FEEDBACK-CALIBRATED PROTEIN OPTIMIZATION WITH BATCH-ALIGNED TAIL ARBITRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein optimization aims to discover high-fitness sequences under a limited experimental budget. Existing machine-learning methods use task-specific predictors, biological priors, or ranking-aware objectives to guide which variants are tested in the next experimental round. However, these methods cannot adapt to shifts in the reliability of predictive evidence as measurements accumulate and ensure the correct ranking of key high-fitness candidates. To address these challenges, we propose **Batch-Aligned Tail Arbitration (BATA)**, which uses experimental feedback to adaptively combine prior-informed and task-specific rankings for next-batch selection, with calibration focused on the batch-aligned high-fitness region. Across measured GB1, PABP, and TrpB landscapes, BATA achieves the best mean task rank (1.67) in final best fitness after 480 measurements. Controlled comparisons further show task-dependent gains from high-fitness calibration and batch alignment. Our work introduces feedback-calibrated predictor arbitration, where experimental feedback dynamically determines how predictive evidence guides next-batch selection, opening a new direction for protein optimization.

1 INTRODUCTION

Protein optimization aims to discover amino acid sequences with improved functional properties for applications such as therapeutics, industrial biotechnology, and environmental engineering (Notin et al., 2024; Listov et al., 2024). The performance of a protein for a target function is commonly measured by its *fitness*, and the relationship between protein sequences and their fitness values defines a *protein fitness landscape* (Freschlin et al., 2022). However, protein fitness landscapes are vast and often rugged, while the fitness of a candidate sequence can usually be determined only through costly experimental measurements (Yang et al., 2024; Sandhu et al., 2025). To reduce experimental costs under a limited assay budget, modern protein optimization commonly uses machine-learning models to predict fitness from available measurements and iteratively select promising variants for subsequent experiments (Rapp et al., 2024; Herrera-Rocha et al., 2025).

Existing machine-learning methods for protein optimization can be broadly grouped into three categories: (1) Active-learning and Bayesian-optimization methods iteratively train a fitness predictor on measured variants and use its predictions to select candidates for the next round of experiments (Yang et al., 2025; Bal et al., 2025). (2) Prior-guided methods incorporate evolutionary or generative sequence priors to guide the search when task-specific measurements are limited (Amin et al., 2025). (3) Ranking-aware methods focus more directly on ordering promising variants so that high-fitness candidates can be prioritized for experimental evaluation (Karimi et al., 2025; Xu et al., 2026). Despite their different designs, these approaches all rely on predictive evidence to guide which variants should be tested in the next experimental round.

However, the predictive evidence used by these methods is not equally reliable throughout an optimization: (1) When only a few experimental measurements are available, a task-specific predictor is trained on limited data and may be unreliable, especially on sparse or difficult fitness landscapes (Ghaffari et al., 2024; Li et al., 2025; Ibarraran et al., 2026). In this stage, broader biological priors can provide useful guidance. (2) As more measurements are collected, the task-specific predictor can become more informative, while the relative contribution of the biological prior may change across optimization rounds. (3) Even accurate prediction across all measured variants does not necessarily guarantee correct ranking among the key high-fitness candidates considered for the next

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

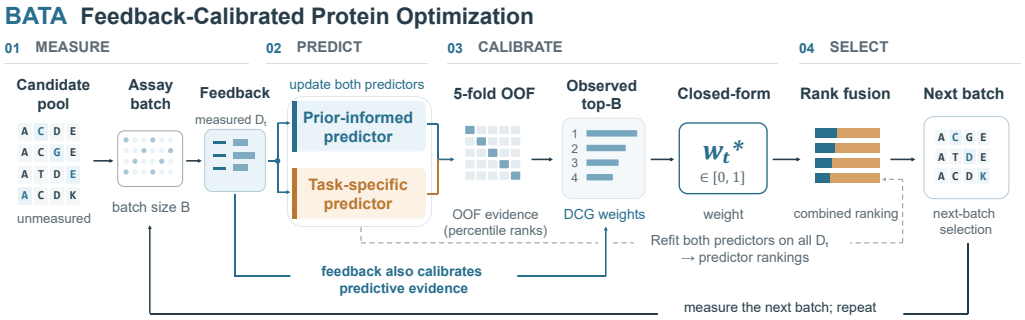


Figure 1: **BATA overview.** The prior-informed and task-specific predictors provide complementary predictive evidence. Five-fold out-of-fold (OOF) predictions assess both predictors on held-out measurements, and percentile ranks place their outputs on a common scale. A batch-sized rank cutoff and discounted cumulative gain (DCG) weights focus mixture calibration on the observed high-fitness region. After refitting on all measured variants, the learned mixture combines the two predictor rankings over unmeasured sequences for next-batch selection.

batch, which will decide the success of the next experiment. Therefore, the main challenge is to use experimental feedback to determine how much each source of predictive evidence should influence next-batch selection, while correctly ranking the most promising high-fitness candidates.

To address these challenges, we propose **Batch-Aligned Tail Arbitration (BATA)**, a method for feedback-calibrated protein optimization (Figure 1). For the first challenge, BATA retains both a prior-informed and task-specific predictor, allowing the biological prior to provide complementary guidance when task-specific measurements are still limited. Their predictions are converted to percentile ranks so that the two sources of evidence can be compared on a common scale. For the second challenge, BATA uses five-fold out-of-fold (OOF) predictions on measured variants to assess the current usefulness of each predictor. It then learns a mixture weight from the measured fitness values and updates this weight after every batch, allowing the relative contribution of the two predictors to change as experimental evidence accumulates. For the third challenge, BATA calibrates this mixture weight specifically on the highest-fitness measured variants instead of all observations. The calibration cutoff is set to the next batch size, and a logarithmically discounted cumulative gain (DCG) weight places greater emphasis on variants near the top of the fitness ranking. The obtained weighted rank objective has a closed-form solution, and the learned mixture is used to rank unmeasured sequences for the next experimental batch.

Our main contributions are summarized as follows:

(1) We formulate feedback-calibrated protein optimization as a decision problem in which experimental measurements determine not only how task-specific predictors are updated, but also how prior-informed and task-specific evidence should contribute to next-batch selection.

(2) We propose BATA, which evaluates both predictors using out-of-fold evidence, adaptively learns their mixture weight from measured fitness, and calibrates this weight on a batch-aligned high-fitness region using a DCG-weighted rank objective with a closed-form solution, to adjust the relative contribution of different evidence while prioritizing the promising high-fitness candidates.

(3) Across GB1, PABP, and TrpB, BATA achieves the best mean task rank (1.67) in final best fitness over 35 matched initializations per task, while controlled comparisons reveal when high-fitness calibration, batch alignment, and prior-informed evidence are useful. Our work extends the role of experimental feedback from updating predictors to arbitrating complementary predictive evidence, offering a new direction for feedback-calibrated protein optimization.

2 Related Work

Learning from experimental feedback. Machine-learning-assisted directed evolution uses measured protein variants to fit a fitness predictor and select candidates for the next experiment (Wu

et al., 2019; Yang et al., 2019). EVOLVEpro combines protein-language-model representations with task-specific regression, while ProSpero updates a fitness predictor from feedback to guide search beyond wild-type neighborhoods (Jiang et al., 2025; Kmicikiewicz et al., 2025). EvoMax further combines complementary models for protein evolution with sparse experimental data (Wan et al., 2026). These studies show how experimental feedback can improve predictions and guide search. BATA extends this feedback loop by using measured outcomes not only to update predictors but also to recalibrate how predictive evidence guides the next experimental batch.

Searching protein fitness landscapes. Protein optimization also benefits from methods that change the search space or the way it is explored. GGS smooths a learned fitness landscape, and LatProtRL uses reinforcement learning to search a continuous protein representation (Kirjner et al., 2024; Lee et al., 2024). KnowRLM incorporates biological knowledge into reinforced language modeling, while VLGPO combines a generative prior with fitness guidance in a latent-space optimization method (Wang et al., 2024; Bogensperger et al., 2025). In contrast, BATA leaves the representation and search space unchanged and instead focuses on how predictive evidence is combined within batch selection.

Priors and sequential decisions. Prior-guided optimization studies how existing knowledge can support a sequence of experimental decisions. In Bayesian optimization, DynMeanBO changes the influence of an expert prior, while ToSFiT and BOLT use language models for sequential optimization and transfer across tasks (Qu et al., 2026; Menet et al., 2026; Zeng et al., 2026). For proteins, Data Distillation and REAP learn candidate preferences or rankings to prioritize promising variants (Karimi et al., 2025; Xu et al., 2026). Building on these, BATA uses a prior-informed predictor to provide biological guidance when task-specific measurements are still limited.

3 Feedback-Calibrated Protein Optimization

Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be a finite protein sequence candidate pool and $f(x)$ its experimental fitness. At round t, the available data are

$$\mathcal{D}_t = \{(x_i, y_i) : x_i \in \mathcal{M}_t,\ y_i = f(x_i)\}, \tag{1}$$

where $\mathcal{M}_t$ contains the sequences measured so far. At round t, the optimizer selects a batch of B distinct unmeasured sequences, $\mathcal{B}_{t+1} \subseteq \mathcal{X} \setminus \mathcal{M}_t$ for the next experiment. Their fitness values are revealed together after the batch is selected and added to form $\mathcal{D}_{t+1}$. Because only B candidates can be tested in each experiment, feedback calibration should focus on the ranking of the most promising candidates near the top of the selection list.

The goal of the protein optimization task is to maximize the best observed fitness under a fixed measurement budget. We start with 96 measurements and add four batches of $B = 96$. The primary endpoint, Final@480, is the highest fitness found among these 480 measurements:

$$\text{Final@480} = \max_{x \in \mathcal{M}_4} f(x). \tag{2}$$

Thus, Final@480 is the best fitness found after 480 measurements. The secondary metric is the normalized area under the best-so-far fitness curve (Query-AUC), computed at batch endpoints from 96 to 480 measurements (Appendix A). A higher value means that the optimizer finds high-fitness variants earlier.

A prior-informed predictor and a task-specific predictor use different representations and models to fit the measured fitness values. They can therefore rank the same candidate sequences differently. We define *feedback calibration* as learning from the measured data how to combine these rankings. Because the next experiment can test only B sequences, calibration should emphasize the ordering of promising variants. This motivates evaluating the combined ranking on the observed high-fitness region at the scale of the next batch.

4 Batch-Aligned Tail Arbitration

Batch-Aligned Tail Arbitration (BATA) uses measured fitness to determine how much each predictor contributes to the next batch (Figure 1). Calibration focuses on the high-fitness end, or tail, of the

observed landscape. At each round, BATA obtains held-out predictions, learns a mixture weight, and uses that weight to combine rankings of unmeasured sequences.

4.1 PRIOR-INFORMED AND TASK-SPECIFIC PREDICTORS

The prior-informed predictor uses direct coupling analysis (DCA) features, which represent evolutionary sequence statistics (Morcos et al., 2011). A ridge-regression ensemble, consisting of several fitted linear models, maps these features to fitness. The task-specific predictor uses one-hot or protein-language-model representations with the prediction model specified for each benchmark (Appendix A). Both predictors are fitted to the same measured variants and updated after each batch. The prior-informed predictor provides evolutionary guidance when task-specific measurements are limited, whereas the task-specific predictor captures information revealed by accumulated experimental feedback. Thus, evolutionary information enters through the prior-informed representation, while experimental measurements train both predictors.

To assess the predictors on held-out measurements, we divide the measured variants into five folds. Each fold is predicted by models fitted on the other four folds. Averaging the ensemble members gives one out-of-fold (OOF) prediction per variant for each predictor. Thus, a variant's measured fitness is used to assess a model that was not trained on that variant. This provides an unbiased estimate of each predictor's current ranking ability for feedback calibration. We then convert the measured fitness values and both sets of OOF predictions to percentile ranks.

$$r_i^y = \rho(y_i), \qquad r_i^p = \rho(\hat{y}_i^{\text{prior}}), \qquad r_i^a = \rho(\hat{y}_i^{\text{task}}). \tag{3}$$

Here ρ is the ascending rank divided by the number of values; tied values receive their average rank. Higher fitness or predicted scores therefore give larger percentile ranks. The superscripts y, p, and a denote measured fitness, the prior-informed predictor, and the task-specific predictor, respectively. This transformation puts the two predictors on the same scale.

4.2 CALIBRATION ON THE OBSERVED HIGH-FITNESS REGION

Only a small batch can be measured next, so BATA fits the mixture to the highest-fitness measured variants. Let $n_t = |\mathcal{D}_t|$, $k_t = \min(B, n_t)$, and p_i be variant i's descending fitness rank, with average ranks for ties. Discounted cumulative gain (DCG) is a ranking measure that reduces the contribution of lower-ranked items (Järvelin & Kekäläinen, 2002). We use its logarithmic discount to define the calibration weights:

$$q_i = \begin{cases} 1/\log_2(p_i + 1), & p_i \le k_t, \\ 0, & p_i > k_t. \end{cases} \tag{4}$$

The cutoff k_t matches the next batch size whenever at least B measurements are available. Within this region, higher-fitness variants receive larger weights; variants below the cutoff receive zero weight. A tie at the boundary can change the number of weighted variants because p_i is an average rank. In our protocol, all 96 initial variants are weighted, and later rounds focus on a smaller fraction of the growing measured set.

Let $w \in [0, 1]$ be the weight of the prior-informed predictor and $1 - w$ the weight of the task-specific predictor. Their combined rank score is $r_i(w) = wr_i^p + (1-w)r_i^a$. We choose w to identify the predictor combination whose ranking best matches the experimentally observed ordering of promising variants.

$$\mathcal{L}_{\text{BATA}}(w) = \sum_{i=1}^{n_t} q_i \left[r_i^y - wr_i^p - (1-w)r_i^a \right]^2. \tag{5}$$

The loss is the squared difference between measured fitness ranks and combined predictor scores, weighted by q_i. Thus, DCG supplies the sample weights for a squared rank-error objective.

4.3 CLOSED-FORM WEIGHT AND CANDIDATE SELECTION

Define the predictor difference $d_i = r_i^p - r_i^a$ and the task-specific rank error $z_i = r_i^y - r_i^a$. Substituting these quantities into the loss gives $\sum_i q_i(z_i - wd_i)^2$, a convex quadratic in one variable. The

Algorithm 1 BATA for sequential protein optimization

Require: Candidate pool $\mathcal{X}$, initial data $\mathcal{D}_0$, batch size B, rounds T, selection rule $\mathcal{A}$

1: **for** $t = 0, \ldots, T-1$ **do**
2: Obtain five-fold OOF predictions from both predictors on $\mathcal{D}_t$.
3: Compute percentile ranks r^y, r^p, r^a and weights q (Eqs. 3–4).
4: Compute w_t^* by Eq. 6, with the stated numerical convention.
5: Refit both predictors on $\mathcal{D}_t$ and compute weighted pool rankings by Eq. 7.
6: Select $\mathcal{B}_{t+1} = \mathcal{A}(s, B) \subseteq \mathcal{X} \setminus \mathcal{M}_t$.
7: Reveal the full batch and add its measurements to form $\mathcal{D}_{t+1}$.
8: **end for**
9: **return** The highest-fitness measured variant.

minimizing weight over $[0, 1]$ is

$$w_t^* = \Pi_{[0,1]}\left(\frac{\sum_i q_i d_i z_i}{\sum_i q_i d_i^2}\right), \tag{6}$$

where $\Pi_{[0,1]}$ clips the ratio to $[0, 1]$. If the denominator is zero, we set $w_t^* = 0$; the implementation also uses zero when the denominator is no greater than the float64 machine epsilon. A larger w_t^* gives more influence to the prior-informed predictor, while a smaller value favors the task-specific predictor. Because w_t^* is re-estimated after each batch using the newly available measurements, BATA can adapt the relative contribution of the two predictors as measurements accumulate.

After estimating w_t^* from the OOF predictions, we refit both predictors on all measured variants before candidate selection. Thus, OOF predictions are used to calibrate the mixture weight, while the refitted predictors use all currently available measurements to rank the unmeasured sequences. Let $g_{t,m}^{\text{prior}}$ and $g_{t,m}^{\text{task}}$ denote the prediction functions of ensemble member m at round t. For each unmeasured sequence x_j, we convert the predictions from each member to percentile ranks over the unmeasured pool and combine the two rankings using w_t^*:

$$s_{j,m} = w_t^* \rho\big(g_{t,m}^{\text{prior}}(x_j)\big) + (1 - w_t^*)\rho\big(g_{t,m}^{\text{task}}(x_j)\big). \tag{7}$$

When a predictor has only one model, its rank vector is reused for every member of the other ensemble. GB1 and TrpB use Thompson-style selection (Russo et al., 2018): sample a member for each choice and select the remaining sequence with its highest combined score. PABP selects the highest mean scores across members. Candidates are selected without replacement, and fitness is revealed only after the full batch is chosen.

4.4 Properties of the calibration rule

Proposition 1 (Optimality for the observed rank loss)**.** *For fixed ranks and nonnegative q_i, Eq. 6 uniquely minimizes Eq. 5 over $[0, 1]$ when $\sum_i q_i d_i^2 > 0$. If this sum is zero, the loss is constant and $w_t^* = 0$ is a minimizer.*

Proposition 2 (Score-scale invariance)**.** *With a fixed tie convention, strictly increasing transformations of either predictor's fixed OOF and pool scores leave the calibration weight and fused percentile-rank scores unchanged.*

Proposition 1 gives the minimum calibration loss for the observed data. Proposition 2 shows that changing score units while preserving their order leaves calibration and combined pool scores unchanged. The proofs are given in Appendix D.

5 Experiments

5.1 Experimental setup

We evaluate BATA by replaying sequential optimization on the measured GB1, PABP, and TrpB fitness landscapes (Melamed et al., 2013; Wu et al., 2016; Johnston et al., 2024). Each campaign begins with 96 measured variants and selects four additional batches of 96, giving 480 unique measurements. At each round, model fitting, calibration, and candidate selection use only the fitness

Table 1: **Main optimization results.** Best observed fitness after 480 measurements (Final@480), reported as mean ± sample standard deviation over 35 matched initializations per task. The last columns average task-level ranks for Final and discovery speed (Query-AUC). Higher fitness and lower mean ranks are better; bold marks the best values.

Method	GB1	PABP	TrpB	Final rank	AUC rank
BATA	0.8512 ± 0.1083	$\mathbf{2.4169 \pm 0.4232}$	$\mathbf{0.9209 \pm 0.0422}$	**1.67**	**2.00**
ALDE	$\mathbf{0.9347 \pm 0.0904}$	1.8183 ± 0.1814	0.8958 ± 0.1099	3.00	2.67
EVOLVEpro-650M	0.7344 ± 0.0966	2.3676 ± 0.4745	0.8982 ± 0.0839	3.00	3.00
REAP100-650M	0.8958 ± 0.0944	1.9960 ± 0.2371	0.8858 ± 0.1177	3.33	3.67
RF	0.8294 ± 0.1070	1.9849 ± 0.2533	0.8934 ± 0.1224	4.00	3.67

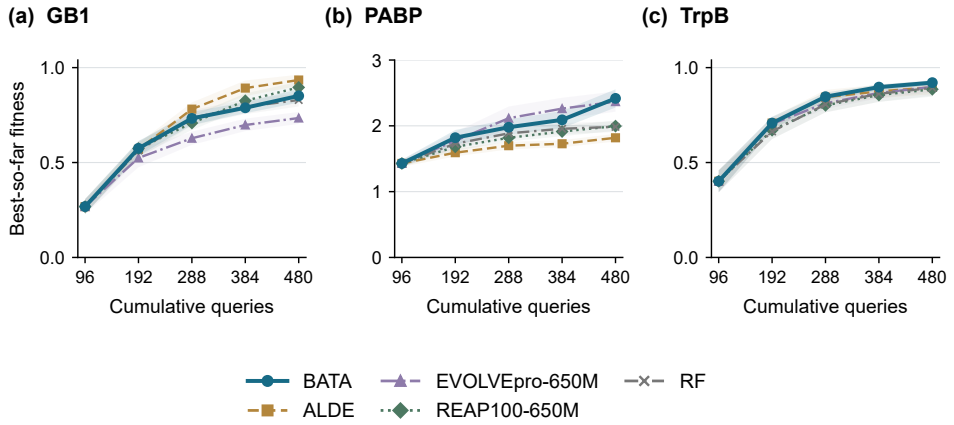


Figure 2: **Optimization trajectories.** Mean best-so-far fitness and pointwise 95% bootstrap confidence intervals over 35 Main-35 runs per method and task. Points mark completed batches. Vertical axes use each task's fitness scale.

values revealed so far. Final@480 measures the best fitness found after 480 measurements, while Query-AUC measures discovery speed. The main evaluation, Main-35, uses 35 matched initialization sets per task. A separate sensitivity evaluation uses 70 initializations for GB1 and TrpB while retaining the same 35 PABP initializations.

We compare BATA with ALDE, EVOLVEpro-650M, REAP100-650M, and random-forest (RF) selection (Jiang et al., 2025; Yang et al., 2025; Xu et al., 2026). Within each task and evaluation cohort, all methods share the same initial measured variants and the 480-measurement budget. BATA uses a DCA-based ridge-regression ensemble as its prior-informed predictor. Its task-specific predictor uses one-hot features on GB1 and ESM2-650M features with random forests on PABP and TrpB. Appendix A details the model sizes, selection rules, and uncertainty analyses.

5.2 Optimization quality and discovery speed

BATA achieves the best mean Final@480 rank across the three Main-35 tasks (1.67; Table 1), with the highest mean Final@480 on PABP and TrpB, while ALDE leads on GB1. BATA also achieves the best mean Query-AUC rank (2.00). Figure 2 further separates endpoint quality from discovery speed: on PABP, EVOLVEpro achieves higher Query-AUC, whereas BATA reaches higher Final@480. These results show that BATA achieves the strongest aggregate endpoint performance, while discovery speed remains task dependent. Appendix B reports the complete Query-AUC results and campaign-level distributions.

5.3 Sensitivity to initialization cohorts

We next examine how method rankings change across initialization cohorts (Figure 3). In the larger sensitivity evaluation, BATA and REAP100 tie for the best mean Final@480 rank (2.00), while

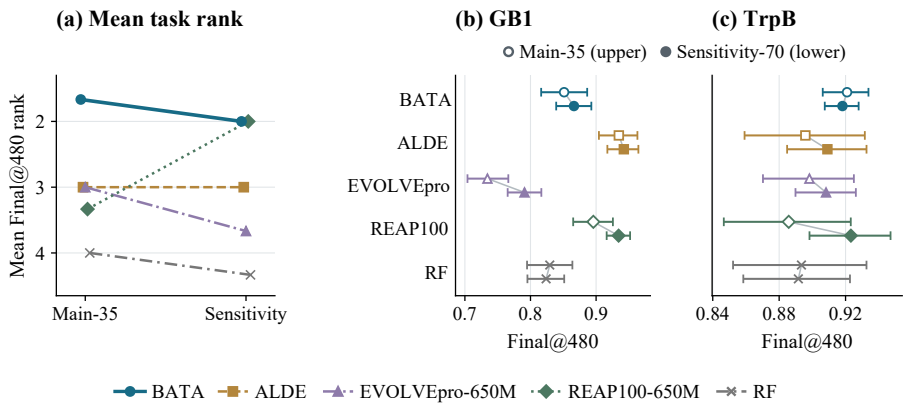


Figure 3: **Initialization sensitivity.** (a) Mean Final@480 ranks in Main-35 and the sensitivity setting, which uses 70 initializations for GB1 and TrpB and retains 35 for PABP. (b,c) Mean Final@480 and 95% bootstrap intervals for each cohort. Open markers denote Main-35 and filled markers denote the sensitivity cohort. Colors and marker shapes identify methods. Gray lines connect cohort-level summaries and do not represent paired changes.

ALDE remains best on GB1. On TrpB, REAP100 achieves a slightly higher mean Final@480 than BATA (0.9232 versus 0.9181), whereas BATA has higher Query-AUC (0.7911 versus 0.7687). These results show that method rankings depend on the initialization cohort, while BATA remains tied for the best aggregate Final@480 rank in the larger evaluation. Figure 3 compares cohort-level summaries rather than paired changes between cohorts. Appendix B reports the sensitivity results.

5.4 How does feedback change the mixture?

The prior-informed predictor receives less weight as measurements accumulate on GB1 and TrpB (Figure 4). Across the four decisions, its mean weight decreases from 0.466 to 0.089 on GB1 and from 0.518 to 0.163 on TrpB. The final weight is lower than the initial weight in 33 of 35 GB1 campaigns and 34 of 35 TrpB campaigns. PABP shows a smaller mean change, from 0.460 to 0.403, with 19 decreases and 16 increases. These results show that BATA adapts the predictor balance as experimental feedback accumulates, with stronger shifts toward the task-specific predictor on GB1 and TrpB than on PABP.

5.5 Why batch-aligned tail calibration?

We compare four calibration objectives to examine three design choices in BATA: whether calibration should focus on the high-fitness region, whether variants near the top should receive greater weight, and whether the calibration cutoff should match the next batch size. G-Rank uses rank error over all measured variants, T-Uniform-96 restricts calibration to the top 96 with uniform weights, T-DCG-96 applies DCG weighting at the batch-sized cutoff used by BATA, and T-DCG-192 uses the same weighting with a larger cutoff. Predictor configurations, five-fold OOF predictions, rank combination, selection rules, initialization identities, and measurement budgets are fixed. Each task has 24 matched groups, giving 288 runs across the three tasks and four objectives.

T-DCG-96 achieves the best mean task rank for both Final@480 and Query-AUC (1.33; Figure 5b). On TrpB, it improves Final@480 over G-Rank by 0.0256 and over T-DCG-192 by 0.0115. On PABP, T-DCG-96 improves over uniform top-96 weighting by 0.1719. GB1 shows little endpoint difference between the top-96 and top-192 cutoffs (-0.0006). These results show that tail-focused calibration, DCG weighting, and batch alignment provide task-dependent benefits, while their combination gives the best aggregate rank across the three landscapes. Figure 5a shows all paired comparisons, with uncertainty analyses reported in Appendix B.

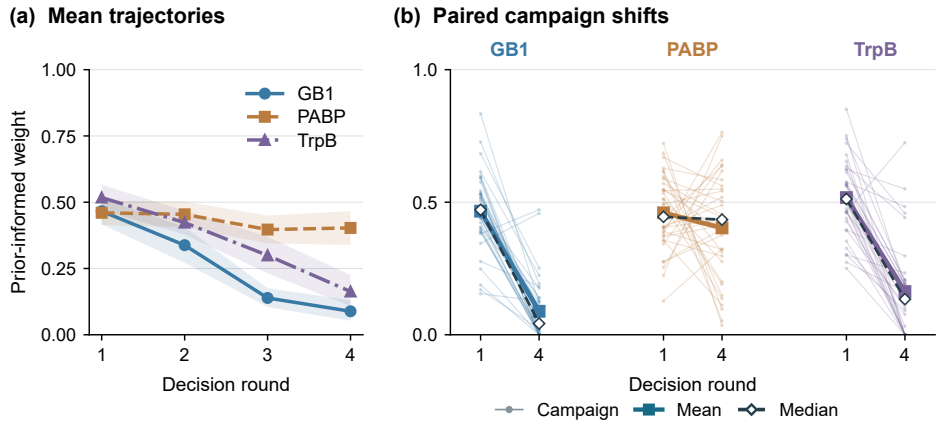
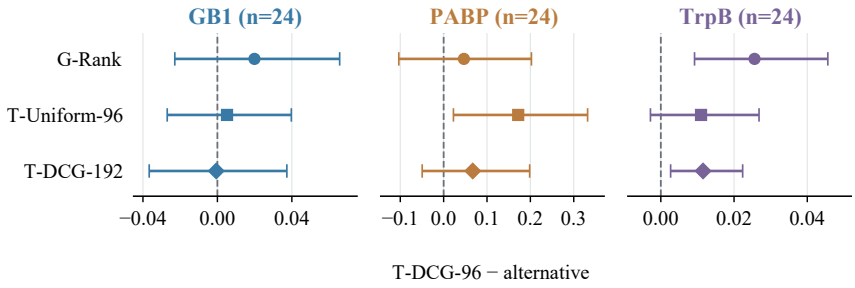


Figure 4: **Predictor weights change with feedback.** (a) Mean prior-informed weight and pointwise 95% bootstrap intervals across 35 Main-35 runs per task. (b) First and last weights for each run, connected by light lines; squares and open diamonds mark means and medians. Lower values assign more weight to the task-specific predictor.

(a) Paired Final@480 differences



(b) Mean task ranks

	G-Rank	T-Uniform-96	**T-DCG-96**	T-DCG-192
Final@480	3.33	3.00	**1.33**	2.33
Query-AUC	3.67	3.00	**1.33**	2.00

Figure 5: **Calibration objectives.** (a) All nine paired Final@480 differences between T-DCG-96 (BATA) and the alternative objectives, with the study's 95% bootstrap intervals over 24 matched groups per task. T-DCG-96 applies discounted cumulative gain (DCG) weights at a measured-fitness rank cutoff of 96. Ties use average ranks, so the number of weighted variants can differ from 96. Positive differences favor T-DCG-96; separate task axes retain the native fitness units. (b) Mean task ranks for Final@480 and Query-AUC from the same 288 optimization runs.

5.6 Comparison with task-specific-only configurations

We first compare BATA with Fine-only M20, a larger task-specific configuration that uses a 20-member ensemble without the prior-informed predictor (Figure 6). This comparison is not a single-factor ablation: Fine-only M20 also changes ensemble capacity and, on some tasks, the selection rule. GB1 and PABP each use 35 matched pairs, HIS7 and GRB2 each use 24, and TrpB uses a separate 70-initialization cohort. On TrpB, Fine-only M20 achieves a higher mean Final@480 than BATA (0.9388 versus 0.9181), with a mean paired advantage of 0.0207. BATA achieves higher mean Final@480 on GB1, PABP, HIS7, and GRB2.

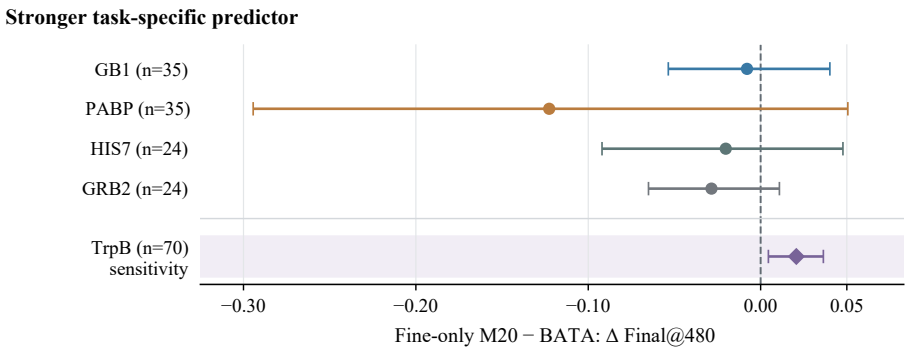


Figure 6: **Stronger task-specific predictor.** Paired mean Final@480 differences (Fine-only M20 minus BATA) with 95% confidence intervals. Positive values favor Fine-only M20. GB1 and PABP use $n = 35$, HIS7 and GRB2 use $n = 24$, and the separate TrpB sensitivity cohort uses $n = 70$. Fine-only M20 uses a 20-member task-specific ensemble, removes the prior-informed predictor, and changes the selection rule on some tasks.

To isolate the predictor-combination mechanism more directly, Appendix C keeps the task-specific predictor capacity and selection rule fixed. Under these controls, BATA improves over equal-rank combination on TrpB, while its advantage over task-specific-only prediction is less conclusive. A separate switching experiment further shows that a low prior-informed weight alone does not reliably identify when the larger task-specific ensemble will improve optimization. These results indicate that the value of prior-informed evidence depends on the task-specific predictor configuration as well as the learned mixture weight.

6 Discussion and Limitations

Our results show that the relative value of prior-informed and task-specific evidence changes with both the available measurements and the predictor configuration. On GB1 and TrpB, BATA progressively shifts weight toward the task-specific predictor as measurements accumulate, whereas PABP shows a smaller and less consistent shift. The calibration-objective comparisons further show task-dependent benefits from focusing on the high-fitness region, emphasizing higher-ranked variants, and aligning the cutoff with the next batch. These results indicate that feedback calibration is useful for adapting a fixed pair of predictors, while the learned mixture weight should not be treated as a general criterion for choosing between different predictor configurations.

Our evaluation shows optimization on measured fitness landscapes under matched initialization and budget settings. Prospective laboratory optimization would additionally need to account for measurement noise, assay variation, and other experimental constraints. BATA currently relies on a DCA-based prior-informed predictor, a single mixture weight per round, and repeated five-fold OOF fitting. Extending the framework to richer biological priors, more flexible arbitration, and lower-cost calibration may improve its applicability to experimental protein optimization.

7 Conclusion

In this work, we propose BATA for feedback-calibrated protein optimization under a limited measurement budget. BATA uses held-out experimental evidence to adaptively combine prior-informed and task-specific rankings, with calibration focused on the high-fitness region relevant to the next batch. Across GB1, PABP, and TrpB, BATA achieves the best mean Final@480 rank in Main-35, while controlled comparisons reveal task-dependent benefits from tail-focused calibration, batch alignment, and prior-informed evidence. Our work extends the role of experimental feedback from updating predictors to determining how complementary predictive evidence guides the next experimental decision, opening a new direction for protein optimization.

AI USE STATEMENT

Generative AI tools were used to assist literature retrieval, research ideation and experimental workflows, manuscript drafting and revision, software review, and figure preparation. All AI-assisted research outputs, code, experimental results, citations, and interpretations were reviewed and verified by the authors. All reported numerical results originate from recorded model evaluations and protein fitness-landscape analyses. The authors take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendices A–D provide the evaluation protocols, full results, and proofs. Experiment records specify the configurations, initializations, seeds, measured batches, and source data.

REFERENCES

Alan Amin, Nate Gruver, Yilun Kuang, Yucen Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6cdd4ce9330025967dd1ed0bed3010f5-Abstract-Conference.html.

Melis Ilayda Bal, Pier Giuseppe Sessa, Mojmir Mutny, and Andreas Krause. Optimistic games for combinatorial Bayesian optimization with application to protein design. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/434743b59a2ef42d657e61e06f0dd97b-Abstract-Conference.html.

Lea Bogensperger, Dominik Narnhofer, Ahmed Allam, Konrad Schindler, and Michael Krauthammer. A variational perspective on generative protein fitness optimization. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4700–4712. PMLR, 2025. URL https://proceedings.mlr.press/v267/bogensperger25a.html.

Chase R. Freschlin, Sarah A. Fahlberg, and Philip A. Romero. Machine learning to navigate fitness landscapes for protein engineering. *Current Opinion in Biotechnology*, 75:102713, 2022. doi: 10.1016/j.copbio.2022.102713.

Saba Ghaffari, Ehsan Saleh, Alex Schwing, Yu-Xiong Wang, Martin Burke, and Saurabh Sinha. Robust model-based optimization for challenging fitness landscapes. In *ICLR*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f84ceafb242f0de36f8c49452fbae6de-Abstract-Conference.html.

Fabio Herrera-Rocha, David Medina-Ortiz, Fabian Mauz, Juergen Pleiss, and Mehdi D. Davari. Best practices for machine learning-assisted protein engineering. *Journal of Chemical Information and Modeling*, 65(23):12655–12667, 2025. doi: 10.1021/acs.jcim.5c01983.

Sebastian Ibarraran, Shriram Chennakesavalu, Frank Hu, and Grant M. Rotskoff. Efficient, Few-Shot Directed Evolution with Energy Rank Alignment. *Journal of Chemical Information and Modeling*, 66(17):10609–10621, 2026. doi: 10.1021/acs.jcim.6c01333. URL https://doi.org/10.1021/acs.jcim.6c01333.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R. Sgrizzi, Lukas Villiger, Alisan Kayabolen, B. J. Kim, Josephine K. Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, Jonathan S. Gootenberg, and Omar O. Abudayyeh. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, 387(6732):eadr6006, 2025. doi: 10.1126/science.adr6006. URL https://doi.org/10.1126/science.adr6006.

Kadina E. Johnston, Patrick J. Almhjell, Ella J. Watkins-Dulaney, Grace Liu, Nicholas J. Porter, Jason Yang, and Frances H. Arnold. A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proceedings of the National Academy of Sciences*, 121(32):e2400439121, 2024. doi: 10.1073/pnas.2400439121. URL https://doi.org/10.1073/pnas.2400439121.

Mostafa Karimi, Sharmi Banerjee, Tommi Jaakkola, Bella Dubrov, Shang Shang, and Ron Benson. Data distillation for extrapolative protein design through exact preference optimization. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a9ea92ef18aae17627d133534209e640-Abstract-Conference.html.

Andrew Kirjner, Jason Yim, Raman Samusevich, Shahar Bracha, Tommi Jaakkola, Regina Barzilay, and Ila Fiete. Improving protein optimization with smoothed fitness landscapes. In *ICLR*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/cbb7a23649b25001e797a726cf75498e-Abstract-Conference.html.

Michal Kmicikiewicz, Vincent Fortuin, and Ewa Szczurek. ProSpero: Active Learning for Robust Protein Design Beyond Wild-Type Neighborhoods. In *NeurIPS*, volume 38, pp. 131015–131048, 2025. doi: 10.52202/085713-4364. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/bde1c75d7d89e64d6d93dce1c68dc3d8-Abstract-Conference.html.

Minji Lee, Luiz Felipe Vecchietti, Hyunkyu Jung, Hyun Joo Ro, Meeyoung Cha, and Ho Min Kim. Robust optimization in protein fitness landscapes using reinforcement learning in latent space. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26976–26990. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24x.html.

Francesca-Zhoufan Li, Jason Yang, Kadina E. Johnston, Emre Gürsoy, Yisong Yue, and Frances H. Arnold. Evaluation of machine learning-assisted directed evolution across diverse combinatorial landscapes. *Cell Systems*, 16(9):101387, 2025. doi: 10.1016/j.cels.2025.101387. URL https://doi.org/10.1016/j.cels.2025.101387.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://doi.org/10.1126/science.ade2574.

Dina Listov, Casper A. Goverde, Bruno E. Correia, and Sarel Jacob Fleishman. Opportunities and challenges in design and optimization of protein function. *Nature Reviews Molecular Cell Biology*, 25(8):639–653, 2024. doi: 10.1038/s41580-024-00718-y.

Daniel Melamed, David L. Young, Caitlin E. Gamble, Christina R. Miller, and Stanley Fields. Deep mutational scanning of an RRM domain of the Saccharomyces cerevisiae poly(A)-binding protein. *RNA*, 19(11):1537–1551, 2013. doi: 10.1261/rna.040709.113. URL https://doi.org/10.1261/rna.040709.113.

Nicolas Menet, Aleksandar Terzic, Michael Hersche, Andreas Krause, and Abbas Rahimi. Thompson sampling via fine-tuning of LLMs. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/707a2d58641b2192203b4bf4c532cfe1-Abstract-Conference.html.

Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S. Marks, Chris Sander, Riccardo Zecchina, José N. Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011. doi: 10.1073/pnas.1111471108. URL https://doi.org/10.1073/pnas.1111471108.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In *NeurIPS*, volume 36, pp. 64331–64379, 2023. doi: 10.52202

/075280-2810. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/cac723e5ff29f65e3fcbb0739ae91bee-Abstract-Datasets_and_Benchmarks.html.

Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nature Biotechnology*, 42(2):216–228, 2024. doi: 10.1038/s41587-024-02127-0.

Chongqi Qu, Meiqin Liu, Jian Lan, Shanling Dong, and Zhunga Liu. Incorporating expert priors into Bayesian optimization via dynamic mean decay. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/020e313d40a7c060ed07a10cef287750-Abstract-Conference.html.

Jacob T. Rapp, Bennett J. Bremer, and Philip A. Romero. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nature Chemical Engineering*, 1(1):97–107, 2024. doi: 10.1038/s44286-023-00002-4.

Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A Tutorial on Thompson Sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018. doi: 10.1561/2200000070. URL https://doi.org/10.1561/2200000070.

Mahakaran Sandhu, John Z. Chen, Dana S. Matthews, Matthew A. Spence, Sacha B. Pulsford, Barnabas Gall, Joe A. Kaczmarski, James Nichols, Nobuhiko Tokuriki, and Colin J. Jackson. Computational and experimental exploration of protein fitness landscapes: Navigating smooth and rugged terrains. *Biochemistry*, 64(8):1673–1684, 2025. doi: 10.1021/acs.biochem.4c00673.

Shijie Wan, Jackson Gold, Pranay Vure, Casey S. Mogilevsky, Ananya Talikoti, Tianrong Chen, Aman Gupta, Trisha Biswas, Zheng You, Vir Acharya, Pranam Chatterjee, Xiao Wang, and Xue Gao. Adaptive model-guided protein evolution with sparse data optimizes compact eukaryotic genome editors. *Nature Biotechnology*, 2026. doi: 10.1038/s41587-026-03272-4. URL https://doi.org/10.1038/s41587-026-03272-4.

Yuhao Wang, Qiang Zhang, Ming Qin, Xiang Zhuang, Xiaotong Li, Zhichen Gong, Zeyuan Wang, Yu Zhao, Jianhua Yao, Keyan Ding, and Huajun Chen. Knowledge-aware reinforced language models for protein directed evolution. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52260–52273. PMLR, 2024. URL https://proceedings.mlr.press/v235/wang24cq.html.

Nicholas C Wu, Lei Dai, C Anders Olson, James O Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5:e16965, 2016. doi: 10.7554/elife.16965. URL https://doi.org/10.7554/elife.16965.

Zachary Wu, S. B. Jennifer Kan, Russell D. Lewis, Bruce J. Wittmann, and Frances H. Arnold. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, 2019. doi: 10.1073/pnas.1901979116. URL https://doi.org/10.1073/pnas.1901979116.

Jingyi Xu, Yan Zheng, Rajamanikandan Sundarraj, Kenneth Woycechowsky, Zhiguang Yuchi, and Yingjin Yuan. Rank-guided learning accelerates automated enzyme engineering. *Nature Communications*, 17:9584, 2026. doi: 10.1038/s41467-026-76264-2. URL https://doi.org/10.1038/s41467-026-76264-2.

Jason Yang, Francesca-Zhoufan Li, and Frances H. Arnold. Opportunities and challenges for machine learning-assisted enzyme engineering. *ACS Central Science*, 10(2):226–241, 2024. doi: 10.1021/acscentsci.3c01275.

Jason Yang, Ravi G. Lal, James C. Bowden, Raul Astudillo, Mikhail A. Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H. Arnold. Active learning-assisted directed evolution. *Nature Communications*, 16(1):714, 2025. doi: 10.1038/s41467-025-55987-8. URL https://doi.org/10.1038/s41467-025-55987-8.

Kevin K. Yang, Zachary Wu, and Frances H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019. doi: 10.1038/s41592-019-0496-6. URL https://doi.org/10.1038/s41592-019-0496-6.

Yimeng Zeng, Natalie Maus, Haydn Jones, Jeffrey Tao, Fangping Wan, Marcelo Der Torossian Torres, Cesar de la Fuente, Ryan Marcus, Osbert Bastani, and Jacob Gardner. Scaling multi-task Bayesian optimization with large language models. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a4a3689e5ca538e340f58cdd29f513a9-Abstract-Conference.html.

A Reproducibility and Protocol Details

A.1 Measurements and metrics

Each optimization run starts with 96 measured variants and selects four additional batches of 96 distinct variants. The simulator reveals a batch's fitness values after all its candidates have been chosen. The optimizer excludes previously measured variants and uses only the currently available measurements for fitting, cross-validation, calibration, and selection. A complete run contains 480 unique measurements.

Let b_q denote the best fitness observed after q measurements. Final@480 equals b_{480}. The normalized area under the best-so-far fitness curve, Query-AUC, is

$$\text{Query-AUC} = \frac{b_{96} + 2(b_{192} + b_{288} + b_{384}) + b_{480}}{8}. \tag{8}$$

This formula applies the trapezoidal rule to batch endpoints from 96 to 480 measurements and divides by the query range. Each run contributes one Final value and one Query-AUC value. Lines in the trajectory plots connect completed batches; measurements within a batch arrive at the same time.

A.2 Sequence data and predictors

Each task uses the candidate order and fitness scale in its recorded data release. PABP, HIS7, and GRB2 use the fixed ProteinGym v1.3 substitution tables (Notin et al., 2023). GB1 uses the scale-to-maximum table from the energy-rank-alignment release, and TrpB uses the fitness table distributed with ALDE. Input identities are checked against the experiment records before replay. All candidates in these recorded inputs are retained, with no additional filtering based on outcomes.

The prior-informed predictor contains five ridge-regression models fitted to direct coupling analysis (DCA) features. Features and fitness values are standardized using the fitting subset. The regularization parameter is selected from 17 logarithmically spaced values between 10^{-4} and 10^{4}, and five models are fitted to bootstrap samples of the training data. Table A1 lists the task-specific configurations. ESM2-650M provides fixed protein-language-model representations (Lin et al., 2023); only the downstream fitness predictor is refitted after each batch.

Table A1: Task-specific predictor configurations for the main BATA comparison. Every task uses five DCA-based ridge-regression models as the prior-informed predictor.

Task	Representation	Task-specific predictor	Selection
GB1	One-hot	Five-member Ridge ensemble	Thompson-style
PABP	ESM2-650M	One random forest, 100 trees	Greedy mean
TrpB	ESM2-650M	Five bootstrap forests, 100 trees each	Thompson-style

Five-fold out-of-fold (OOF) prediction starts from a seeded permutation of measured-variant indices. Each fold is predicted by models fitted on the other four folds. Ensemble predictions are averaged before conversion to OOF ranks. For candidate selection, each refitted model is ranked separately over the unmeasured pool. On PABP, the single task-specific rank vector is paired with each of the five prior-informed rank vectors. Greedy selection uses mean combined scores; Thompson-style selection samples one member for each choice. Both exclude candidates already selected and break score ties by candidate-index order.

Percentile ranks are average ascending ranks divided by the number of values. Descending fitness ranks in Eq. 4 also average ties. Consequently, a tied group at the cutoff can change the number of weighted observations, while the rank cutoff remains $\min(B, n_t)$. The implementation sets the mixture weight to zero when the denominator in Eq. 6 is no greater than float64 machine epsilon. This numerical threshold extends the exact zero-denominator convention in Proposition 1.

A.3 SEPARATE EVALUATION COHORTS

Main-35 contains 525 runs: five methods evaluated on three tasks with 35 initializations each. The sensitivity analysis uses 700 additional run records for five methods on GB1 and TrpB with 70 initializations each; PABP retains its original 35. Methods are matched within each task and initialization set. Figure 3 compares summaries of these sets, so its connecting lines are not paired run differences. Aggregate ranks are obtained by ranking method means within each task and then averaging across tasks.

The calibration-loss comparison uses four objectives and 24 matched groups per task, totaling 288 runs. Predictor configurations, five-fold OOF fitting, percentile ranks, selection rules, initializations, and budgets are fixed. The 72 T-DCG-96 trajectories are reused after confirming exact agreement with the current procedure. The same-capacity controls, larger Fine-only M20 ensemble, and threshold-based switching test each use separate recorded initialization sets, as specified in Appendix C.

A.4 UNCERTAINTY AND REPRODUCIBILITY CHECKS

Curve bands are pointwise 95% percentile-bootstrap intervals obtained by resampling complete runs. Paired comparisons resample matched run differences within each task. Figure 5 uses the recorded objective-study intervals. The same-capacity analysis uses the prespecified 20,000 resamples; its TrpB task-specific-only interval also includes zero with 100,000 resamples. Standard deviations describe variation across runs, while confidence intervals describe uncertainty in means or mean differences. Paired differences, not overlap between pointwise curve bands, are used to assess paired comparisons.

The experiment records store candidate identities, initialization sets, algorithm seeds, revealed batches, and source-file hashes. Checks against the source tables recover the reported endpoints and batch trajectories. Selected exact replays check that candidate selection is reproduced, while coverage checks verify the presence of all required run records. Reproduction uses the recorded configuration and initialization set for each comparison.

B Additional Experimental Results

B.1 Full main and sensitivity results

Table A2 reports the Query-AUC values for the same Main-35 runs as Table 1. Figure A1 shows the distribution of all 35 outcomes per method and task. Each cumulative curve gives the fraction of runs whose result is at or below the value on the horizontal axis, allowing comparison of the full distribution in addition to the mean.

Table A2: **Main-35 discovery speed.** Query-AUC means ± sample standard deviations over 35 matched initializations per task. Higher Query-AUC and lower mean rank indicate better performance.

Method	GB1	PABP	TrpB	Mean rank
BATA	0.6638 ± 0.0727	1.9530 ± 0.2650	$\mathbf{0.7783 \pm 0.0553}$	**2.00**
ALDE	$\mathbf{0.7111 \pm 0.0736}$	1.6602 ± 0.1351	0.7689 ± 0.0888	2.67
EVOLVEpro-650M	0.5877 ± 0.0945	$\mathbf{2.0122 \pm 0.3526}$	0.7579 ± 0.0808	3.00
REAP100-650M	0.6723 ± 0.0847	1.7809 ± 0.1391	0.7426 ± 0.1056	3.67
RF	0.6600 ± 0.0920	1.8190 ± 0.1699	0.7449 ± 0.1035	3.67

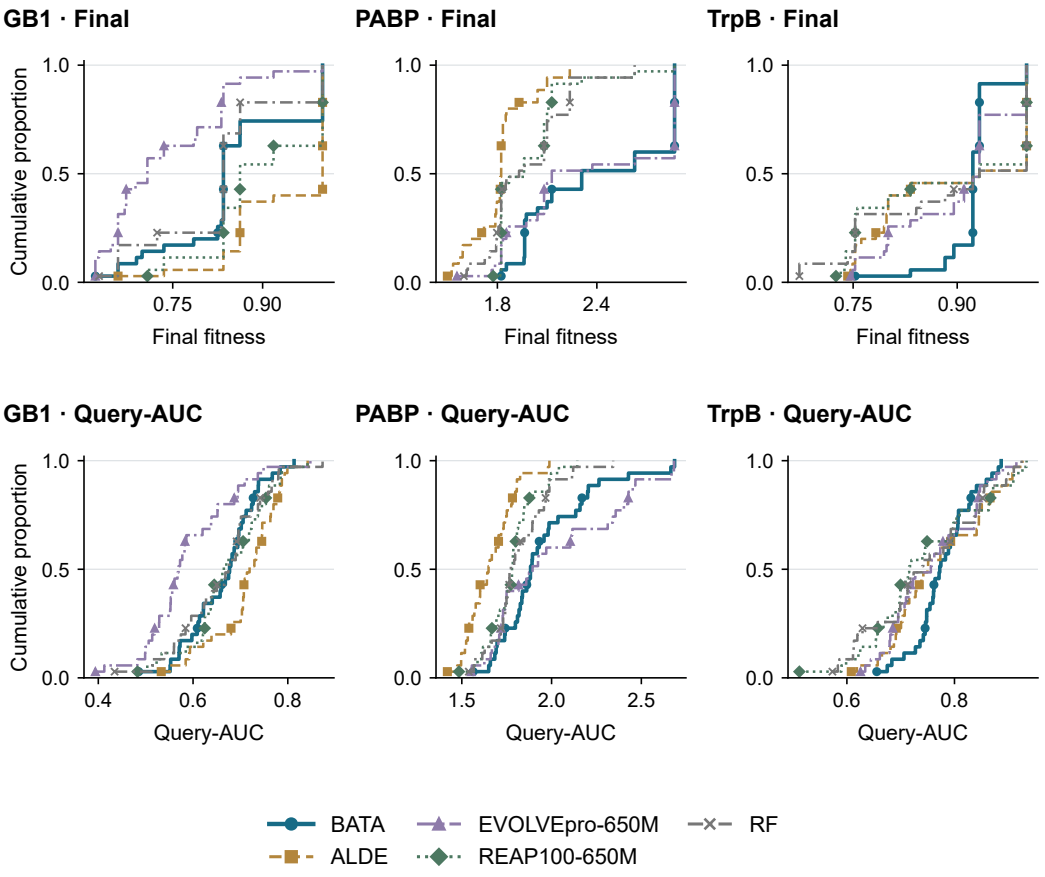


Figure A1: **Distribution of optimization outcomes.** Cumulative fractions of the 35 Main-35 runs for Final@480 and Query-AUC. Horizontal axes retain each task's fitness scale.

The separate sensitivity evaluation uses 70 initializations for GB1 and TrpB and the original 35 for PABP (Table A3). BATA and REAP100 tie for the best mean Final rank, and BATA has the best mean Query-AUC rank. These results describe each initialization set separately; the change from Main-35 is a comparison of set-level summaries.

Table A3: **Initialization sensitivity.** Means ± sample standard deviations and mean task ranks for Final and Query-AUC. GB1 and TrpB each use 70 initializations; PABP uses 35.

Method	GB1 (70)	PABP (35)	TrpB (70)	Mean rank ↓
BATA	0.8662 ± 0.1147	2.4169 ± 0.4232	0.9181 ± 0.0441	2.00
ALDE	0.9423 ± 0.1015	1.8183 ± 0.1814	0.9091 ± 0.1032	3.00
EVOLVEpro-650M	0.7909 ± 0.1106	2.3676 ± 0.4745	0.9082 ± 0.0784	3.67
REAP100-650M	0.9343 ± 0.0769	1.9960 ± 0.2371	0.9232 ± 0.1045	2.00
RF	0.8238 ± 0.1214	1.9849 ± 0.2533	0.8916 ± 0.1387	4.33

Final fitness = mean ± SD; cohorts retain their own identities.

Method	GB1 (70)	PABP (35)	TrpB (70)	Mean rank ↓
BATA	0.6816 ± 0.0878	1.9530 ± 0.2650	0.7911 ± 0.0518	2.33
ALDE	0.7525 ± 0.0975	1.6602 ± 0.1351	0.7734 ± 0.0906	2.67
EVOLVEpro-650M	0.6374 ± 0.1050	2.0122 ± 0.3526	0.7656 ± 0.0831	3.33
REAP100-650M	0.7188 ± 0.0932	1.7809 ± 0.1391	0.7687 ± 0.0960	3.00
RF	0.6850 ± 0.1190	1.8190 ± 0.1699	0.7469 ± 0.1128	3.67

Query-AUC = mean ± SD; cohorts retain their own identities.

B.2 Calibration outcomes and weight variation

The four calibration losses use identical predictor configurations and selection rules. Table A4 reports task means, and Figure A2 shows the outcomes of all 24 matched groups per task. T-DCG-96 gives the highest mean Final on PABP and TrpB, while T-DCG-192 has a slightly higher mean on GB1. Figure 5a reports all nine paired differences with their confidence intervals.

Table A4: **Comparison of calibration losses.** Mean Final@480 over 24 matched groups per task. Final and Query-AUC ranks are computed from task means and then averaged; lower ranks are better.

Objective	GB1	PABP	TrpB	Final rank	AUC rank
G-Rank	0.8297	2.3605	0.9045	3.33	3.67
T-Uniform-96	0.8445	2.2353	0.9191	3.00	3.00
T-DCG-96	0.8496	**2.4072**	**0.9301**	**1.33**	**1.33**
T-DCG-192	**0.8503**	2.3401	0.9186	2.33	2.00

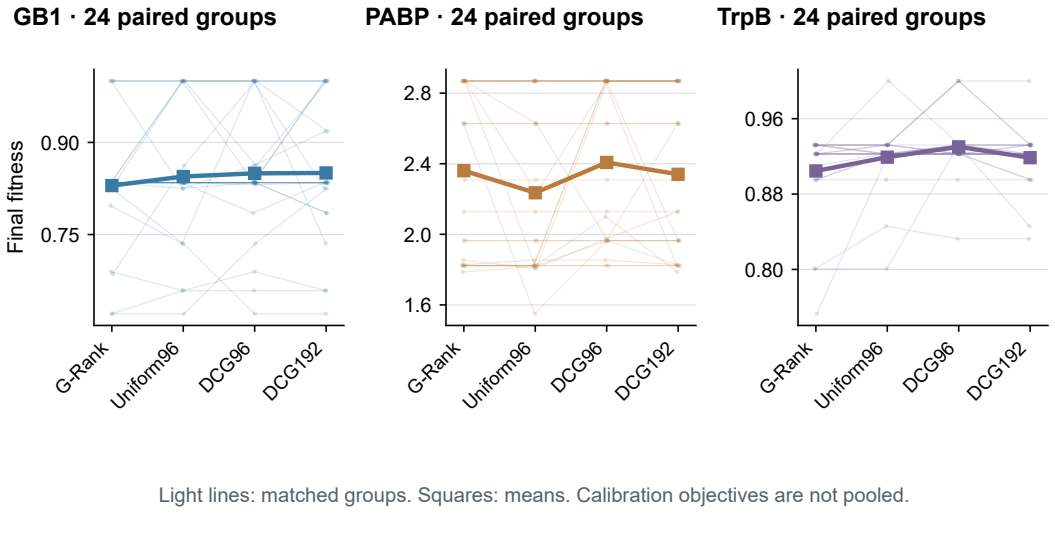


Figure A2: **Calibration outcomes by initialization.** Light lines connect the four outcomes within each matched group, and squares mark their means across groups. Each task contains 24 matched groups.

Figure A3 displays all 420 mixture weights from three tasks, 35 runs per task, and four decisions per run. Rows follow the original run order. Weights generally decrease on GB1 and TrpB, giving the task-specific predictor more influence, while PABP shows more varied changes across runs.

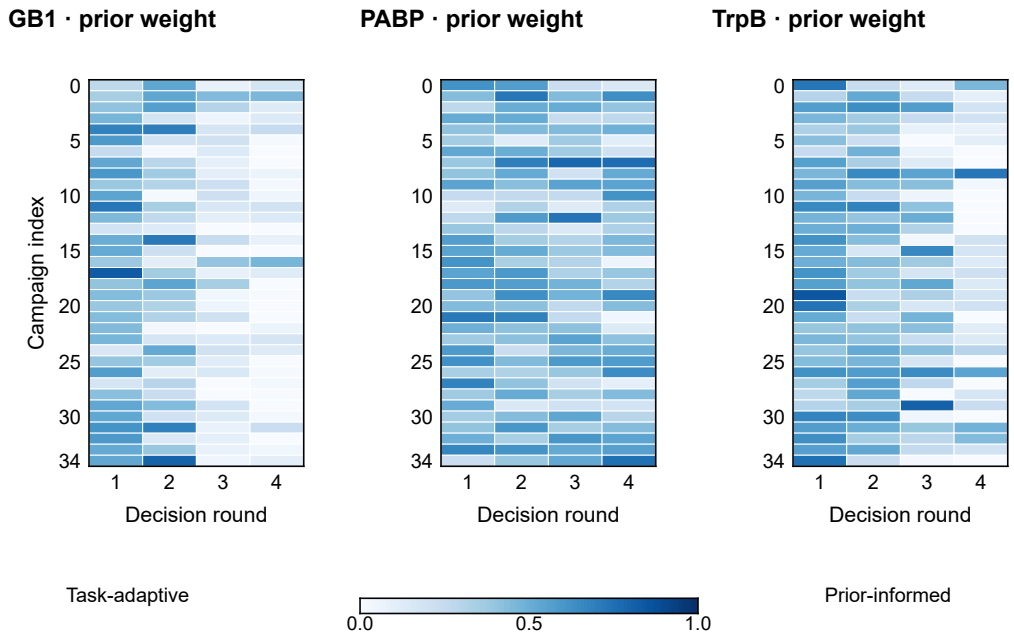
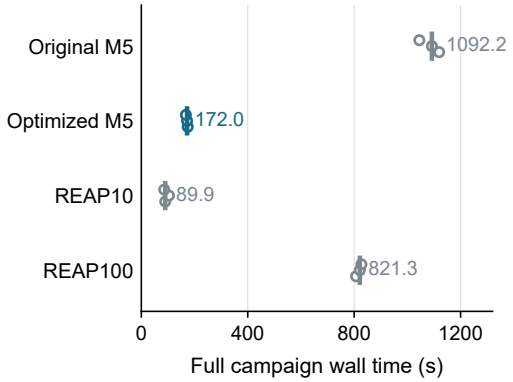


Figure A3: **Mixture weights across rounds.** Each task has 35 runs and four decisions per run. Zero assigns all weight to the task-specific predictor; one assigns all weight to the prior-informed predictor.

B.3 RUNTIME, CAPACITY, AND TASK-LEVEL LIMITS

The optimized implementation of the five-member BATA configuration (M5) reduced median runtime from 1092.19 to 171.96 seconds with cached inputs, while preserving the recorded candidate trajectories and metrics. Its median runtime was 1.91 times that of REAP10 under the same timing protocol. Figure A4a shows all three completed timings for each configuration. Panel (b) reports three attempts with the larger 100-member configuration (M100). Each stopped after two rounds at its time limit, so these elapsed times cover only part of an optimization run.

(a) Equivalent M5 acceleration



(b) M100 capacity gate

Run	Limit (s)	Stopped (s)
1	1614	1642
2	1643	1670
3	1655	1682

3/3 stopped after two rounds.

No full runtime or Final claim.

Engineering cohort: n = 3 completed timings/variant. M100 attempts are incomplete.

Figure A4: **Runtime and ensemble size.** (a) Wall times and medians for three complete runs per configuration. (b) Elapsed times for three M100 attempts stopped after two rounds. The second panel reports partial-run times.

The matched 24-initialization comparison evaluates BATA with ESM2-650M against the recorded EVOLVEpro-15B adaptation. BATA reaches mean Final values of 2.4072 on PABP and 0.9301 on TrpB, compared with 2.1177 and 0.8257 for that adaptation. A separate development comparison increases the BATA ensemble to 20 members (M20); this raises runtime and remains below REAP100 on the complete TrpB development set. Because the prediction pipelines also differ, these results compare complete configurations.

Additional comparisons identify settings where other configurations perform better. Global-MSE, a global mean-squared-error calibration configuration, exceeds BATA on PABP in a separate 24-initialization study. FolDE performs better under the separate 48-query PABP protocol, and RF and REAP100 perform better on HIS7 and GRB2, respectively. Global-MSE changes raw-score combination, covariance estimation, and cross-validation fitting, so it tests more than the calibration loss alone. Each comparison retains its own protocol and initialization set.

C Predictor Comparisons and Switching Tests

C.1 Larger task-specific ensemble

Fine-only M20 uses a 20-member task-specific ensemble without the prior-informed predictor. Figure 6 combines summaries of 188 matched pairs: 35 each for GB1 and PABP, 24 each for HIS7 and GRB2, and 70 from the separate TrpB sensitivity set. On TrpB, Fine-only M20 and BATA reach means of 0.9388 and 0.9181, respectively. Their paired difference is 0.0207, with a 95% confidence interval of [0.0045, 0.0364]. This TrpB comparison uses its own matched initializations, separate from Main-35.

Fine-only M20 changes both predictor capacity and the use of prior information. It also applies Thompson-style selection to every task, including PABP and GRB2, where BATA uses greedy selection. Therefore, the following controls retain BATA's task-specific predictor capacity and selection rule to examine the predictor combination more directly.

C.2 Controls with matched task-predictor capacity

The same-capacity comparison contains 216 runs: BATA, the task-specific predictor alone, and equal-weight rank combination, each evaluated on 24 matched initializations per main task. The task-specific predictor is a five-member ridge-regression ensemble on GB1, five bootstrap random forests on TrpB, and one 100-tree random forest on PABP. The PABP task-specific-only results reuse the recorded EVOLVEpro/RF100 control. Figure A5 reports BATA minus each control; Figure 6 reports Fine-only M20 minus BATA.

On TrpB, BATA exceeds the task-specific predictor alone by a mean of 0.0345306. The prespecified 20,000-resample interval is $[-0.000608, 0.071601]$, and a 100,000-resample check also includes zero. An earlier 10,000-resample calculation gave a lower bound of 0.000262; the prespecified interval is used for interpretation. Against equal-weight rank combination, BATA has a mean gain of 0.020349 with interval [0.011045, 0.030704]. Thus, learned weighting improves over equal weighting in this comparison, while the gain over task-specific prediction alone remains uncertain. The separate TrpB-70 comparison evaluates the larger M20 configuration.

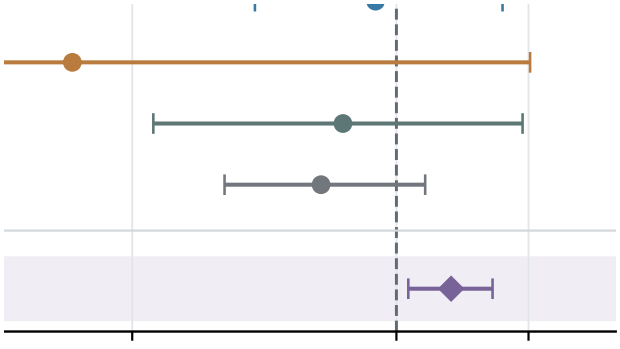


Figure A5: **Controls with fixed task-specific predictor capacity.** Mean paired Final differences and 95% intervals for BATA minus the task-specific-only and equal-weight rank controls. Each main task uses 24 matched initializations; positive differences favor BATA.

C.3 Threshold development and independent confirmation

G20 is a round-wise switching rule: use Fine-only M20 when BATA assigns the prior-informed predictor a weight below 0.20, and otherwise use BATA. Development compared thresholds of 0.20, 0.30, and 0.40 on five seeds per task across five tasks, totaling 100 runs. The selection criterion was macro normalized regret, the recorded normalized regret score averaged across tasks, with lower values preferred. G20 scored 0.1040, compared with 0.5040 and 0.5495 for the higher thresholds. Its threshold was fixed before the separate confirmation initializations were generated.

G30 and G40 use thresholds of 0.30 and 0.40, respectively. In development, G20 chose Fine-only M20 in 19 of 100 decisions, compared with 32 for G30 and 45 for G40. These more frequent switches did not improve the development criterion. Confirmation compared BATA, Fine-only M20, and G20 on ten new initializations per task, giving 150 complete runs of 480 measurements. The recorded initialization identities and algorithm seeds do not overlap with the historical experiment records.

Table A5: **Independent test of the switching rule.** Mean Final@480 over ten matched initializations per task. Bold marks the best task mean. Ranks use the recorded tie rule: decreasing mean fitness, then increasing configuration identifier.

Task	BATA	G20	Fine-only M20
GB1	**0.8802**	0.8519	0.8549
PABP	**2.3132**	2.2571	2.0830
TrpB	0.9399	0.9554	**0.9665**
HIS7	**1.4138**	**1.4138**	1.3227
GRB2	**0.6276**	**0.6276**	0.5964
Mean rank	**1.40**	2.20	2.40

In confirmation, G20 selected Fine-only M20 in 30% of decision rounds on GB1 and TrpB, 5% on PABP, 10% on GRB2, and none on HIS7. Relative to BATA, its paired Final differences were -0.0284 on GB1 (95% CI $[-0.0731, 0]$), -0.0560 on PABP ($[-0.1681, 0]$), $+0.0155$ on TrpB ($[0, 0.0359]$), and zero on HIS7 and GRB2. Figure A6 shows method ranks and switching frequencies. The changes favor different configurations across tasks, indicating that a low current prior-informed weight alone is not a reliable predictor of gains from the larger task-specific ensemble.

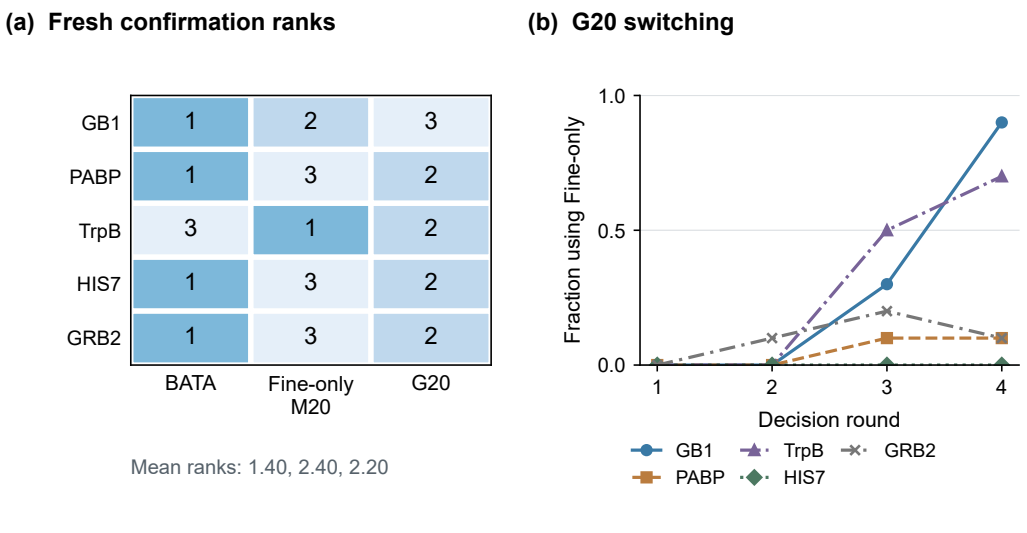


Figure A6: **G20 switching results.** (a) Task-level Final ranks and mean ranks over the independent ten-initialization sets. (b) Fraction of runs selecting Fine-only M20 at each round under the fixed threshold of 0.20. Rounds within a run share the same accumulating measurements.

D PROOFS FOR THE CALIBRATION RULE

We prove the two properties for one calibration round. Fitted predictions, percentile ranks, and sample weights are fixed throughout the argument, and the optimized quantity is the loss in Eq. 5.

D.1 PROOF OF PROPOSITION 1

Define $A = \sum_i q_i d_i^2$, $C = \sum_i q_i d_i z_i$, and $K = \sum_i q_i z_i^2$. The weighted squared loss expands to

$$\mathcal{L}_{\mathrm{BATA}}(w) = Aw^2 - 2Cw + K, \qquad A \geq 0. \tag{9}$$

When $A > 0$, completing the square gives

$$\mathcal{L}_{\mathrm{BATA}}(w) = A\left(w - \frac{C}{A}\right)^2 + K - \frac{C^2}{A}. \tag{10}$$

The squared term is minimized at C/A. Under the constraint $w \in [0, 1]$, the unique minimizing weight is the nearest point in that interval, $\Pi_{[0,1]}(C/A)$. When $A = 0$, nonnegative weights imply $d_i = 0$ for every $q_i > 0$. Therefore, $C = 0$ and the loss is constant in w, so $w = 0$ is a minimizer. Since both endpoints are feasible, the minimum calibration loss is no greater than the loss obtained by either predictor alone.

The implementation sets $w = 0$ when A is no greater than float64 machine epsilon, avoiding division by a numerically tiny value. This numerical convention covers nearly flat losses. The exact uniqueness result above applies to $A > 0$ in real arithmetic.

D.2 PROOF OF PROPOSITION 2

A strictly increasing function h preserves order and ties: $u < v$ if and only if $h(u) < h(v)$, and $u = v$ if and only if $h(u) = h(v)$. Therefore, average ranks and percentile ranks are unchanged. Applying such a transformation to either fixed out-of-fold (OOF) prediction vector leaves r^p, r^a, d, z, and the weight in Eq. 6 unchanged. The measured fitness ranks and sample weights q_i also remain fixed.

The argument also applies to each ensemble member's prediction vector over the unmeasured pool. The percentile ranks in Eq. 7 remain the same, and so does their weighted combination. Since the calibration numerator and denominator are unchanged, the implementation's numerical threshold produces the same weight as well.