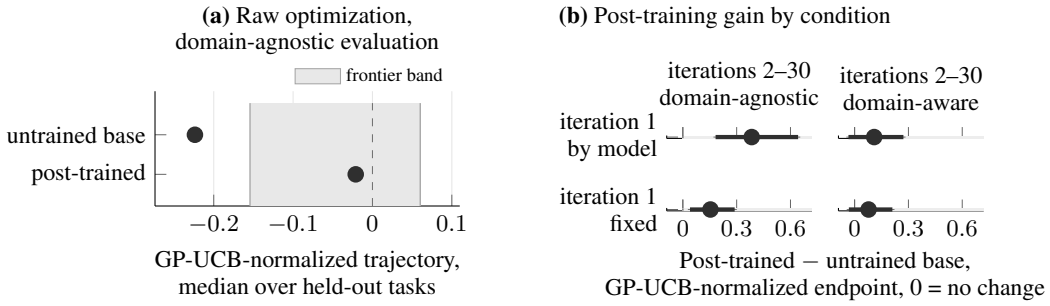

Figure 5: **Post-training on the optimization loop transfers under domain-agnostic prompts and not under domain-aware ones. (a)** The post-trained model and its untrained base against the band spanned by ten frontier models' medians on the same held-out tasks, all evaluated under domain-agnostic prompts. **(b)** Post-trained Llama 3.1 8B minus the untrained base, paired within task and normalized to each task's GP-UCB reference, so only the iteration-1 design (rows) and the prompt condition for iterations 2–30 (columns) change. The shaded half is where post-training improves. Both domain-agnostic intervals clear the no-change line and neither domain-aware interval does. Panel (a) shows trajectory and (b) endpoint, with 95% bootstrap intervals over four seeds per task.

Post-training improves raw optimization, but the gain does not transfer to biology-aware optimization. Under domain-agnostic prompts the DPO-trained model gains $+0.385$ on trajectory and $+0.384$ on endpoint, positive on 13 and 14 of 16 tasks ($p = 0.0004$, $p = 0.0009$), which brings an 8B model to GP-UCB parity and into the band spanned by ten frontier models on the same tasks (Fig. 5a). Under domain-aware prompts the same post-training gain is null on both metrics, $+0.026$ and $+0.109$, and the difference between the two prompt conditions is itself significant, $+0.359$ [$+0.124$, $+0.655$] on trajectory ($p = 0.0065$). Neither SFT-only arm improves on the untrained base by more than noise, 6 of 11 tasks on trajectory ($p = 0.45$), so the measured improvement comes from the preference stage (Appendix C). On the 8 held-out tasks that are also oracle-aligned the domain-agnostic gain strengthens and the domain-aware gain stays null (Appendix A.4). Handing every arm the same iteration-1 design does not change it (Fig. 5b). Because iteration 1 is matched, this comparison concerns the biological names used during iterations 2–30.

Biology-aware optimization is therefore a capability distinct from raw optimization and from the model's ability to supply a good initial design. In this experiment, raw optimization scores did not predict performance once the parameter names were biologically meaningful.

7 LIMITATIONS

We describe three main limitations. First, anchoring to convention is one failure mode rather than a complete account of the domain-awareness gap. Second, oracles stand in for a wet lab, interpolating within a design space rather than extrapolating across papers. Wherever a conclusion depends on the oracle peaking where the biology does, we check it against the source papers instead, re-reporting the main contrast on the 18 oracle-aligned tasks and reading convention and optimum from the published rows before any model runs. Relying on oracles is also why we cannot post-train directly on domain-aware trajectories. Third, the post-training result rests on one model family over four seeds. Whether a larger model would transfer the gain is untested, though the domain-awareness gap appears throughout the 19-model panel even as absolute raw optimization ability increases, which makes scale an unlikely explanation.

8 DISCUSSION AND CONCLUSION

We find that biological context does not reliably improve LLM-guided experimental optimization, and we characterize four components of that gap. First, it is specific to biology, with domain-aware prompting ahead on 68.5% of education task-model pairs by trajectory against 46.3% on biology (§5.1). Second, biological context improves a model's initial design by 7.7% and worsens its response to feedback by 10.3% (§5.2). Third, it is concentrated rather than diffuse, appearing where literature convention and the literature optimum diverge and not where they agree (§5.3). Fourth, biology-aware optimization looks like a separate capability from raw optimization, since post-training raises domain-agnostic performance without any of the gain transferring (§6).

Before trusting an LLM as an optimizer, practitioners should check whether the optimum is likely to diverge from conventional choices in the literature. Where divergence is likely, an LLM is most useful for the initial design, with a prior-free optimizer running the rest. Specialist models trained on non-text-native representations remain a reasonable bet, given that our setup indicates biology-aware optimization is the bottleneck and specialist biology tools could be an advantage here.

Taken together, biological priors, optimization over abstract parameters, and optimization under biological context behave as three separate capabilities, with LLMs specifically weak at biology-aware optimization. Targeting it directly could be fruitful for autonomous laboratory loops, as could testing tool calling and the other capabilities under the same controlled comparison.

REPRODUCIBILITY STATEMENT

All 45 biology and 10 education task definitions, the fitted oracles, and the GP-UCB baseline outputs are deposited for reviewers at `https://dataverse.harvard.edu/previewurl.xhtml?token=acf76ef1-2d11-4cd5-9264-87ab45774014`. The analysis code is at `https://anonymous.4open.science/r/learning-efficiency-E280`. The naming-condition result reported here is reproduced by `eval/compute_naming_condition_stats.py`, and each figure is emitted by a named script in `paper_scripts/`. The panels drawn from released data files rebuild from the code alone. Per-run trajectories are available on request. Sampling settings and the compute used are given in Appendix A.1. Both links preserve anonymity for double-blind review.

ETHICS AND BROADER IMPACT

This work evaluates optimizer behavior against oracles fitted to already-published experimental data. No new wet-lab experiments were run and no hazardous protocols were generated or optimized. The tasks are process-optimization problems drawn from published protein expression, cell-culture and formulation work, and none of them concerns a pathogen, a toxin or any capability of dual-use concern; the benchmark is built entirely from data its source papers already made public. We note

the dual-use question anyway, because the method under study is generic: a system that proposes experimental conditions more efficiently is indifferent to what is being optimized, so the same loop applied to a hazardous objective would pose risks this benchmark does not measure. Our result cuts against capability in that setting rather than for it, since it finds that biological context degrades rather than improves the design an LLM finally selects. The practical finding is a caution for practitioners deploying LLMs in closed experimental loops, where an unnoticed prior-driven failure would waste scarce laboratory rounds.

AI USE STATEMENT

LLMs are the object of study in this paper, and all evaluated models are listed in Appendix A.1. LLMs were also used as a tool in four places. First, in constructing the benchmark. An LLM-assisted pipeline searched the literature for candidate papers and extracted the parameter values and measured outcomes that became each task's data points, including unit conversion into a common scale. Every extracted task passed several verification stages, the last of which was human review of the parameter definitions, units, and best literature rows against the source papers. Second, in surveying related work. An LLM summarized roughly 100 candidate papers and organized them into an outline of the literature, which the authors reviewed. Third, in preparing the manuscript. The outline and the paragraphs were written by the authors, and an LLM assisted with copyediting and sentence phrasing. Fourth, in writing and editing code. An LLM assisted with the evaluation harness and the analysis and figure scripts, all of which the authors reviewed, and each reported number is regenerated from the released scripts. The experiments were designed by the authors, as were the statistical analyses and every claim. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Pablo Carbonell, Adrian J. Jervis, Christopher J. Robinson, Cunyu Yan, Mark Dunstan, Neil Swainston, Maria Vinaixa, Katherine A. Hollywood, Andrew Currin, Nicholas J. W. Rattray, Sandra Taylor, Reynard Spiess, Rehana Sung, Alan R. Williams, Donal Fellows, Natalie J. Stanford, Paul Mulherin, Rosalind Le Feuvre, Perdita Barran, Royston Goodacre, Nicholas J. Turner, Carole Goble, George Guoqiang Chen, Douglas B. Kell, Jason Micklefield, Rainer Breitling, Eriko Takano, Jean-Loup Faulon, and Nigel S. Scrutton. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. *Communications Biology*, 1(1), 2018. doi: 10.1038/s42003-018-0076-9.

Alexander I. Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan-Rhys Griffiths, Alexandre Max Maraval, Hao Jianye, Jun Wang, Jan Peters, and Haitham Bou Ammar. HEBO: Pushing the limits of sample-efficient hyperparameter optimisation. *Journal of Artificial Intelligence Research*, 74:1269–1349, 2022.

Samagra Dvivedi, Nitin Minocha, Shuang Chen, Beata Chertok, Daniel Estabrook, Yuchen Fan, Suyong Han, Deepak Jain, Manish S. Kelkar, Alison Lui, Nandkishor K. Nere, Phenil J. Patel, Kenneth G. Rodriguez, Eric R. Sacia, Priyanka G. Singh, Dimitri Skliar, Emad I. Wafa, and Meenesh R. Singh. Towards self-driving laboratories in the biopharmaceutical industry. *Nature Synthesis*, 2026. doi: 10.1038/s44160-026-01131-3.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Rushil Gupta, Jason Hartford, and Bang Liu. LLMs for bayesian optimization in scientific domains: Are we there yet? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. arXiv:2509.21403.

Beichen Huang, Xingyu Wu, Yu Zhou, Jibin Wu, Liang Feng, Ran Cheng, and Kay Chen Tan. Exploring the true potential: Evaluating the black-box optimization capability of large language models. *arXiv preprint arXiv:2404.06290*, 2024.

Katarzyna Kobalczyk, Zhiyuan Jerry Lin, Benjamin Letham, Zhuokai Zhao, Maximilian Balandat, and Eytan Bakshy. LILO: Bayesian optimization with natural language feedback. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2510.17671.

Alan B. Krueger. Experimental estimates of education production functions. *The Quarterly Journal of Economics*, 114(2):497–532, 1999.

Jakub Kuzilek, Martin Hlosta, and Zdenek Zdrahal. Open university learning analytics dataset. *Scientific Data*, 4:170171, 2017.

Jon M. Laurent, Joseph D. Janizek, Michael Ruzo, Michaela M. Hinks, Michael J. Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D. White, and Samuel G. Rodriques. LAB-Bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2402.03921.

Robert MacKnight, Jose Emilio Regio, Jeffrey G. Ethier, Luke A. Baldwin, and Gabe Gomes. Pretrained knowledge elevates large language models beyond traditional chemical reaction optimizers. *arXiv preprint arXiv:2509.00103*, 2025.

Carl-Fredrik Mandenius and Anders Brundin. Bioprocess optimization using design-of-experiments methodology. *Biotechnology Progress*, 24(6):1191–1203, 2008. doi: 10.1002/btpr.67.

Jamison Meindl, Yunsheng Tian, Tony Cui, Veronika Thost, Zhang-Wei Hong, Jie Chen, Wojciech Matusik, and Mina Konaković Luković. Semanticopt: Towards llm-based semantic black-box optimization. *arXiv preprint arXiv:2510.25404*, 2025.

Felix Mohr and Jan N. van Rijn. Learning curves for decision making in supervised machine learning: a survey. *Machine Learning*, 113(11):8371–8425, 2024.

Siddharth Narayanan, James D. Braza, Ryan-Rhys Griffiths, Manu Ponnapati, Albert Bou, Jon Laurent, Ori Kabeli, Geemi Wellawatte, Sam Cox, Samuel G. Rodriques, and Andrew D. White. Aviary: Training language agents on challenging scientific tasks. *arXiv preprint arXiv:2412.21154*, 2024.

Tung Nguyen and Aditya Grover. LICO: Large language models for in-context molecular optimization. In *International Conference on Learning Representations (ICLR)*, 2025.

OpenAI. Accelerating biological research in the wet lab. `https://openai.com/index/accelerating-biological-research-in-the-wet-lab/`, 2025.

Ethan Prihar, Manaal Syed, Korinn Ostrow, Stacy Shaw, Adam Sales, and Neil Heffernan. Exploring common trends in online educational experiments. In *Proceedings of the 15th International Conference on Educational Data Mining (EDM)*, 2022.

Yusuf H. Roohani, Andrew Lee, Qian Huang, Jian Vora, Zachary Steinhart, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. BioDiscoveryAgent: An AI agent for designing genetic perturbation experiments. In *International Conference on Learning Representations (ICLR)*, 2025.

Adam Rychert, Gasper Spagnolo, and Evgenii Posashkov. Reproducibility study of large language model bayesian optimization. *arXiv preprint arXiv:2511.18891*, 2025.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016.

Alexus A. Smith, Edmund L. Wong, Ronan C. Donovan, et al. Using a GPT-5-driven autonomous lab to optimize the cost and titer of cell-free protein synthesis. *bioRxiv*, 2026. doi: 10.64898/2026.02.05.703998. URL https://www.biorxiv.org/content/10.64898/2026.02.05.703998v1.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 1015–1022, 2010.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Paul van Lent, Joep Schmitz, and Thomas Abeel. Simulated design–build–test–learn cycles for consistent comparison of machine learning methods in metabolic engineering. *ACS Synthetic Biology*, 12(9):2588–2599, 2023. doi: 10.1021/acssynbio.3c00186.

Ian Walsh, Fumi Shozui, Ayaka Sato, Uiseon Park, Hyeran Cho, and Charmaine Chia. Toward machine learning-guided cho bioprocess and media optimization for improved titer and glycosylation. *Biotechnology Journal*, 20(11), 2025. doi: 10.1002/biot.70149.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*, 2024.

Haijian Yang, Qiaoning He, and Chunxiang Hu. Feasibility of biodiesel production and co_2 emission reduction by *monoraphidium dybowskii* lb50 under semi-continuous culture with open raceway ponds in the desert area. *Biotechnology for Biofuels*, 11(1), 2018. doi: 10.1186/s13068-018-1068-1.

A Benchmark and evaluation details

How the models were sampled, what each condition was shown, and how much the oracle-alignment restriction changes the main result.

A.1 Models, sampling, and provenance

Table 1 lists all 19 models in release order with the API slug used to route each one and the number of biology tasks contributing to each metric. Thirteen models were routed through the shared client registry, which records the exact slug. The remaining six were run through per-model evaluation harnesses that recorded the model label but not the resolved slug or the request timestamp. We report those by label and mark the slug unavailable. This asymmetry is the same one noted in Section 4, and it is why we draw no release-cohort conclusion.

Claude Opus 4.7 refuses a subset of domain-aware biology prompts, returning an empty response rather than a parameter set on tasks whose names use pathogen-target, antibody-binding, or antimicrobial terminology. The domain-agnostic renderings of those same tasks complete normally. Its coverage in the categorical analyses is therefore fewer tasks than for most models, and its estimate in Fig. 3 rests on 56 task-model cells where the best-covered model has 447. Claude Opus 4.7 is also the one model whose match rate rises between iteration 1 and iteration 30, on the least well covered estimate in the panel.

Every task-condition pair is run four times. Table 1 counts task comparisons contributing to each metric, excluding ties and pairs without both conditions, rather than all completed task runs. Each request sets only the model and a per-model token cap, leaving sampling at each provider's default, so the four runs of a cell differ by provider-side nondeterminism rather than by a set seed or temperature. One request is issued per iteration, so a model covering the full biology panel in both conditions makes $45 \times 2 \times 4 \times 30 = 10{,}800$ requests. Post-training ran on a single NVIDIA A100 80GB (Vertex `a2-ultragpu-1g`) for the supervised stage and each of the four preference seeds. GP-UCB uses 20 independent seeds per task. Open markers in Fig. 7 denote models scored from their own evaluation summaries rather than the curated task table.

Table 1: The 19 evaluated models in release order, with the API slug used and the number of biology tasks contributing to each metric. Slugs are the exact routing identifiers recorded by the shared client registry. Six models were run through per-model evaluation harnesses that did not record the resolved slug, shown as "–". In the endpoint column, "–" denotes unavailable endpoint data.

Model	API slug	Tasks (traj.)	Tasks (endp.)
llama-3.1-8b	`meta-llama/llama-3.1-8b-instruct`	44	44
deepseek-v3.2	`deepseek/deepseek-v3.2`	36	35
gemini-3-flash	`google/gemini-3-flash-preview`	44	43
glm-4.7	`z-ai/glm-4.7`	45	–
claude-sonnet-4.6	`anthropic/claude-sonnet-4-6`	42	41
gemini-3.1-pro	`google/gemini-3.1-pro-preview`	45	42
gpt-5.4	`openai/gpt-5.4`	45	44
gpt-5.4-mini	`openai/gpt-5.4-mini`	44	44
claude-opus-4.7	`anthropic/claude-opus-4.7`	36	34
gpt-5.5	`openai/gpt-5.5`	45	45
gemini-3.5-flash	`google/gemini-3.5-flash`	45	45
claude-opus-4.8	`anthropic/claude-opus-4.8`	35	33
glm-5.2	`z-ai/glm-5.2`	45	44
gpt-5.6-sol	-	45	45
gpt-5.6-terra	-	45	45
muse-spark-1.1	-	21	21
kimi-k3	-	45	45
gemini-3.6-flash	-	45	44
claude-opus-5	-	32	29

Power. Per-model tests are underpowered by design. Four runs per pair is why nearly every per-model Wilson interval spans 50%. Primary inference uses the pooled and mixed-effects analyses. The per-model panel checks whether the aggregate result is driven by one model.

A.2 Prompt templates

Both arms propose one configuration per iteration and receive one scalar in return. The prompts are reproduced below as the code emits them.

Domain-aware. The system prompt is written per task and names the biology. For CHO antibody titer it begins:

```
You are an expert in CHO cell culture and monoclonal antibody
production.  You are helping design experiments to maximize
monoclonal antibody titer (g/L).
When generating hypotheses, return ONLY a valid JSON array with
these fields:
- hypothesis_name (string, 3-5 words)
- cho_cell_line (string:  "CHO-K1", "CHO-S", or "CHO-DG44")
- cell_specific_perfusion_rate (number:  100-500 pL/cell/day) ...
```

The opening turn asks for a hypothesis, and each later turn returns the oracle's score and asks for a revision:

```
Generate {n} diverse optimization hypotheses to {maximize|minimize}
{target description}.
Return ONLY the JSON array:

Here are the results from checking your hypotheses against the
{task} dataset:
## Hypothesis 1:  {name}
{metric description}:  {value} {unit}
Based on this feedback, generate {n} NEW hypotheses that:
1.  First, explain what pattern or rule you think explains the
```

```
results so far -- what causes high vs low {target description}?
2.  Use that reasoning to propose new parameter combinations that
test or exploit your theory
3.  In the rationale field for each hypothesis, state the specific
causal claim you are testing
Return ONLY the JSON array:
```

Domain-agnostic. The system prompt is generated from the same `TaskConfig` with the names replaced and the biology removed:

```
You are optimizing a black-box system by choosing parameter
values to {maximize|minimize} an objective function.  You have no
information about what the system does -- only the parameter space
and the feedback you receive.
When proposing a configuration, return ONLY a valid JSON array with
one object containing:
- X1 (number, range {min}-{max})
- C1 (string, one of:  "A", "B", ...)
- rationale (string, 1 sentence explaining your reasoning)
Maximize the objective value based on prior feedback.
```

Example, second iteration. At iteration 2 the request carries the system prompt above and three further messages, the opening turn, the model's own proposal, and the feedback turn:

```
[user]
Propose a configuration to {maximize|minimize} the objective
function.
Return ONLY the JSON array:

[assistant]
[
  {
    "X1":  {value},
    "C1":  "A",
    "rationale":  "{one sentence}"
  }
]

[user]
The objective value for your last configuration was:  {value}
Propose a new configuration that you expect to achieve a
{higher|lower} objective value.  Learn from the previous result.
Return ONLY the JSON array:
```

Iteration k appends that assistant and user pair $k-1$ times, so the model sees every proposal it has made and every score it has received.

The domain-aware arm accumulates its conversation the same way, so the domain-agnostic turn naming only the last value is a statement about the newest turn, not about what the model can see. Both arms are asked to reason from the feedback and to record a rationale, and both see the same parameter space, the same oracle and the same budget. The domain-aware turn phrases that request in the vocabulary of the task and the domain-agnostic turn in the vocabulary of an objective function. One asymmetry follows from these templates and we tested it directly. The domain-agnostic turn restates the objective direction every iteration while the domain-aware turn states it only in the system prompt and the opening turn. A control that restates the direction in every domain-aware turn does not rescue the minimize-objective tasks (-0.140 trajectory and -0.154 endpoint, 0 of 5 tasks favoring the restated arm), and leaves maximize tasks unchanged as a placebo, so the asymmetry is not what produces the effect. GP-UCB receives no text. Each configuration is encoded as scaled numeric parameters with one-hot categoricals, and the acquisition is $\mu(x) + \kappa\sigma(x)$.

A.3 COMPARISON AGAINST HEBO

We also compare against HEBO (Cowen-Rivers et al., 2022), an alternative Bayesian optimizer for mixed numeric and categorical spaces. This completed comparison covers the eight models in Table 2, rather than the full 19-model panel. HEBO receives four runs per task at the same 30-iteration budget. We compare each model's median bsf-AUC@30 against HEBO's within task, respecting the objective direction. The 95% intervals resample tasks and runs within tasks, including both LLM and HEBO runs (2,000 bootstrap replicates). No model's interval lies wholly above 50% in either prompt condition, consistent with the GP-UCB comparison.

Table 2: Percentage of biology tasks on which each model outperforms HEBO by bsf-AUC@30, with 95% bootstrap intervals. These are the original eight-model comparison results, using available trajectories for each model and condition. Task counts are shown as domain-aware/domain-agnostic and refer to this comparison's archived cohort.

Model	Tasks	Domain-aware	Domain-agnostic
Claude Opus 4.7	36/36	64 [44, 78]	39 [31, 67]
Gemini 3.1 Pro	45/45	36 [22, 49]	60 [44, 73]
Claude Sonnet 4.6	43/45	30 [19, 47]	42 [31, 60]
DeepSeek V3.2	41/40	44 [27, 56]	38 [25, 55]
Gemini 3 Flash	45/45	27 [16, 42]	38 [27, 56]
GPT-5.4	45/45	36 [22, 49]	33 [22, 51]
GPT-5.4 Mini	45/45	27 [13, 38]	29 [18, 44]
Llama 3.1 8B	45/45	16 [7, 29]	7 [2, 20]

A.4 ORACLE-ALIGNMENT ROBUSTNESS

A task is oracle-aligned when the oracle scores the published-best design within the top 5% of the range it assigns across that task's rows. Eighteen of the 45 biology tasks qualify and 27 do not, 5 of which are unscorable on the saturation scale below. Fitting the same mixed-effects model as Section 4 within each split yields, for the domain-aware win rate:

Split	Observations	Trajectory	Endpoint
All biology tasks	781 / 725	41.1% [30.6, 51.6]	38.4% [28.8, 48.0]
Oracle-aligned (18 tasks)	302 / 290	45.0% [28.9, 61.2]	41.5% [24.6, 58.4]
Not aligned (27 tasks)	479 / 435	38.5% [26.4, 50.7]	36.3% [24.8, 47.8]

On the same two splits, the saturation scale of Section 4 yields:

Split	Tasks	Domain-agnostic	Domain-aware	Reach 95%
Oracle-aligned	18	0.780	0.636	5
Not aligned	22	0.921	0.909	12
Alignment unscorable	5	1.186	1.106	4

Entries are medians of per-task medians, on 13 models. The paired domain-agnostic minus domain-aware difference is $+0.088$ $[-0.050, +0.227]$ on the aligned split (12 of 18, $p = 0.071$) and $+0.122$ $[+0.010, +0.246]$ on the not-aligned split (13 of 22, $p = 0.080$): the same direction and a comparable magnitude either way, though the bootstrap interval and the signed-rank test disagree about which split clears 0.05, so neither establishes the effect on this scale by itself. Nine of the 22 not-aligned tasks and 4 of the 5 unscorable ones exceed the oracle's own optimum, which is why the ceiling argument is read on the aligned split.

Task or model as the unit. The two units disagree on this subset, so we give both. Treating each model as one observation and testing its win rate against 50% across models, the endpoint contrast is -11.9% $[-15.8, -7.9]$ on all 45 tasks (17 of the 18 models with endpoint scores below 50%, $p = 0.0003$) and -7.4% $[-13.4, -1.6]$ on the aligned subset (10 of 18, $p = 0.023$). Under the mixed-effects model the aligned endpoint estimate is 41.5% with $p = 0.33$. Both estimators are

reported. The model means are correlated through task difficulty, so the per-model interval is the narrower of the two. The task variance component is roughly twenty times the model component, which puts the effective sample size nearer the number of tasks than the number of models. The per-model figures do answer a different and legitimate question, whether the domain-awareness gap is idiosyncratic to a few models, and there the answer is no.

Both win-rate splits sit below 50% on both metrics, and the aligned split is the weaker of the two, so the effect is not an artifact of oracles that disagree with the literature. The aligned intervals span 50%: with 18 tasks and task-level variance dominating (§3.4), the subset does not have the power to establish the effect by itself, which is why the main result is reported on all 45 tasks and this split is a direction check.

Restricting the post-training contrast to the 8 held-out tasks that are also oracle-aligned strengthens the domain-agnostic gain, from $+0.385$ to $+0.542$ on trajectory and from $+0.384$ to $+0.518$ on endpoint, where it is positive on 8 of 8 tasks. The domain-aware arm stays null, $+0.026 \rightarrow +0.032$ and $+0.109 \rightarrow +0.107$. The same restriction attenuates the main result of §4, so oracle misspecification inflates that comparison but cannot explain this one. Separately, literature rows are frequently partial: the median best literature row reports 71% of its task's parameters, reflecting incomplete reporting in the source literature rather than selection. Scoring is identical across rows and the key categorical used in Section 5.3 is required to be present by construction, and completeness does not predict alignment (Spearman $\rho = 0.13$, $p = 0.42$). Restricted to the four divergent tasks whose best literature row reports at least half its parameters, the categorical effect is $+11.4\%$ $[+5.6, +18.5]$ on 4 of 4, compared with $+18.1$ on all six. The two dropped tasks hold the two largest effects, so both numbers should be read together.

A.5 Per-model win rates

No evaluated model outperforms GP-UCB by a meaningful margin in either condition, which is the comparison §4 cites. The second figure shows each model's domain-aware win rate under both metrics, showing that no individual model drives the pooled result. Education win rates and their biology comparators (46.3% trajectory, 41.0% endpoint) are unweighted proportions over task-model pairs. The biology mixed-effects estimates are 41.1% and 38.4%.

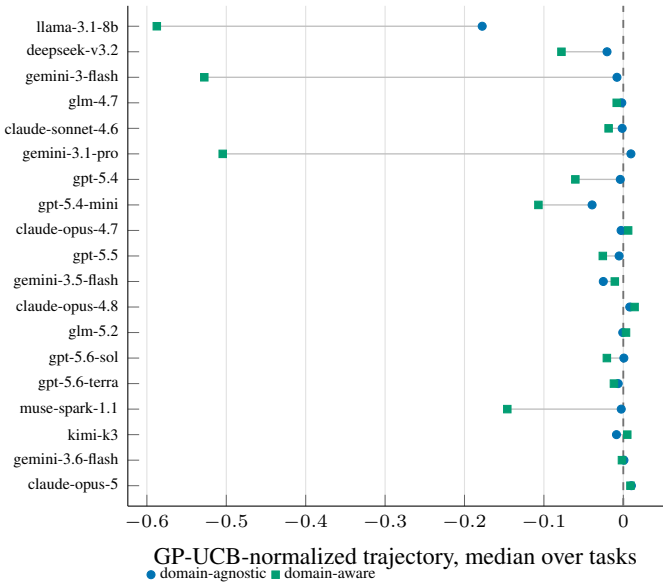

Figure 6: **No evaluated model outperforms GP-UCB by a meaningful margin, in either condition.** Median GP-UCB-normalized trajectory score for each of the 19 evaluated models, in release order, with 0 the GP-UCB reference. Circles are domain-agnostic, squares domain-aware, and the gray rule joins the two conditions for one model. Most models sit at or below the reference in both conditions and the largest positive margin is $+0.014$, while domain-agnostic is the higher of the two conditions on 14 of 19. No post-trained arms appear here.

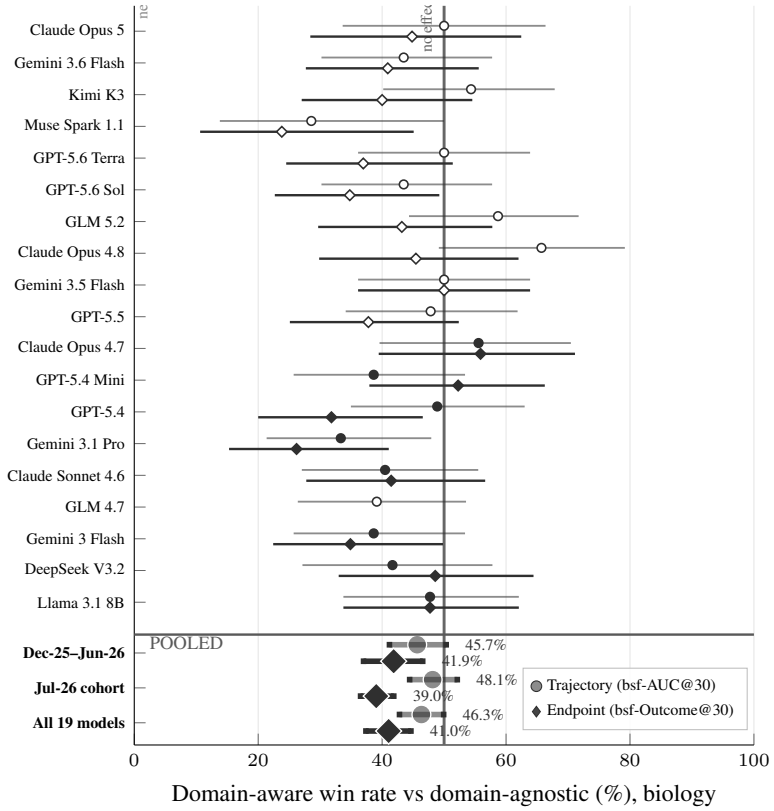

Figure 7: **No individual model shows a significant domain-aware advantage.** Per-model share of biology tasks on which domain-aware prompting outperforms domain-agnostic, under the trajectory metric (circles) and the endpoint metric (diamonds), for all 19 evaluated models in release order. Bars are Wilson 95% confidence intervals and the solid rule is the 50% no-effect null. Almost every per-model interval spans the null. Per-model tests are underpowered by design at four runs per pair, and primary inference uses the pooled and mixed-effects analyses (Fig. 2). The three rows below the rule pool task-model pairs with a bootstrap over models, for comparison with the mixed-effects estimates. Those pooled rows are cell-weighted, so they differ from the mixed-effects values. We do not draw a release-cohort conclusion from this panel. The six models with unavailable routing slugs were all scored through per-model evaluation summaries rather than the shared client registry (§A.1), so recency and scoring path are not separable here. Open markers identify those models. Four runs per task-condition pair.

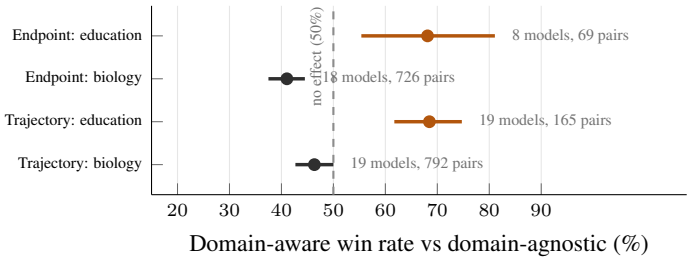

Figure 8: **Real parameter names help on education tasks but not on biology, so the null is not a general property of descriptive prompts.** Pooled rate at which domain-aware prompting outperforms domain-agnostic, under both metrics, on the biology panel (cobalt) and on the 10 education tasks (teal). Bars are 95% bootstrap intervals resampling models. The dashed line is the 50% no-effect null. On education both intervals sit well above the null, and on biology the endpoint interval lies below it. Counts at the right give the models and task-model pairs behind each row. Education endpoint rests on fewer models, since the later models' education runs do not all include the endpoint metric. Four runs per task-condition pair. Per-domain statistics in §5.1.

B THE CATEGORICAL MECHANISM

The six divergent tasks, the anchoring contrast per model and over the full population, and the forced-retrieval protocol that tests whether the effect is missing knowledge.

B.1 THE SIX DIVERGENT TASKS

Table 3 names the six tasks of §5.3, each with its key categorical parameter, the literature convention for that parameter, the literature optimum it disagrees with, and the values both were read from.

B.2 CATEGORICAL ANCHORING, PER MODEL AND FULL POPULATION

Fixing the iteration-1 design separates what comes from the first proposal from what happens during iterations 2–30. Injecting one identical design and varying only the prompt condition for iterations 2–30 moves the domain-aware median from -0.662 to -0.107, a paired gain of $+0.725$ $[-0.056, +1.914]$ on 6 of 8 tasks ($p = 0.098$), and with that design held fixed moving to domain-agnostic prompts is worth a further $+0.177$ $[+0.038, +0.317]$ on 7 of 8 ($p = 0.039$) and $+0.210$ on endpoint on 8 of 8 ($p = 0.004$). The prior term is the larger estimate but its interval includes zero, and the optimization term is smaller with an interval that excludes it.

The anchoring contrast is shared across models rather than produced by one of them (Fig. 9). Repeating the convention-versus-optimum split over all 118 task-categorical pairs, with no selection of one parameter per task, moves it in the same direction but not significantly, $+3.4\%$ $[-2.7, +9.4]$, so we read the split on the six divergent tasks and treat the whole-population version as directional only.

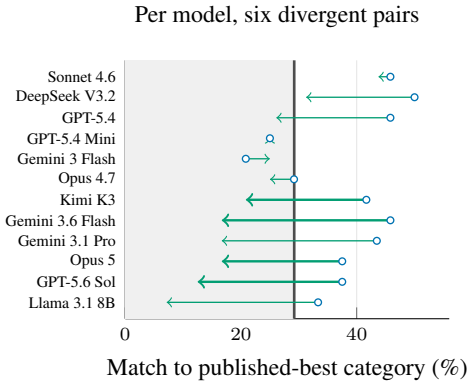

Figure 9: **The anchoring effect holds model by model, so no single model produces it.** Each arrow runs from a model's domain-agnostic rate (open circle) to its domain-aware rate on the six pairs where convention and the literature optimum diverge. Heavier arrows are models released after March 2026. The shaded region is everything below GP-UCB, an optimizer given no biological context at all. The three-way ordering is descriptive, since neither pairwise gap reaches significance. The direction is shared across nearly every model, so no single model drives the pooled result.

B.3 FORCED-RETRIEVAL PROTOCOL

Retrieval is tested dynamically. With web search enabled at inference, can the model look the literature optimum up? We re-ran Claude Opus 4.7 on the six literature-divergent tasks with search forced through OpenRouter's `tool_choice='any'` and the `web_search_20250305` tool, which guarantees a search call at every iteration, logging at minimum 240 calls per task across four runs of 30 iterations. Entries are the fraction of iterations whose proposal matches the literature optimum (Table 4).

Domain-agnostic exceeds domain-aware on three of the six tasks with search off and four of six with it on. A paired Wilcoxon comparing the (agnostic minus aware) difference on to off yields $p = 0.31$

Table 3: **The six literature-divergent tasks.** On each of these tasks, the categorical value used most often across the source papers is not the value used by the single best-performing published experiment. Domain-agnostic prompting proposes that best-experiment value more often than domain-aware on all six, a paired mean of $+18.1$ percentage points. Two rows carry data caveats. The CHL-YN convention is a 3–3 modal tie and the transaminase Published best entry exceeds 100, and dropping either or both strengthens the effect to $+20.0$, $+19.5$ and $+22.2$ percentage points respectively, positive on every task in every variant (§5.3). Source DOIs are recorded in the `source_paper` column of the deposited task datasets.

Task	Key categorical	Literature convention	Published best
HEK293 prime editing	delivery method	Transfection	Baculovirus
Microalgae biomass	reactor type	Photobioreactor	Open raceway pond
VHH nanobody expression	expression vector	pET-22b	pCold-I
CHL-YN cell line	feed-rate strategy	Dynamic	Adaptive
CHO antibody titer	CHO cell line	CHO-S	CHO-K1
Transaminase amination	cosolvent	Toluene	ChCl:EG:1,2-PD (deep eutectic)

Table 4: **Forcing web search at every iteration does not close the gap on the six literature-divergent tasks.** Fraction of iterations whose proposal matches the literature optimum, for Claude Opus 4.7 over four runs of 30 iterations per task.

	Search off		Search on	
Task	Aware	Agnostic	Aware	Agnostic
`hek293_prime_editing_indel_freq`	16.7	40.8	0.8	38.3
`microalgae_biomass_multi_strain`	40.0	32.5	35.8	90.0
`vhh_nanobody_expression_ecoli`	0.0	4.2	0.0	2.5
`chl_yn_cell_line_optimization`	64.2	96.7	56.7	94.2
`cho_antibody_titer`	15.0	8.3	16.7	5.8
`transaminase_amination_conversion`	2.5	0.8	2.5	1.7

at $n = 6$, so we cannot reject the null that search availability makes no difference at this n. Taking direction alone, the asymmetry persists or strengthens. These counts cover Claude Opus 4.7 alone. They are not in tension with the 6 of 6 reversal in Section 5.3, which is the same contrast averaged over 13 models. Pooling models recovers the unanimous direction that a single model at $n = 6$ does not. The result was replicated on a second sample, the five tasks where Opus's domain-aware match rate is worst, two of them overlapping the divergent set, with the same conclusion.

B.4 Biology-expert review of the anchoring classification

Anchoring is defined here by a quantitative rule, so we also asked whether a domain practitioner would identify the same pattern. Three biology experts with experimental-biology backgrounds, blinded to our hypotheses and to the condition labels, reviewed four domain-aware trajectories, one each from Gemini 3 Flash, GPT-5.4 Mini, Claude Sonnet 4.6, and Gemini 3.1 Pro. Each trajectory showed five iterations of proposed designs, rationales, and oracle scores. Reviewers were given a written definition of anchoring, persisting with similar or domain-typical designs despite weak or stagnant outcomes, and asked to classify each trajectory as anchored or responsive and to rate its reasoning from 1 (generic) to 5 (expert-level). They were compensated well above minimum wage.

Reviewers agreed with our classification on 10 of 12 judgments. The two trajectories we labeled anchored drew 3 of 3 and 2 of 3 anchored votes, and the two we labeled responsive drew 3 of 3 and 2 of 3 responsive votes. Reasoning ratings separate in the same direction, 3.17 for anchored against 4.17 for responsive. We report these ratings descriptively because each expert rated all four trajectories. Asked in free response whether anchoring is a recognizable pattern in experimental science, one reviewer wrote that "a couple did show quite firm anchoring, persisting with changing one tiny thing that clearly wasn't working", and another that researchers "will often make incremental adjustments within a familiar design space".

This is supporting evidence, not a validation of the construct. The sample is three reviewers, agreement is fair rather than strong (Fleiss' $\kappa = 0.33$), the four trajectories were curated to span the anchored and responsive range rather than sampled at random, the selection was not pre-registered, and five iterations cannot distinguish a trajectory that would later escape anchoring from one that would not. Anchoring was also defined in the instructions rather than discovered, so only the free-response answers speak to independent identification.

C Post-training details and arms

SFT uses 2,607 proposals collected from three frontier teachers across the 29 training tasks, rendered domain-agnostically, with the rationale field removed so the objective is the proposal itself. DPO is warm-started from that adapter, policy and frozen reference both, on 5,197 preference pairs built from real rollout histories pooled across the evaluated models. For each observation history we ask what GP-UCB would propose next, and emit a pair only where the oracle confirms that proposal beat what the rollout actually did. Pairs are therefore on-policy in their prefixes and oracle-verified in their labels. Both stages are LoRA at rank 16 on Llama 3.1 8B. SFT runs 2.5 epochs at learning rate 10^{-5} with a cosine schedule, batch size 2 and gradient accumulation 8, sequence length 8192. DPO runs 1 epoch at learning rate 5×10^{-6} with $\beta = 0.1$, a cosine schedule, batch size 1 and gradient accumulation 16, sequence length 8192. Both use warmup ratio 0.1. Four DPO seeds train from the single fine-tuned adapter and the arms differ only in that seed. Evaluation is on the 16 held-out tasks, four runs each.

SFT alone does not move the score. All four arms below are evaluated under domain-aware prompts on the held-out tasks, GP-UCB-normalized. The arm fine-tuned on domain-agnostically rendered data reaches -0.167 where the untrained base reaches -0.251, which is 6 of 11 tasks on the trajectory metric ($p = 0.45$) and 5 of 11 on the endpoint ($p = 0.39$). DPO on top of it reaches -0.050 and improves on it on 8 and 9 of 11 ($p = 0.062$, $p = 0.042$). Fine-tuning is the warm start the preference stage needs, and the preference stage is where the gain comes from.

The domain-agnostic evaluation is also null: fine-tuning did not reliably improve raw optimization even before the biological names were introduced. Evaluating the same adapter domain-agnostically on the held-out tasks, its median moves from -0.357 for the untrained base to -0.079, but the paired contrast does not clear zero, $+0.215$ [-0.047, $+0.500$] on 10 of 17 tasks ($p = 0.23$). Two smaller reruns of the same pair give -0.285 on 6 of 11 and $+0.035$ on 9 of 17, so the direction is not even stable across runs. Supervised fine-tuning on its own therefore does not reliably improve optimization in either condition.

Within the post-trained model at a fixed iteration-1 design, moving to domain-agnostic prompts is worth $+0.155$ on the endpoint metric [$+0.025$, $+0.304$] on 12 of 16 tasks ($p = 0.018$), and restricting to the oracle-aligned held-out tasks strengthens that gain (8 of 8, $p = 0.004$).

Test results for the figures in this section, kept out of their captions. The DPO arm gains $+0.40$ [$+0.19$, $+0.64$] under domain-agnostic prompts and $+0.03$ [-0.11, $+0.16$] under domain-aware prompts (Fig. 10c). Given one identical injected iteration-1 design, post-training helps under domain-agnostic prompts ($p = 0.037$) and does not under domain-aware ones ($p = 0.11$), and in the same-prior comparison Claude outperforms the untrained base ($p = 0.037$) while Claude and the DPO-trained 8B are indistinguishable ($p = 0.29$).

Task coverage differs across the matched-prior arms. Both post-trained arms cover all 17 evaluated tasks. The base arms were API-served and lost tasks to client errors. The domain-agnostic base covers 9 of the 10 seeded tasks and the domain-aware base 7, missing `adcp_target_phagocytosis` and `antibody_expression_stability_cho` in addition to the degenerate `baculovirus_titer_sf9` excluded throughout. The base-referenced contrasts therefore run at $n = 9$ and $n = 7$, while the within-model contrast, which needs no base arm, runs at $n = 16$. These counts concern the matched-prior experiments. The main post-training comparison in Fig. 10 uses 16 paired tasks in each condition. Unscorable proposals are rare and symmetric across conditions, under 0.6% of iterations in every arm, so no arm searches on a shortened effective budget.

Fig. 10 places base, DPO-trained and frontier models on one axis in both conditions, and Fig. 11 shows the same-prior comparison.

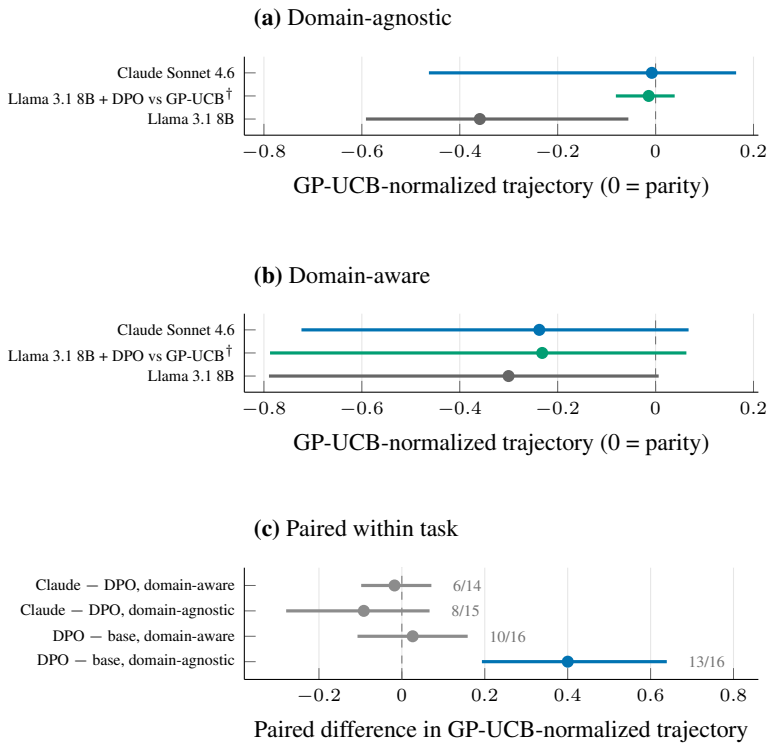

Figure 10: **Direct preference optimization against GP-UCB's choices lifts an 8B model to parity with it and to frontier-model performance, but only when the parameter names are not biologically meaningful.** GP-UCB-normalized trajectory score on 16 held-out biology tasks. Circles are medians over tasks, bars 95% bootstrap intervals on the median. Zero is GP-UCB parity in *(a)* and *(b)*, and no difference between arms in *(c)*. *(a)* Under domain-agnostic prompts the DPO-trained model reaches parity with an interval an order of magnitude tighter than either other arm. *(b)* Under domain-aware prompts the same three optimizers all fall well below parity and the intervals overlap completely, because task-to-task difficulty in this condition is far larger than the differences between arms. *(c)* Differencing within task removes that difficulty and is where the comparison can actually be read. DPO is worth $+0.40$ under domain-agnostic prompts (13 of 16 tasks) but $+0.03$ under domain-aware prompts, so the gain does not transfer. Claude and the DPO-trained 8B are indistinguishable in both conditions. Counts at the right are tasks favoring the first arm. †averaged over four training seeds, four runs per task each. The other arms are single models with four runs per task. One task (`baculovirus_titer_sf9`) is excluded, where a near-zero GP denominator sends the normalized score to $+32.8$ (see §C).

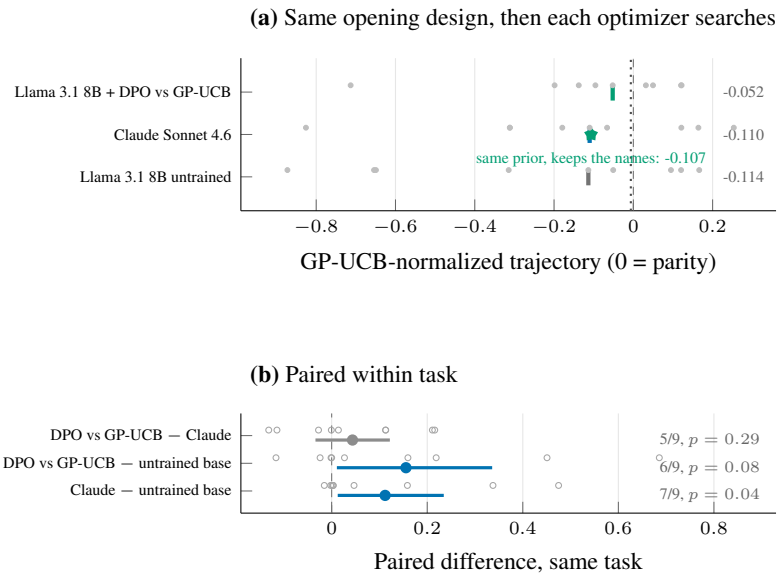

Figure 11: **Most of what looks like a frontier optimization advantage is the opening proposal, and the four arms largely converge once every optimizer is handed the same first design.** Every arm receives one identical Claude Sonnet 4.6 domain-aware iteration-1 design per task, injected at iteration 1, and then optimizes iterations 2–30 itself under domain-agnostic prompts, on nine held-out biology tasks. *(a)* One faint dot per task, with the arm's median marked. The dotted line is GP-UCB given the same opening. The four arms span 0.11 in median, compared with a spread more than twice that when each chooses its own start, and the untrained 8B moves from well below Claude to within 0.004 of it. *(b)* The paired differences this design permits, one circle per task with the mean, a 95% bootstrap interval, and a one-sided Wilcoxon. Differencing within task restores the ordering that the medians hide. Claude still outperforms the untrained base ($+0.112$, 7 of 9), which is the power check, while Claude and the DPO-trained 8B remain indistinguishable. The star marks the same Claude model given the same iteration-1 design but allowed to keep the biological names for iterations 2–30. That arm's median, -0.107, is indistinguishable from the -0.110 the same model scores under domain-agnostic prompts. With the prior held fixed the prompt effect is not visible between tasks and appears only when differenced within them, which is what Appendix B.2 does. Two arms are served locally and two through the API, so the Claude-versus-base contrast is the like-for-like one.