RESOLVE: JOINTLY VALUING MODEL ADAPTATION AND EXPERIMENTS FOR PROTEIN ENGINEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein engineering requires choosing which variants to measure and how to adapt predictive models from limited experimental data. Nonmyopic experimental design values measurements for both immediate discovery and their ability to improve later choices. When models are fine-tuned on new observations, these choices also depend on changing representations. We introduce RESOLVE, a controller that combines acquisition lookahead with a common discovery objective for model adaptation and experimental selection. Its evaluator accounts for representation changes through hypothetical predictor and representation updates and assesses frozen and adapted recommendations through the experimental batches they propose. Our theoretical analysis connects valuation accuracy to immediate selection quality. Retrospective experiments across diverse protein assays and feedback schedules demonstrate improved discovery over acquisition and planning baselines under matched measurement budgets. Ablation studies show that accounting for adapted representations improves value assessment, while selecting between frozen and adapted recommendations improves discovery. These results support representation-aware valuation and coupled action-batch selection as useful components of nonmyopic experimental design for protein engineering.

1 INTRODUCTION

Protein engineering seeks variants with useful biological properties under limited experimental budgets (Romero et al., 2013; Yang et al., 2025). Protein language models and assay-specific adaptation offer different predictive perspectives for guiding these experiments (Lin et al., 2023; Zhou et al., 2024). Their usefulness may depend on the target property and the measurements available. Even with the same observed labels, different predictors can recommend different variants. This raises a practical question about how an experimental campaign should use competing recommendations. Can it determine which recommendation to follow as data accumulate, and does making this choice improve discovery over consistently following one predictor? The choice affects both the variants discovered immediately and the evidence available for subsequent learning. A useful selection strategy must therefore consider what a proposed experiment may discover now and how its outcomes could inform later experiments.

We study a minimal instance of this broader problem using frozen and adapted views of a single ESM-2 backbone. Each view combines a sequence representation with a predictor fitted to acquired labels and proposes a batch of unmeasured variants. These views provide competing predictive perspectives while holding the source of pretrained information fixed. This setting separates the benefit of recommendation selection from access to additional pretrained models. Adaptation can change which variants appear promising, but better fit to observed labels or greater confidence in a proposed batch need not imply greater discovery value. Active search already values observations through subsequent decisions (Garnett et al., 2012; Jiang et al., 2017). Ensemble and portfolio methods retain multiple predictive hypotheses or competing recommendations (Lu et al., 2023; Hoffman et al., 2011; Angermueller et al., 2020). We ask whether evaluating current and subsequent discovery improves selection between frozen and adapted recommendations over fixed views and simpler selection rules. Both views are prepared before selection. The controller values the use of adaptation through the recommendation it executes.

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.

We introduce RESOLVE, a controller that values model adaptation through the experiments it recommends (Figure 1). A weighted pool of predictive hypotheses supplies possible outcomes for each proposed batch. Temporary predictor and representation updates estimate how those observations could affect discovery with the remaining budget. RESOLVE compares the recommendations using shared current scores and simulated future value, executing the adapted batch only when its estimated benefit justifies departing from the frozen recommendation. Only real measurements are retained between rounds. Across 24 protein assays and six feedback schedules, RESOLVE improves discovery over fixed views, simpler selectors, and the evaluated acquisition and planning baselines under matched measurement budgets. Component comparisons show that retaining a predictive pool and sampling alternative worlds for virtual outcomes contribute to discovery. Virtual representation updates further improve campaign recall over repeated predictor-only evaluation and reduce gain-forecast error on shared reference states. Local theory characterizes the current-score cost of departing from the frozen recommendation and its sensitivity to score error.

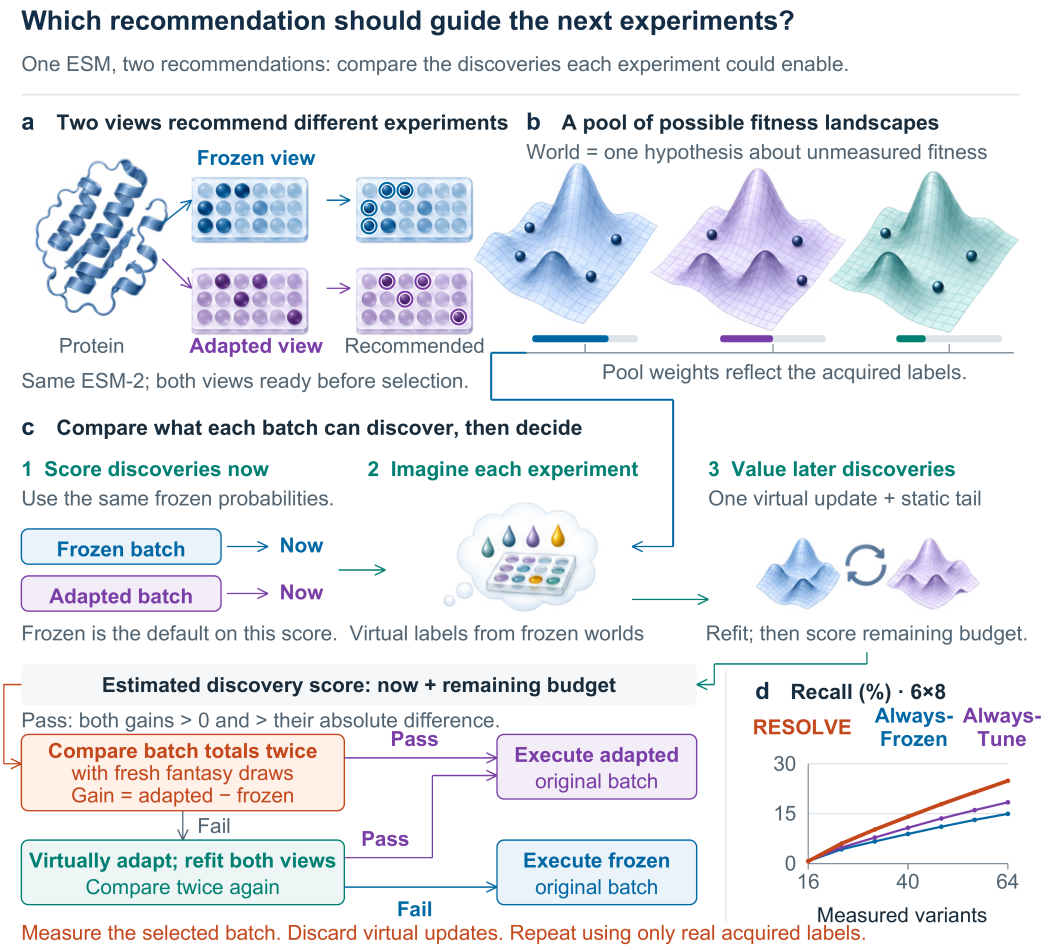


Figure 1: **RESOLVE selects experimental recommendations by valuing current and subsequent discovery.** (a) Frozen and adapted views of the same ESM-2 model recommend different experiments, motivating a choice between their discovery opportunities. (b) A weighted pool of fitness hypotheses represents uncertainty and supplies possible experimental outcomes. (c) The controller evaluates the proposed batches through their current scores and simulated learning outcomes, then executes one batch and observes its real labels. (d) Recall as measurements accumulate over six rounds of eight variants after 16 initial measurements, averaged over five seeds and 24 assays.

2 Related Work

Prediction and experimental discovery. GPs provide predictions and uncertainty estimates for guiding experiments, including in protein engineering (Romero et al., 2013). Active learning, Bayesian optimization, and active search use predictive models for different experimental objectives. These include improving prediction, gaining information, finding high-value outcomes, and discovering more target candidates (Cohn et al., 1994; Kirsch et al., 2019; Srinivas et al., 2009; Hennig & Schuler, 2012; Garnett et al., 2012). RESOLVE addresses finite-library active search with elite recall as its objective. Predictors support this decision process, but predictive accuracy alone is not the campaign objective. We evaluate a recommendation by the useful variants it helps discover within a fixed measurement budget.

Predictive perspectives in protein engineering. Protein-engineering workflows already use different representations, predictors, and adaptation strategies. ALDE evaluates encodings, GP and neural predictors, and multiple acquisition strategies. EVOLVEpro iterates prediction and experimental selection using protein-language-model features (Yang et al., 2025; Jiang et al., 2024). LoRA enables parameter-efficient adaptation, FSFP applies adaptation to fitness prediction with scarce labels, and AFP-DE couples fine-tuning with active sampling (Hu et al., 2021; Zhou et al., 2024; Qin et al., 2023). These approaches motivate studying how a campaign should use recommendations from different predictive perspectives. RESOLVE examines a controlled instance with frozen and adapted views of one backbone. It asks whether selecting between their proposed experiments improves discovery over consistently following either view.

Ensembles and recommendation portfolios. Ensemble-GP Bayesian optimization maintains a kernel dictionary and uses its mixture posterior for Thompson sampling (Lu et al., 2023). Stacking combines predictive distributions using predictive evidence (Yao et al., 2018). Selection among competing methods also has established precedents. P3BO allocates biological proposals among methods, and DyNA PPO selects predictive simulators for sequence-policy training (Angermueller et al., 2020; 2019). GP-Hedge rewards acquisition nominees using an updated common GP, while the Entropy Search Portfolio compares nominees through a common information criterion (Hoffman et al., 2011; Shahriari et al., 2014). CAKE evolves GP kernels, and BAKER combines model evidence with a kernel's own acquisition value (Suwandi et al., 2026). RESOLVE builds on these ideas while focusing on the experimental consequences of choosing a frozen or adapted recommendation. It fixes the kernel dictionary and compares proposed batches using a shared current score and simulated subsequent discovery. Its pool supports both live predictions and virtual outcomes, whose roles are examined through component comparisons.

Learning responses in nonmyopic experimental design. Active search, Efficient Nonmyopic Search (ENS), batch ENS, rollout, and multistep BO value hypothetical measurements through their effects on subsequent decisions (Garnett et al., 2012; Jiang et al., 2017; 2018; Lee et al., 2020; Jiang et al., 2020). ENS also accounts for the remaining search budget. RESOLVE follows this nonmyopic perspective through one virtual learning update and a static approximation of discovery over the remaining budget. Its two evaluation stages distinguish refitting predictors on fixed features from updating the representation before refitting. We test whether these simulated learning responses improve selection between the two experimental recommendations through campaign and forecast-error comparisons. The static tail remains an approximation, and the agreement rule provides neither a long-horizon optimality guarantee nor calibrated statistical confidence.

3 Problem Formulation

We consider protein engineering as a sequential search for high-performing variants in a finite sequence library $\mathcal{X}$. Before round t, the controller has access to the labels acquired so far, denoted by D_t, while unmeasured labels remain hidden. The retrospective evaluator defines an elite set E containing k high-performing variants. The controller knows k but not which variants belong to E. For a set of measured variants C, discovery performance is

$$\mathrm{Recall}(C) = \frac{|C \cap E|}{k}. \tag{1}$$

The objective is to maximize this recall under a fixed measurement budget. Each campaign begins with 16 common initial measurements and acquires 48 additional labels over n rounds of q measure-

ments, where $nq = 48$. At zero-based round t, the number of decisions remaining after the current batch is $R = n - t - 1$. The initial measurements count toward both the budget and recall.

Selecting among experimental recommendations. Let $\mathcal{V}$ be a prespecified set of predictive views. Each view combines a sequence representation with a fitted predictor and recommends a feasible batch $B_{v,t}$ using the available sequences and acquired labels. The controller selects one batch from $\{B_{v,t} : v \in \mathcal{V}\}$, measures its variants, and adds their labels to D_t. The resulting observations inform the next round of predictions and recommendations. A selection rule is therefore evaluated through the discoveries made over the entire campaign, including the effects of its choices on subsequent learning. Predictive hypotheses within a view describe uncertainty and need not be separately selectable views.

A controlled setting with two views. We study $\mathcal{V} = \{\mathrm{Frozen}, \mathrm{Tune}\}$. Frozen uses frozen ESM-2 features, while Tune uses the same backbone adapted to the acquired labels. Both views fit predictors to the available observations and recommend batches from their respective candidate sets. This provides a minimal instance of competing predictive perspectives while holding the source of pretrained information fixed. We test whether selecting recommendations according to their estimated current and subsequent discovery value improves recall over consistently following either view or using a simpler selection rule. Additional encodings and backbones extend this general problem but are not evaluated in this study.

4 Method: Value Adaptation through Its Proposed Experiments

RESOLVE compares batches proposed by frozen and adapted views of the same protein language model. Both receive a shared current score from the frozen pool. Because the frozen proposal maximizes this score, the adapted proposal must offer sufficient estimated subsequent benefit to offset its current-score disadvantage. The controller first evaluates virtual predictor updates and, after rejection, also evaluates a virtual representation update. A stage selects the adapted batch only when two comparisons agree on sufficient benefit. Otherwise, the frozen batch is measured. Virtual state is discarded, and only real measurements enter the next round.

4.1 Build two pools of possible fitness predictions

A *view* specifies a representation and a recommendation procedure. Within each view, a *world* is a fitted fitness hypothesis with predictive uncertainty, and a *pool* retains several worlds with data-dependent weights. Both views use mean-pooled ESM-2 650M features (Lin et al., 2023). Frozen retains the pretrained representation, while Tune trains a fresh ten-epoch LoRA from the original backbone on acquired labels each round (Hu et al., 2021).

Each view uses ten GP worlds (Williams & Rasmussen, 2006), comprising one linear kernel and nine linear-plus-RBF kernels. For standardized features with linear Gram g,

$$k_m(i,j) = g_{ij} + \lambda_m \exp(-\gamma_m d_{ij}^2), \qquad d_{ij}^2 = \max(0, g_{ii} + g_{jj} - 2g_{ij}). \tag{2}$$

Kernel specifications remain fixed, while acquired labels update GP predictions and world weights. Appendix A provides the dictionary, fitting procedure, and training settings.

Targets are rank-normal scores computed from acquired labels. For latent predictive moments $\mu_{v,m}(i), \sigma_{v,m}(i)$, the pooled threshold score is

$$p_v(i) = \sum_m w_{v,m} \Phi\left(\frac{\mu_{v,m}(i) - \tau}{\sigma_{v,m}(i)}\right). \tag{3}$$

Here Φ is the standard normal cumulative distribution function, and zero variance uses the strict threshold indicator. The common τ is derived from the frozen pooled mean and known elite count k, then fixed throughout the round and its virtual branches. These scores describe surrogate threshold exceedance rather than calibrated true elite membership.

Frozen current scores and virtual-outcome generation use normalized marginal-likelihood weights $\exp(-\mathrm{NLL})$. Adapted nomination uses in-sample Gaussian residual-density weights. After virtual refitting, both pools use marginal-likelihood weights and contribute equally to the common future score. These weighting rules use acquired or virtual training data without held-out evidence.

4.2 FIX THE TWO BATCHES AND THEIR CURRENT OPPORTUNITY COST

Let $\mathcal{U}_t$ be the eligible unmeasured library. The adapted support $\mathcal{S}_t$ contains up to 4096 candidates selected for re-embedding by a separately fitted linear GP. With deterministic index tie-breaking,

$$B_F = \mathrm{Top}_q(p_F; \mathcal{U}_t), \qquad B_T = \mathrm{Top}_q(p_T; \mathcal{U}_t \cap \mathcal{S}_t), \qquad V_0(B) = \sum_{i \in B} p_F(i). \tag{4}$$

The frozen pool anchors current value and virtual outcomes to a shared reference. Thus $\delta = V_0(B_F) - V_0(B_T) \geq 0$ prices departure from Frozen, without assuming greater accuracy. If B_T is unavailable or equals B_F, execute B_F. Otherwise, fix both batches throughout virtual evaluation.

4.3 SIMULATE LEARNING AND VALUE THE REMAINING BUDGET

For each batch, sample one frozen world and conditionally independent Gaussian targets from its latent marginals, omitting within-GP covariance. Append these values to the current rank-normal targets without reranking. The *head* stage refits both pools on fixed features. The *representation* stage first trains a fresh five-step LoRA from the original backbone and re-embeds the real window together with the virtual batch. Frozen features are substituted outside the adapted map. This stage changes both learned features and virtual-batch coverage (Algorithms 1 and 2).

For stage z and comparison j, let $p^{z,j}_{C,B}$ be the common post-refit score following virtual observations of batch B, with half its mixture mass assigned to each view. With R rounds remaining,

$$W^{z,j}(B) = \sum_{i \in \mathrm{Top}_{Rq}(p^{z,j}_{C,B}; \mathcal{U}_t \setminus B)} p^{z,j}_{C,B}(i), \qquad A^z_j = -\delta + W^{z,j}(B_T) - W^{z,j}(B_F). \tag{5}$$

The sum excludes acquired candidates and the hypothetical first batch, stopping if finite-score candidates are exhausted. It represents one virtual update followed by a static remaining-budget tail, without additional learning between future batches.

Two comparisons with fresh virtual draws pass when

$$A^z_1 > 0, \qquad A^z_2 > 0, \qquad \min(A^z_1, A^z_2) > |A^z_1 - A^z_2|. \tag{6}$$

If the head stage passes, execute B_T. Otherwise, when $R > 0$, evaluate the representation stage with fresh virtual targets and the same original proposals. Passing selects B_T; otherwise, execute B_F. The last round executes B_F. The modes head, repr, and frozen identify how the decision was reached, not three separate actions. This agreement rule is a heuristic rather than a confidence test.

4.4 WHAT THE SELECTION RULE GUARANTEES, AND WHAT IT DOES NOT

Proposition 1 (Current-score dominance and repayment)**.** *If B_F is the size-q frozen Top-q batch over finite eligible scores, it maximizes V_0 over all feasible size-q batches. Shared immediate selection therefore reduces to Always-Frozen under the same tie rule. At either virtual stage, writing $h_j = W^{z,j}(B_T) - W^{z,j}(B_F)$, acceptance is equivalent to*

$$\min(h_1, h_2) > \delta + |h_1 - h_2|. \tag{7}$$

For $R = 0$, neither stage can justify an override.

The estimated future benefit must cover both the current-score disadvantage and disagreement between simulations. Actual first-batch hit differences can still have either sign.

Proposition 2 (Immediate loss under score error)**.** *Condition on history and controller randomness H. Let $\pi_i = \Pr(i \in E \mid H)$, $e_i = |\pi_i - p_F(i)|$, and $u(B) = \sum_{i \in B} \pi_i$. Let $B^\star$ maximize expected hits over the same finite-score size-q feasible family. For the executed batch B_s, define $\delta_s = V_0(B_F) - V_0(B_s)$. Then*

$$u(B^\star) - u(B_s) \leq \delta_s + \sum_{i \in B^\star \triangle B_s} e_i \leq \delta_s + 2q\epsilon, \tag{8}$$

where the second inequality requires all relevant errors to be at most ϵ.

This bound separates current-score loss from candidate-level score error. These errors differ from the batch-level gain-forecast errors in Figure 4d. A pool alone does not guarantee small errors, and neither proposition establishes campaign optimality. Appendix B provides the proofs, static-tail identity, and limits of the agreement rule.

5 Results: Discovery Performance and Its Mechanism

We evaluate 24 ProteinGym substitution assays (Notin et al., 2023), each drawn from a distinct MMseqs2 sequence cluster (Steinegger & Söding, 2017). The benchmark includes single-mutant, mixed, and combinatorial libraries spanning different mutation depths and candidate counts (Figure 2a). Where necessary, baseline implementations were adapted to this evaluation setting. Appendix C.3 describes these modifications, the components retained from the original methods, and the resulting comparison scope. Hyperparameters were fixed using 12 development assays from separate clusters before evaluation. Each campaign begins with 16 measurements and allocates 48 additional measurements across 2, 3, 4, 6, 8, or 12 rounds, with equal batch sizes within each schedule. Methods share initialization, candidate libraries, elite definitions, and measurement budgets, while unmeasured labels remain hidden. We compare five-seed means, average paired differences across schedules within each assay, and weight assays equally. Recall gains are reported in percentage points (pp). Detailed benchmark definitions and evaluation procedures are provided in Appendix C.

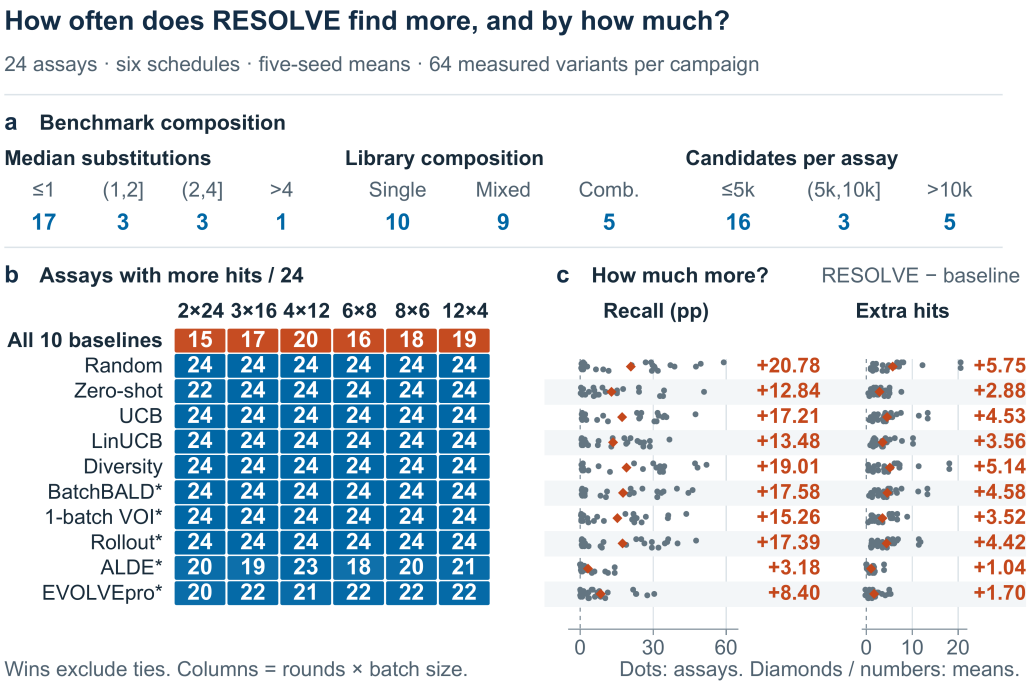


Figure 2: **RESOLVE improves discovery across assays and experimental schedules.** (a) Benchmark composition across mutation depths, library types, and candidate counts. (b) Number of assays where RESOLVE achieves strictly higher five-seed mean hits than each baseline. The orange row counts wins over all ten baselines. Columns indicate rounds and variants measured per round. (c) Paired gains in recall and additional elites. Dots represent assays averaged over six schedules, and diamonds report means across assays. Stars indicate task-adapted baselines described in Appendix C.3.

5.1 RESOLVE improves discovery over external baselines

RESOLVE achieves higher mean terminal recall than all ten external baselines. Across schedules with 2, 3, 4, 6, 8, and 12 rounds, it strictly outperforms all ten simultaneously on 15, 17, 20, 16, 18, and 19 of the 24 assays, respectively (Figure 2b). Wins compare five-seed mean hits, with ties excluded from wins. Figure 2c complements these counts with effect sizes and assay-level variation, including negative differences against some comparators.

RESOLVE reaches 23.66% terminal recall versus 2.87% for Random, improving recall by 20.78 pp and finding 5.75 additional elites per campaign. Against label-free Zero-shot, it gains 12.84 pp and

2.88 elites, showing the benefit of the complete adaptive workflow over pretrained ranking. It also outperforms UCB, LinUCB, Diversity, and BatchBALD on every assay under every schedule, with mean gains of 13.48–19.01 pp. The advantage therefore extends to the evaluated uncertainty-based, diversity-based, and information-based acquisition strategies.

Against 1-batch VOI and Rollout, RESOLVE improves recall by 15.26 and 17.39 pp. Differences in predictors and objectives make these workflow comparisons rather than isolated tests of lookahead. ALDE is the strongest external comparator on average, but RESOLVE gains 3.18 pp (95% interval $[1.54, 5.08]$) and 1.04 elites over it. Against EVOLVEpro, the gains are 8.40 pp and 1.70 elites. These results support improved discovery across the evaluated planning and protein-engineering workflows. The following matched analyses examine recommendation selection and representation-aware evaluation more directly. Detailed endpoints and assay-bootstrap intervals are provided in Table 6.

5.2 RESOLVE IMPROVES DISCOVERY OVER FIXED-VIEW AND SIMPLER SELECTION STRATEGIES

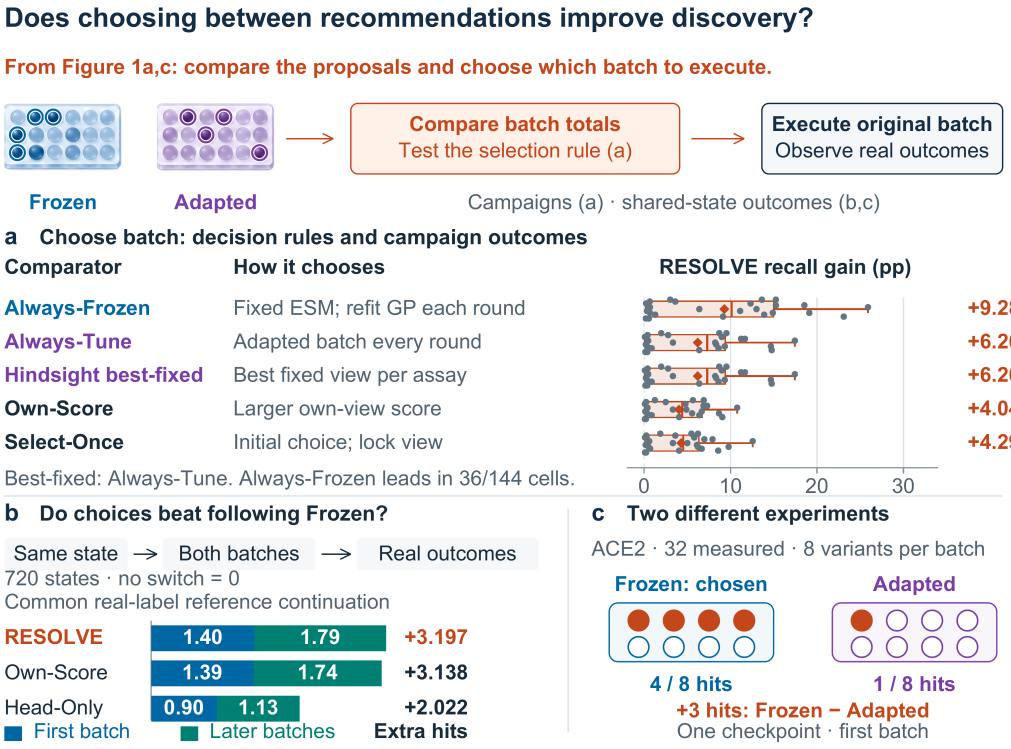


Figure 3: **Valuing recommendations improves discovery beyond fixed-view and simpler selection rules.** The upper schematic identifies the proposal-selection step under study. (a) Campaign recall gains over alternative selection strategies. Dots represent assays and diamonds report means. (b) Additional hits over following the frozen recommendation at shared states, separated into first-batch and later discoveries under a common reference continuation. (c) An ACE2 example where selecting the frozen recommendation yields three more first-batch hits. Orange circles mark elites.

We test repeated proposal valuation against fixed views and simpler selection rules, as located by the upper schematic in Figure 3. RESOLVE improves recall over Always-Frozen and Always-Tune by 9.28 and 6.20 pp (Figure 3a). Always-Frozen retains pretrained ESM-2 features but refits its GP pool on acquired labels each round, unlike label-free Zero-shot. Hindsight best-fixed selects the better fixed view per assay after averaging six schedules. It selects Always-Tune for all 24 assays, yielding the same 6.20 pp contrast. Before schedule averaging, Frozen leads in 36 of 144 assay–schedule combinations, Tune in 106, and two are tied. RESOLVE also exceeds Own-Score, which compares view-specific current scores, by 4.04 pp and Select-Once, which retains its initial

view choice, by 4.29 pp. Shared current-score selection reduces to Always-Frozen by Proposition 1. Repeated discovery valuation therefore improves on the tested fixed views and simpler selection rules.

To examine individual choices, we evaluate both original proposals at 720 shared midpoint states from independent Always-Frozen campaigns. Each branch measures its proposed batch and follows a common real-label reference continuation through the remaining budget (Appendix D.4). Figure 3b separates gains over the frozen branch into first-batch and later discoveries. RESOLVE gains 1.404 hits immediately and 1.793 later, totaling 3.197 additional hits when decisions that retain Frozen contribute zero. Its total gain exceeds Own-Score by 0.060 hits (95% assay-bootstrap interval $[-0.024, 0.146]$) and Head-Only by 1.175 hits. The local advantage over Own-Score is therefore uncertain. These paired comparisons fix the starting history and continuation.

Figure 3c illustrates the experimental consequence at an ACE2 checkpoint after 32 measurements in a campaign with six rounds of eight variants. The views propose disjoint eight-variant batches containing four and one elites. RESOLVE selects Frozen and finds three more first-batch elites. This case shows how recommendation selection changes the measured variants, while the campaign and shared-state analyses establish aggregate performance. Detailed endpoints, branch outcomes, and candidate identities are provided in Tables 6, 8, and 9.

5.3 The world pool and virtual representation update improve discovery

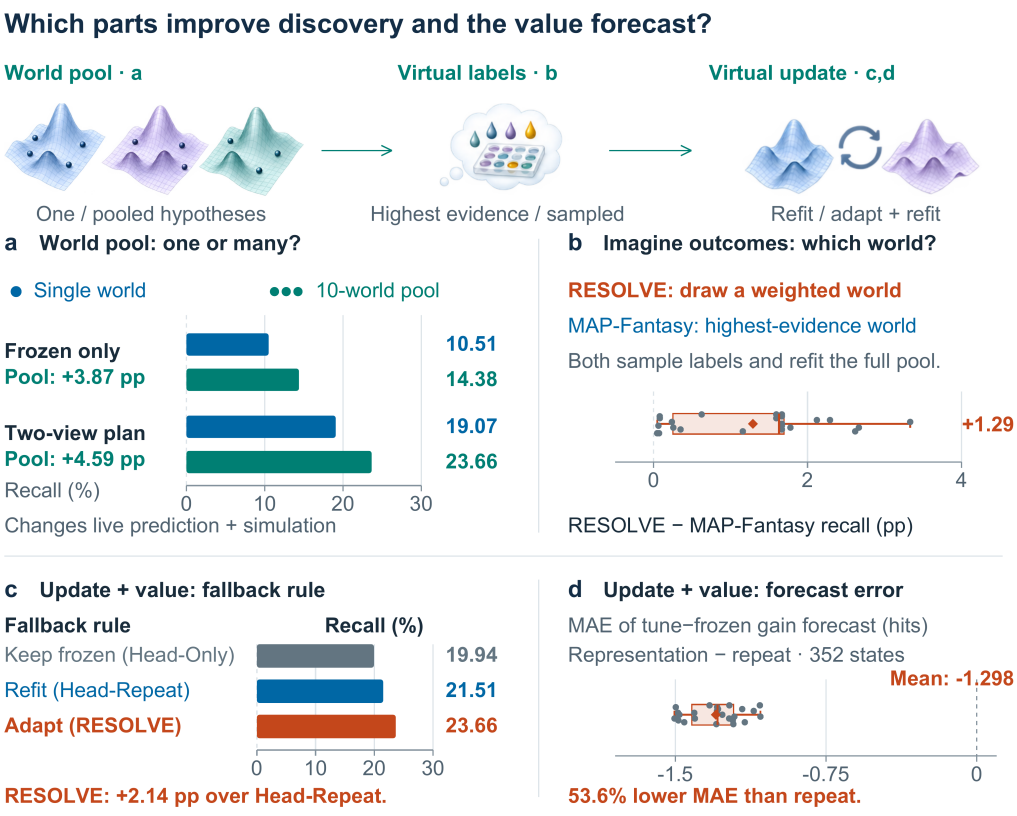


Figure 4: **The world pool and virtual representation update contribute to discovery.** The upper schematic locates the components examined. (a) Single-world and pooled predictions in frozen-only and two-view configurations. (b) Recall gained by sampling fantasy worlds rather than using only the highest-evidence world. (c) Campaign performance after stopping, repeating a predictor refit, or updating the representation following head rejection. The representation stage also expands adapted-feature coverage. (d) Paired assay-level differences in gain-forecast MAE, computed as representation minus repeat. Negative values favor representation.

We next examine the components used to value competing recommendations. The upper schematic in Figure 4 identifies the predictive pool, virtual-outcome source, and virtual learner update. In Figure 4a, retaining the ten-world pool instead of its highest-evidence member improves recall by 3.87 pp in the frozen-only configuration and 4.59 pp with two-view planning. The corresponding interaction is +0.71 pp. Pooling therefore helps both configurations, with a somewhat larger gain under two-view planning. This comparison changes live prediction and simulation together. Single-world variants still fit all ten candidates, and two-view planning also introduces the adapted recommendation, so the interaction concerns these complete configurations.

Figure 4b isolates a narrower change to virtual-outcome generation. MAP-Fantasy retains the live proposals, current scores, full-pool refits, and future mixtures, but samples virtual labels only from the highest-evidence frozen world. RESOLVE improves recall by 1.29 pp over this variant. Sampling alternative worlds therefore contributes beyond maintaining a pool for live prediction alone. The contrast changes the fantasy distribution as a whole rather than separating its mean, uncertainty, and between-world disagreement.

Figure 4c compares stopping after head rejection (Head-Only), repeating predictor-only evaluation with fresh outcomes (Head-Repeat), and updating representations (RESOLVE). Recall is 19.94%, 21.51%, and 23.66%, respectively. RESOLVE adds 2.14 pp beyond repetition with unchanged real adaptation and checkpoint proposals. Gains are 3.13 pp on 16 full-library assays and 0.17 pp on eight shortlist assays (Table 10). Lower forecast error in the latter yields little campaign improvement. Coverage cannot expand in the former, so expansion alone cannot explain the gain. Feature learning remains unisolated, and assay differences preclude causal attribution of this heterogeneity. Controls match evaluation opportunities, not computation.

Figure 4d evaluates forecasts of the tune-minus-frozen total-hit difference under the common real-label reference continuation. On 352 shared head-rejected states across 144 assay–schedule cells, MAE is 1.979 hits for the first head evaluation, 2.423 for Head-Repeat, and 1.125 for representation. Representation reduces MAE by 53.6% relative to repetition and 43.1% relative to the first head evaluation, with lower MAE in all 144 cells for both comparisons. Errors are computed per state before averaging. The plotted representation-minus-repeat difference averages -1.298 hits across assays. These gains do not imply lower forecast error at every state. Detailed results and aggregation are provided in Appendices C.2 and E.

6 Discussion

When predictive views recommend different experiments, which recommendation should receive the next measurements? RESOLVE addresses this question by valuing adaptation through its proposed batch and the discoveries its observations may enable. Studying two views of one backbone separates recommendation selection from access to additional pretrained information.

The results support evaluating recommendations by their experimental consequences. RESOLVE improves discovery over the tested fixed views and simpler selectors. Predictive pooling and sampling alternative worlds both contribute, while virtual representation updates improve campaigns beyond another predictor-only evaluation and reduce gain-forecast error on shared reference states. Campaign outcomes additionally reflect repeated choices and the measurement histories they produce.

The evidence concerns retrospective finite-library search and the specified baseline implementations. Virtual evaluation approximates future learning with marginal fantasy draws, a short representation update, and a static tail. Representation-stage gains are concentrated in full-library assays. Shortlist gains are small, and the combined intervention does not isolate feature learning. The agreement gate provides no calibrated confidence guarantee, and the local theory does not establish campaign optimality. Equal measurement budgets also need not imply equal computation.

Future work could extend selection to additional encodings, foundation models, and specialized predictors while accounting for their costs and uncertainty. Reinforcement learning could learn a selection policy from campaign trajectories, with transfer to new assays evaluated separately. An LLM agent could propose experiments or coordinate prediction and laboratory tools, while a discovery-based evaluator assesses its recommendations. These extensions pursue the same objective of using competing recommendations effectively under a limited measurement budget.

AI USE STATEMENT

We used generative AI tools to assist with drafting the implementation code of the proposed method and auxiliary scripts (e.g., data processing and plotting), and to improve the readability and grammar of the manuscript. The research ideas, method design, experimental design, and interpretation of results are entirely the authors' own; we did not use generative AI tools for ideation, hypothesis generation, synthetic data generation, or formulating or proving mathematical claims. All AI-drafted code was reviewed, modified, and tested for correctness by the authors, and all reported results were produced and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and code.

ETHICS STATEMENT

This study retrospectively analyzes existing protein-variant assay measurements. It reports no new wet-lab experiments or human-participant data collection. Improved assay discovery does not establish a variant's safety, therapeutic efficacy, or suitability for deployment. Protein-engineering methods can support beneficial research but may also have dual-use applications. Prospective use requires application-specific biosafety review and appropriate experimental oversight. Redistribution of source datasets and pretrained model weights should respect their original licenses and access conditions.

REPRODUCIBILITY STATEMENT

Implementation details, theoretical proofs, evaluation protocols, and additional results are provided in Appendices A–E. Code is available at `https://anonymous.4open.science/r/RESOLVE-62C2/README.md`.

REFERENCES

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*, 2019.

Christof Angermueller, David Belanger, Andreea Gane, Zelda Mariet, David Dohan, Kevin Murphy, Lucy Colwell, and D Sculley. Population-based black-box optimization for biological sequence design. In *International conference on machine learning*, pp. 324–334. PMLR, 2020.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

David Cohn, Zoubin Ghahramani, and Michael Jordan. Active learning with statistical models. *Advances in neural information processing systems*, 7, 1994.

Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard Mann. Bayesian optimal active search and surveying. *arXiv preprint arXiv:1206.6406*, 2012.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *nature*, 585(7825):357–362, 2020.

Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

Matthew Hoffman, Eric Brochu, Nando De Freitas, et al. Portfolio allocation for bayesian optimization. In *UAI*, volume 11, pp. 327–336, 2011.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R Sgrizzi, Lukas Villiger, Alisan Kayabolen, BJ Kim, Josephine K Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, et al. Rapid in silico directed evolution by a protein language model with evolvepro. *Science*, 387 (6732):eadr6006, 2024.

Shali Jiang, Gustavo Malkomes, Geoff Converse, Alyssa Shofner, Benjamin Moseley, and Roman Garnett. Efficient nonmyopic active search. In *International Conference on Machine Learning*, pp. 1714–1723. PMLR, 2017.

Shali Jiang, Gustavo Malkomes, Matthew Abbott, Benjamin Moseley, and Roman Garnett. Efficient nonmyopic batch active search. *Advances in Neural Information Processing Systems*, 31, 2018.

Shali Jiang, Daniel Jiang, Maximilian Balandat, Brian Karrer, Jacob Gardner, and Roman Garnett. Efficient nonmyopic bayesian optimization via one-shot multi-step trees. *Advances in Neural Information Processing Systems*, 33:18039–18049, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

Eric Lee, David Eriksson, David Bindel, Bolong Cheng, and Mike Mccourt. Efficient rollout strategies for bayesian optimization. In *Conference on Uncertainty in Artificial Intelligence*, pp. 260–269. PMLR, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Qin Lu, Konstantinos D Polyzos, Bingcong Li, and Georgios B Giannakis. Surrogate modeling for bayesian optimization beyond a single gaussian process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11283–11296, 2023.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in neural information processing systems*, 36: 64331–64379, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Ming Qin, Keyan Ding, Bin Wu, Zhenping Li, Haihong Yang, Zeyuan Wang, Hongbin Ye, Haoran Yu, Huajun Chen, and Qiang Zhang. Active finetuning protein language model: A budget-friendly method for directed evolution. In *ECAI 2023: 26th European Conference on Artificial Intelligence, September 30–October 4, 2023, Kraków, Poland–Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*, pp. 1914–1921. SAGE Publications 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2023.

Philip A Romero, Andreas Krause, and Frances H Arnold. Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201, 2013.

Bobak Shahriari, Ziyu Wang, Matthew W Hoffman, Alexandre Bouchard-Côté, and Nando de Freitas. An entropy search portfolio for bayesian optimization. *arXiv preprint arXiv:1406.4625*, 2014.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Richard Suwandi, Feng Yin, Juntao Wang, Renjie Li, Tsung-Hui Chang, and Sergios Theodoridis. Adaptive kernel design for bayesian optimization is a piece of cake with llms. *Advances in Neural Information Processing Systems*, 38:132690–132723, 2026.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

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.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Using stacking to average bayesian predictive distributions (with discussion). 2018.

Ziyi Zhou, Liang Zhang, Yuanxi Yu, Banghao Wu, Mingchen Li, Liang Hong, and Pan Tan. Enhancing efficiency of protein language models with minimal wet-lab data through few-shot learning. *Nature Communications*, 15(1):5566, 2024.

A Implementation Details of RESOLVE

A.1 State, execution order, and information access

Each campaign retains acquired raw labels, the original ESM-2 backbone, and frozen embeddings. Each round rebuilds the rank-normal targets, adapted features, GP pools, threshold, and proposals. Virtual models and labels are discarded. The controller knows k, while unmeasured labels and elite membership remain hidden.

Algorithms 1 and 2 specify real selection and virtual evaluation. Let I be the acquired indices, $\mathcal{S}$ the unmeasured shortlist, and $\mathcal{W} = I \cup \mathcal{S}$ the real adapted-embedding window. Each stage makes two randomized comparisons, evaluating both original batches in each comparison.

Input: Acquired raw labels D_t, frozen cache, original backbone, k, batch size q, remaining rounds R.
Output: One measured batch and its revealed raw labels.

1. Recompute acquired-only rank-normal targets. Fit a separate linear world and shortlist at most 4096 unmeasured indices by latent mean.
2. From the original backbone, train a fresh ten-epoch adapter on D_t. Embed $\mathcal{W}$ and discard the adapter and linear training head.
3. Fit the two ten-world pools. Set τ from the frozen pooled mean over the entire library. Construct B_F, B_T using Equation 4. Initialize $B \leftarrow B_F$.
4. If B_T exists, differs from B_F, and $R > 0$, evaluate the head stage. For $j = 1, 2$, independently draw virtual targets in each branch and compute
$$A_j^{\text{head}} = V_0(B_T) - V_0(B_F) + W^{\text{head,j}}(B_T) - W^{\text{head,j}}(B_F),$$
where each W is returned by Algorithm 2.
5. If step 4 was evaluated, apply Equation 6 to its two gains. If they pass, set $B \leftarrow B_T$. Otherwise repeat step 4 with fresh virtual targets and the representation stage. Set $B \leftarrow B_T$ only if its two gains pass. If neither stage passes, retain B_F.
6. Reveal the true labels of B together, append them to the real history, and discard all virtual state. Repeat at the next real round.

Algorithm 1: **One real round of RESOLVE.** Both recommendations exist before selection. The last round, identical proposals, and an unavailable adapted proposal all execute B_F.

Input: One original batch B, stage z, acquired targets y, live frozen pool, fixed τ, q, R, and real window $\mathcal{W}$.

1. Draw one frozen world with its live evidence weight. Draw independent marginal Gaussian targets for the members of B, conditional on that world. Append them to y without recomputing ranks.
2. If $z = \text{head}$, reuse the real-round adapted embedding map. If $z = \text{repr}$, train a fresh five-step adapter from the original backbone on acquired plus virtual targets and embed $\mathcal{W} \cup B$.
3. Supply frozen features for any index outside the available adapted map. Refit both GP pools on acquired plus virtual targets, recomputing feature and target standardization, amplitude, noise, and within-pool evidence weights.
4. Give each refitted pool half the future mixture mass. Keep τ fixed. Exclude acquired indices and B from future candidates.
5. Return $W^{z,j}(B)$, the sum of the largest Rq finite common probabilities, or all of them if fewer remain. Discard the virtual models and labels.

Algorithm 2: **Evaluate one batch's virtual future.** Algorithm 1 combines the current-value difference with the difference between the two returned future values. Representation training occurs only in step 2. The tail contains no further observations or training.

A.2 Sequence features and candidate support

The only pretrained checkpoint is `facebook/esm2_t33_650M_UR50D` (Lin et al., 2023). Last-layer mean pooling excludes padding, the first token, and the last attended token, yielding 1280 coordinates. RESOLVE divides each vector by its L2 norm, floored at 10^{-12}. Within each GP fit,

columns are centered and scaled using only its training rows. Population standard deviations at most 10^{-12} are replaced by one. If z_i denotes the resulting vector, the linear Gram and RBF distance are

$$g_{ij} = z_i^\top z_j/2304, \qquad d_{ij}^2 = \max(0, g_{ii} + g_{jj} - 2g_{ij}).$$

The denominator 2304 is a fixed kernel normalization constant, distinct from the 1280-dimensional embedding size. It scales both the linear Gram and the RBF distances and is retained across the matched variants. The dictionary contains $(\lambda, \gamma) = (0, 0)$ and $\{0.25, 1, 4\}^2$, with kernel $g + \lambda \exp(-\gamma d^2)$.

A separate fit of the dictionary's linear GP ranks all unmeasured candidates by latent mean. The best $\min(4096, |\mathcal{U}_t|)$ form $\mathcal{S}_t$, with ties resolved by smaller index. Real LoRA trains on I and re-embeds $\mathcal{W}$. Frozen nomination uses all of $\mathcal{U}_t$, while adapted nomination uses $\mathcal{S}_t$.

Sequences use the twenty standard amino acids and are processed intact without truncation. Embedding forwards use evaluation mode.

A.3 Targets, threshold, and three coordinate systems

Raw fitness is oriented so that larger is better. At each real round, sort acquired indices increasingly and assign zero-based ordinal ranks to their labels in increasing value order. Equal values receive successive ranks in index order, not their average rank. For m acquired observations,

$$y_i = \Phi^{-1}\left(\frac{r_i + 1/2}{m}\right), \qquad r_i \in \{0, \dots, m-1\}. \tag{9}$$

Ranks are recomputed after real measurements. Virtual branches retain the current targets and append virtual values on the same scale without reranking.

LoRA standardizes its entire supplied target vector only for the regression loss. GP fitting separately uses its training-target mean $\bar{y}$ and population standard deviation s_y, replacing $s_y \le 10^{-12}$ by one. GP predictions are restored as $\mu = \bar{y} + s_y \mu_{\text{std}}$ and $\sigma = s_y \sigma_{\text{std}}$. Hence current scores, virtual targets, and τ use the real round's rank-normal coordinate system, not raw fitness or the internal LoRA loss scale.

To construct τ, compute the evidence-weighted frozen latent mean over the entire library, including acquired indices. Let c be its kth largest value. Set

$$\tau = \begin{cases} (c + \max\{\bar{\mu}_i : \bar{\mu}_i < c\})/2, & \text{if a strictly smaller value exists,} \\ c - 10^{-12}, & \text{otherwise.} \end{cases}$$

The threshold stays fixed throughout the round. Tied means can place more than k candidates above it, so threshold exceedance differs from true elite membership. At zero predictive variance, exceedance is $\mathbf{1}\{\mu > \tau\}$.

Table 1: Target and value dataflow. A shared threshold aligns the event being scored, but does not guarantee calibration of different predictors.

Quantity	Data or coordinate system	Updated when
Acquired fitness	Actual measured raw labels	A real batch is revealed
Rank-normal y	Acquired-only ordinal ranks	Each real round; never inside a fantasy
LoRA loss target	Z-score of supplied y or $[y; \tilde{y}]$	Each adapter training call
GP prediction	Restored supplied-target scale	Each real or virtual GP fit
Threshold τ	Live frozen pooled latent mean	Once per real round
Virtual labels	Frozen latent Gaussian on rank-normal scale	Each branch comparison
Future mixture	Half of each refitted pool	One virtual update, then static tail

A.4 Real and virtual adaptation

LoRA (Hu et al., 2021) targets query, key, and value projections throughout ESM-2, with rank 8, scaling parameter 16, dropout 0.05, and no trainable backbone biases. Training updates only the

adapter and a fresh scalar linear regression head on the mean-pooled representation. The backbone remains fixed. Acquisition uses the resulting embeddings rather than the training head's predictions.

Every call starts from the original backbone with a fresh adapter, head, and optimizer. Real adaptation uses ten optimizer steps on acquired targets, and virtual adaptation uses five optimizer steps on acquired plus virtual targets. Both minimize equally weighted mean squared error after centering targets and dividing by their sample standard deviation (denominator $m - 1$) plus 10^{-8}.

Training uses float32 AdamW (Loshchilov & Hutter, 2017) with learning rate 10^{-4}, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay 0.01. Sequences are sorted by decreasing length without shuffling and grouped with a 3072-token budget. Gradients accumulate across chunks, weighted to implement the mean loss over all training rows for each optimizer step. The learning rate and step counts are fixed, and LoRA dropout is active during training.

Table 2: Real and virtual adaptation share a starting backbone but differ in training duration, targets, and embedding support. The head stage uses the real-round embeddings and trains no adapter.

Operation	Real adaptation	Virtual representation branch
Initialization	Original backbone; new adapter/head	Same; no warm start
Optimizer steps	10	5 per batch branch
Training indices	Acquired I	$I \cup B$
Targets	Current rank-normal y	$[y; \widetilde{y}_B]$, no reranking
Re-embedded indices	$\mathcal{W} = I \cup \mathcal{S}$	$\mathcal{W} \cup B$
Outside adapted map	Frozen feature fallback	Frozen feature fallback
Retained state	Embeddings until round ends	None after branch evaluation

A.5 GP FITTING, PREDICTIVE UNCERTAINTY, AND POOL WEIGHTS

For a fixed kernel G, standard GP regression (Williams & Rasmussen, 2006) uses a zero prior mean in standardized target coordinates and

$$C = a^2 G + (s^2 + 10^{-10})I, \qquad C\alpha = y_{\mathrm{std}}, \tag{10}$$

$$k_* = a^2 g_*, \qquad k_{**} = a^2 g_{**}, \tag{11}$$

$$\mu_{\mathrm{std}}(*) = k_*^\top \alpha, \qquad \sigma^2_{\mathrm{std}}(*) = \max(0, k_{**} - k_*^\top C^{-1} k_*). \tag{12}$$

Current scores, virtual targets, and future scores use latent predictive variance, excluding observation noise.

Each world fits amplitude a and noise s by marginal likelihood. Search a 9×9 geometric grid over $a \in [0.05, 20]$ and $s \in [0.05, 3]$, then refine three times with a 7×7 grid spanning half to twice the current optimum. Ties use the first grid pair. Final values are clipped to $a \in [10^{-3}, 100]$ and $s \in [10^{-3}, 10]$. Each world's kernel weight and inverse width remain fixed.

GP calculations use float64, SciPy Cholesky solves for the posterior, and eigendecomposition during amplitude search (Virtanen et al., 2020). Numerical predictive variances are clipped at zero.

The method uses three distinct weighting operations. For negative log marginal likelihood L_m, frozen current probabilities and fantasy-world sampling use

$$w_m^F = \frac{\exp(-L_m^F)}{\sum_\ell \exp(-L_\ell^F)}.$$

Weights are normalized after a stabilizing shift. Nonfinite members receive zero weight, with uniform weights if all are invalid. These are empirical-Bayes evidence weights with fitted hyperparameters.

Adapted nomination instead weights acquired-data residual densities. For acquired indices I, define

$$b_m = \max\left(10^{-6}, \sqrt{|I|^{-1} \sum_{i \in I} (y_i - \mu_{T,m}(i))^2}\right), \qquad \ell_m = \sum_{i \in I} \log \mathcal{N}(y_i; \mu_{T,m}(i), b_m^2),$$

and normalize $\exp(\ell_m)$ across adapted worlds. This is an in-sample score, not cross-validation. A virtual refit recomputes marginal-likelihood weights within each pool and uses

$$p_C(i) = \tfrac{1}{2} \sum_m w_m^{F\prime} p_m^{F\prime}(i) + \tfrac{1}{2} \sum_m w_m^{T\prime} p_m^{T\prime}(i).$$

Each view contributes half the future mixture. All dictionary members are refitted at each update, including those previously assigned zero weight.

A.6 Virtual observations, support, and randomization

Each branch samples one live frozen world for the batch and conditionally independent Gaussian values from its latent marginals, flooring standard deviations at 10^{-12}. The two batches receive separate draws. This retains uncertainty from the sampled world while omitting within-GP covariance (Appendix B).

Head evaluations reuse the real adapted map and substitute frozen features outside it. Representation evaluations re-embed $\mathcal{W} \cup B$ after a fresh five-step fit, with the same fallback elsewhere. The original proposals remain fixed. The representation intervention therefore changes both learned features and coverage of the virtual batch. When $\mathcal{S}_t = \mathcal{U}_t$, both stages already cover the full library, so no coverage expansion occurs. Outside the adapted map, the fallback places frozen vectors in the adapted predictor's input coordinates. This approximation does not align the two feature maps or guarantee predictive calibration.

NumPy PCG64 supplies successive fresh fantasy draws within a round (Harris et al., 2020). LoRA uses a separate training seed. Repeated virtual fits of the same batch share the initialization and dropout seed but use different virtual targets and fresh model and optimizer states.

Seeds are determined by campaign and operation identifiers, with acquired or batch indices included for adaptation calls. Initialization is shared across methods and schedules as described in Appendix C.1. Exact seed construction is specified in the implementation.

Nominations exclude acquired indices and use smaller indices to break ties. If fewer than q finite adapted scores are available, the adapted proposal is infeasible. The last round, identical proposals, and infeasible adapted proposals execute the frozen batch. The gate uses the strict inequalities in Equation 6.

A.7 Software and computational scope

The implementation uses PyTorch 2.11.0 (Paszke et al., 2019), Transformers 4.57.6 (Wolf et al., 2020), and PEFT 0.20.0 (Mangrulkar et al., 2022), with NumPy and SciPy for numerical routines (Harris et al., 2020; Virtanen et al., 2020). GP pools and the controller use custom routines.

RESOLVE constructs both real-round views before selection. Each evaluated stage requires four branch refits, and the representation stage additionally trains four five-step adapters. Fixed-frozen variants omit unused adaptation. Comparisons match measurement budgets rather than GPU time or FLOPs.

B THEORETICAL ANALYSIS AND PROOFS

B.1 CURRENT-SCORE DOMINANCE AND REPAYMENT

Proof of Proposition 1. Order the finite eligible probabilities decreasingly, with index tie-breaking. Exchanging a selected lower-ranked candidate for an unselected higher-ranked candidate cannot decrease their sum. Repeating exchanges yields B_F, proving global score maximality on this feasible family. The same comparison between B_F and B_T therefore chooses B_F, including score ties under the declared frozen tie preference.

For either stage, $A_j = h_j - \delta$. Thus $\min(A_1, A_2) = \min(h_1, h_2) - \delta$ and $|A_1 - A_2| = |h_1 - h_2|$. Substitution proves (7). Its right side is at least δ, so the inequality also implies both $A_j > 0$. The explicit positivity checks are redundant in exact arithmetic with finite values. For $R = 0$, both tails are zero, and $A_j = -\delta \leq 0$. □

The equality of the three greedy descriptions requires the same finite-score support, q, and tie rule. A Top-q control using a frozen/adapted mixture is a different score and is not covered. The result concerns predicted current mass. Actual elite counts can favor the adapted batch even when its frozen mass is smaller.

B.2 IMMEDIATE SCORE ERROR

Proof of Proposition 2. By linearity of conditional expectation, $u(B) = \mathbb{E}[|B \cap E| \mid H] = \sum_{i \in B} \pi_i$. No independence of elite memberships is required. Put $r_i = \pi_i - p_F(i)$. Shared candidates cancel, giving

$$u(B^\star) - u(B_s) = V_0(B^\star) - V_0(B_s) + \sum_{i \in B^\star \setminus B_s} r_i - \sum_{i \in B_s \setminus B^\star} r_i.$$

The first difference is at most $V_0(B_F) - V_0(B_s) = \delta_s$. The remaining terms are at most $\sum_{i \in B^\star \triangle B_s} |r_i|$. For size-$q$ batches this is bounded by $2(q - |B^\star \cap B_s|)\epsilon \leq 2q\epsilon$. □

The bound requires error control on the relevant feasible library, not just the two offered batches. It does not imply that a dictionary mixture is calibrated. For deterministic retrospective labels the same algebra holds with $\pi_i = \mathbf{1}\{i \in E\}$. No full-campaign lower bound follows without controlling subsequent errors and policy transitions.

B.3 THE REMAINING-BUDGET TAIL IS STATIC

Proposition 3 (Static-tail identity)**.** *Fix the post-refit scores and finite unmeasured support after excluding the first batch. Repeatedly taking Top-q, deleting selected points, and summing their scores for R steps equals summing the largest $\min(Rq, N)$ finite scores, where N is the support size.*

Proof. The fixed total ordering of candidates is unchanged by deletion. The first operation selects the first q entries, the second the next q, and so on, until Rq entries or exhaustion. Both procedures therefore select the same union and sum. For $R = 0$ both sums are zero. □

Consequently, conditional on these scores, this tail depends on remaining measurement capacity Rq, not on additional intermediate learning between future rounds. Schedules still affect the real histories, current batch size, fitted predictions, and repeated replanning. The identity does not make the six campaigns equivalent.

B.4 WHY TWO AGREEING SAMPLES ARE NOT A CONFIDENCE GUARANTEE

Let independent gain estimates equal 0.1 with probability 0.9 and -1 with probability 0.1. Their expectation is -0.01. Nevertheless, two draws both equal 0.1 with probability 0.81, and the gate accepts. Even unbiased independent samples therefore need not make the gate a test of positive expected gain. The subsequent representation attempt introduces an additional acceptance opportunity,

motivating Head-Repeat. A safety guarantee would require additional distributional assumptions and a calibrated testing procedure.

B.5 What the pool represents

For a mixture of scalar predictions, the law of total variance gives

$$\mathrm{Var}(Y_i) = \sum_m w_m \sigma_m^2(i) + \sum_m w_m \big(\mu_m(i) - \bar{\mu}(i)\big)^2, \qquad \bar{\mu}(i) = \sum_m w_m \mu_m(i).$$

The first term is within-world uncertainty. The second is disagreement between worlds. This standard identity explains what can be lost when selecting one GP, but does not prove that either term improves discovery. With one shared sampled world index and independent conditional marginal draws, distinct candidates have fantasy covariance $\mathrm{Cov}_m(\mu_m(i), \mu_m(j))$. The within-GP posterior covariance is omitted. A GP world and a joint GP function draw must therefore not be conflated.

C Benchmark and External Baselines

C.1 Benchmark assays and initialization

Tables 3 and 4 list the ProteinGym substitution assays (Notin et al., 2023). Hyperparameters were fixed before evaluation after checks on the 12 development assays. Each evaluation assay represents a distinct MMseqs2 cluster (Steinegger & Söding, 2017) at 30% sequence identity and 80% bidirectional coverage. The development assays belong to different clusters from the evaluation assays under that assignment. The split is defined by this sequence-clustering criterion rather than by broader protein-family annotations.

For evaluation, N counts the valid, unique nonsynonymous substitution records in the processed library. Records retain source order after excluding invalid mutation tokens and duplicate canonical variants. Wild type is not added. No smaller library is sampled. Reference sequences have length below 1,024, without truncation. Elites are the top $k = \max(10, \lceil N/100 \rceil)$ variants by `DMS_score`. Larger scores are better and no additional sign flip is applied. If the boundary fitness value is tied, a fixed assay-specific membership set selects enough boundary variants to retain exactly k elites. This set is identical across methods, seeds, and schedules. Only CCDB requires this tie resolution. Rank-normalization and the controller's surrogate τ do not define evaluation membership.

Initialization samples 16 candidates uniformly without replacement from the full library, without using labels or elite membership. Seeds 0–4 use assay-ID-based SHA-256 initialization with NumPy 1.26.4 PCG64. Indices are zero-based and sorted. The same assay–seed set is shared across methods and schedules. The table reports seed-specific integer initial hits h_0, counted after sampling. Under this sampling rule, the expected initial count for an assay is $16k/N$. Random subsequently samples from all unmeasured candidates, without a shortlist. Endpoint hits and recall include the initial 16 and 48 additional measurements. Unmeasured labels and elite membership remain hidden from the controller. k is known from N.

C.2 Campaign budgets, metrics, and aggregation

All campaigns start with 16 common measurements and acquire 48 further labels, for 64 measurements in total. The six schedules use two rounds of 24 variants, three of 16, four of 12, six of eight, eight of six, or twelve of four. Each assay and schedule is evaluated with five campaign seeds. Methods share the initial indices, candidate library, elite mask, and measurement schedule. Only acquired labels are available to each method during selection.

For assay a, schedule s, and seed r, endpoint recall is the number of measured elites divided by k_a, including the initial measurements. Paired method differences are computed at matching assay, schedule, and seed. We average seeds within each assay–schedule combination, schedules within each assay, and then assays with equal weight. Recall differences are expressed in percentage points, and hit differences count additional elites. This avoids weighting larger libraries or elite sets more heavily in the recall summary.

Figure 2b counts assays with strictly higher five-seed mean hits than the comparator under each schedule. Ties do not count as wins and remain in the 24-assay denominator. Its first row requires a strict win over all ten external baselines. Figure 2c shows paired assay-level differences after averaging the six schedules. Dots in the main figures represent assay summaries, not individual seeds. Where boxes are used, they show the median and interquartile range, with whiskers extending to the most extreme observations within 1.5 interquartile ranges. Diamonds and numerical labels show arithmetic means.

We report pointwise 95% percentile bootstrap intervals for mean paired differences. After averaging the six schedules within each assay, we resample the 24 assays with replacement 20,000 times, preserving the pairing of methods and all observations within an assay. NumPy PCG64 uses seed 20260925. These exploratory intervals treat assays as the sampling units and condition on the evaluated schedules and five-seed means. They do not estimate seed-level uncertainty or provide simultaneous coverage across all comparisons. Figures retain assay points and interquartile ranges, while Tables 6 and 8 report the intervals.

Table 3: The 24 evaluation assays. N is the processed candidate count, k is the fixed elite count, and $h_0[0{:}4]$ lists initial hits for seeds 0–4. The 120 assay–seed initializations contain 21 hits in total, averaging 0.175 hits per initialization. Rel. indicates ProteinGym substitutions release 0.1 or 1. DOI links lead to the original studies. All assays use the top-1% rule with at least ten elites and the fixed boundary mask described above.

Assay ID	N	k	$h_0[0{:}4]$	Rel.	Source
ACE2_HUMAN_Chan_2020	2,223	23	0,0,0,0,1	1	DOI
ADRB2_HUMAN_Jones_2020	7,800	78	0,0,0,0,0	0.1	DOI
BCHB_CHLTE_Tsuboyama_2023_2KRU	1,572	16	0,0,0,0,0	1	DOI
CAPSD_AAV2S_Sinai_2021	42,328	424	0,1,0,0,0	0.1	DOI
CCDB_ECOLI_Adkar_2012	1,176	12	0,0,0,0,0	0.1	DOI
D7PM05_CLYGR_Somermeyer_2022	24,515	246	0,0,0,0,0	1	DOI
DNJA1_HUMAN_Tsuboyama_2023_2LO1	2,264	23	0,0,0,0,0	1	DOI
DOCK1_MOUSE_Tsuboyama_2023_2M0Y	2,915	30	0,0,0,0,0	1	DOI
ENVZ_ECOLI_Ghose_2023	1,121	12	0,1,0,0,0	1	DOI
HEM3_HUMAN_Loggerenberg_2023	5,689	57	0,0,0,1,0	1	DOI
HXK4_HUMAN_Gersing_2023_abundance	8,396	84	0,0,1,0,0	1	DOI
ILF3_HUMAN_Tsuboyama_2023_2L33	1,329	14	1,0,1,1,0	1	DOI
MAFG_MOUSE_Tsuboyama_2023_1K1V	1,429	15	0,0,0,0,0	1	DOI
MYO3_YEAST_Tsuboyama_2023_2BTT	3,297	33	0,0,0,2,0	1	DOI
PHOT_CHLRE_Chen_2023	167,529	1676	0,0,0,2,1	1	DOI
Q6WV12_9MAXI_Somermeyer_2022	31,401	315	1,0,0,0,0	1	DOI
RCRO_LAMBD_Tsuboyama_2023_1ORC	2,278	23	0,1,0,1,0	1	DOI
RFAH_ECOLI_Tsuboyama_2023_2LCL	1,326	14	0,0,0,1,0	1	DOI
RL40A_YEAST_Roscoe_2013	1,195	12	0,1,0,0,0	0.1	DOI
SPG1_STRSG_Wu_2016	149,360	1494	0,0,0,0,0	1	DOI
TADBP_HUMAN_Bolognesi_2019	1,196	12	0,0,0,0,0	0.1	DOI
THO1_YEAST_Tsuboyama_2023_2WQG	1,279	13	0,0,0,0,0	1	DOI
TPK1_HUMAN_Weile_2017	3,181	32	0,0,0,0,1	0.1	DOI
UBR5_HUMAN_Tsuboyama_2023_1I2T	1,453	15	0,0,1,1,0	1	DOI

Table 4: The 12 development assays used for pre-evaluation checks. Counts are source rows, not evaluated campaign-library sizes. Elite-recall campaigns use the separate 24-assay evaluation set. Releases and DOI links follow Table 3.

Assay ID	Source rows	Rel.	Source
A4_HUMAN_Seuma_2022	14,811	1	DOI
AMFR_HUMAN_Tsuboyama_2023_4G3O	2,972	1	DOI
ARGR_ECOLI_Tsuboyama_2023_1AOY	1,287	1	DOI
BBC1_YEAST_Tsuboyama_2023_1TG0	2,069	1	DOI
CATR_CHLRE_Tsuboyama_2023_2AMI	1,903	1	DOI
A0A1I9GEU1_NEIME_Kennouche_2019	922	0.1	DOI
A0A247D711_LISMN_Stadelmann_2021	1,653	1	DOI
AICDA_HUMAN_Gajula_2014_3cycles	209	1	DOI
A4GRB6_PSEAI_Chen_2020	5,004	0.1	DOI
AACC1_PSEAI_Dandage_2018	1,801	0.1	DOI
B2L11_HUMAN_Dutta_2010_binding-Mcl-1	170	1	DOI
CP2C9_HUMAN_Amorosi_2021_abundance	6,370	0.1	DOI

The diagnostic analyses use the same assay and schedule weighting, but their units are paired reference states rather than independently evolving campaigns. Appendix D.4 defines their state sets, continuation policy, and denominators. Appendix E reports the detailed results.

C.3 External baseline implementations

The ten comparator labels identify the following finite-library recipes. They share the initial 16 indices, measurement schedule, eligible sequence library, and retrospective hit definition. They do not share RESOLVE's rank-normal targets, ten-world pools, or LoRA. Embedding-based baselines use the raw ESM-2 mean pool before row normalization. Thus external comparisons measure complete workflows. They do not isolate the controller from differences in predictors, target transformations, or computation. Starred names denote task-adapted recipes rather than exact reproductions of every configuration in the cited framework.

Shared linear predictor. UCB and the linear-model acquisition procedures use Bayesian linear regression on raw labels and unnormalized ESM-2 features, with prior precision one, observation-noise variance one, and jitter 10^{-8}. Its reported predictive standard deviation includes observation noise. This predictor differs from RESOLVE's separately standardized, evidence-fitted linear GP world. Conditional batch calculations use the latent covariance, adding observation noise in the conditioning denominator.

Random and Zero-shot. Random samples uniformly without replacement from unmeasured indices. Zero-shot uses the same ESM-2 masked-language head, motivated by masked-marginal variant scoring (Meier et al., 2021). All positions that differ from the reference are masked simultaneously. The score sums log probability of the variant amino acid minus that of the reference amino acid over those positions. When no reference is supplied, the implementation constructs a position-wise modal reference, breaking ties by the order `ACDEFGHIKLMNPQRSTVWY`. No measured labels update this score. Simultaneous masking and the reference choice are part of this recipe.

UCB and Diversity. UCB ranks candidates by $\mu + 2\sigma_{\mathrm{pred}}$, with a fixed coefficient two. The UCB principle is standard (Srinivas et al., 2009), but the predictor here is the preceding linear model. Diversity uses greedy k-center with squared Euclidean distances in raw embeddings, taking already acquired sequences as the initial centers. Both break score ties by smaller index.

LinUCB acquisition portfolio. The contextual-bandit rule (Li et al., 2010) selects one of five acquisition experts, rather than directly applying a linear bandit to sequence embeddings. The experts return Top-q by posterior mean, $\mu + 0.5\sigma_{\mathrm{pred}}$, expected improvement over the largest acquired raw label, predictive standard deviation, or greedy k-center. With zero-based round t, n scheduled rounds, acquired count m, and unmeasured count u, the ten context entries are 1, t/n, remaining labels divided by nq, $m/(nq)$, $\min(1, \log_{10}(u)/6)$, $\tanh(y_{\max} - y_{\mathrm{med}})$, $\tanh(y_{\mathrm{med}} - y_{\min})$, mean pairwise Jaccard overlap among the five proposed batches, $\tanh(\mathrm{sd}(y))$, and $\min(1, q/u)$. These y values are raw labels.

For each expert e, initialize $A_e = I_{10}$ and $b_e = 0$, and score

$$\theta_e^\top x + \sqrt{x^\top A_e^{-1} x}, \qquad \theta_e = A_e^{-1} b_e.$$

Expert ties use their listed order. After observing the selected batch, let r be its fraction of labels strictly above the pre-batch largest raw label. Update only that expert by $A_e \leftarrow A_e + xx^\top$ and $b_e \leftarrow b_e + rx$. Its reward is improvement over an incumbent, not elite recall.

Gaussian batch information (BatchBALD*). The comparator is a Gaussian regression adaptation of batch information acquisition, rather than the categorical Monte Carlo estimator of Kirsch et al. (2019). Shortlist the $\min(u, \max(512, q))$ unmeasured candidates with largest predictive standard deviations. Greedily choose the largest conditional latent variance and condition the joint Gaussian on that point at its current posterior mean. For selected index i, observation-noise variance $\sigma^2_{\mathrm{obs}} = 1$, and latent covariance K,

$$d = K_{ii} + \sigma^2_{\mathrm{obs}}, \quad \mu \leftarrow \mu + K_{\cdot i}(y - \mu_i)/d, \quad K \leftarrow K - K_{\cdot i}K_{i\cdot}/d.$$

Here $y = \mu_i$, so the mean remains unchanged. No posterior samples or explicit log-determinant computation are used. These steps define the evaluated Gaussian implementation, which is not equivalent to the original classification-based BatchBALD procedure.

One-batch VOI*. Shortlist $\min(u, \max(256, 2q))$ candidates by posterior mean. The utility is the sum of the largest q posterior means over that conditioned shortlist, not elite probability. A candidate's value is the expected utility increase under five-node Gauss–Hermite observations at that point. After selecting a point, condition on its posterior mean and repeat to fill the batch. Previously selected points are excluded from further nomination, but remain eligible in the utility's Top-q sum. Only within-batch lookahead is used. This distinction matters when interpreting the result as a value-of-information recipe rather than a matched discovery-objective planner.

Rollout*. The rollout principle values a choice by simulated continuation (Lee et al., 2020). This implementation combines expected improvement over the largest acquired raw label with expected subsequent improvement under a greedy expected-improvement continuation. With b labels remaining before the round, set $h = \max(0, \lceil b/q \rceil - 1)$ and shortlist

$$\min\{u, \max(96, q(1 + h) + q)\}$$

candidates. A slot-level simulation fills the remaining slots of the current batch and then h future batches of q slots. The outer expectation has three Gauss–Hermite nodes. Candidate comparisons at the same slot share simulation random numbers. Already chosen current-batch points cannot be nominated again. This planner uses its linear predictor and improvement objective, not RESOLVE's representation update or elite-count surrogate.

EVOLVEpro*. The adapted recipe retains the protein-language-model-feature, random-forest, and iterative ranking structure of Jiang et al. (2024). It fits a 100-tree random forest (Breiman, 2001) to raw ESM-2 features and raw labels, uses `max_features="sqrt"` and one worker, and selects the largest predicted means. It does not use forest disagreement for acquisition. This comparison applies to the specified backbone and forest settings rather than every possible EVOLVEpro configuration.

ALDE*. ALDE covers multiple representations, predictors, and acquisition choices (Yang et al., 2025). This recipe uses five MLPs on raw ESM-2 mean-pool inputs. Each has two hidden layers of width 256 and trains on a 90% bootstrap sample of acquired rows, for at most 100 epochs with Adam (Kingma & Ba, 2014) at learning rate 10^{-3}. Training stops after ten epochs without a loss decrease of at least 10^{-6}. There is no validation set. Standardization uses all acquired rows rather than each bootstrap separately. At each batch position, sample an ensemble member uniformly and select its highest-scoring remaining candidate. No additional Gaussian perturbation is added. The predictor is an MLP ensemble. Its input still comes from ESM-2.

Comparison scope and computation. Random needs no predictor. Linear acquisition, forests, and the ALDE predictor operate on the CPU. ESM embedding, zero-shot masked scoring, and RESOLVE adaptation use the GPU. The comparisons match measurements and feedback schedules, not predictor families or GPU time. Restricting all methods to the enumerated library permits retrospective outcomes under the same hidden-label interface, but does not establish equivalence to open-ended generation or prospective laboratory campaigns. The matched variants in Appendix D address recommendation selection and virtual learning more directly.

Table 5: Actual external decision rules and the scope of their comparison. Each recipe uses the same measurement budget. Internal objectives and predictors differ.

Recipe	Implemented decision	Main difference from RESOLVE
UCB	Linear mean plus $2\sigma_{\mathrm{pred}}$	Raw labels; one linear predictor
LinUCB	Contextual selection among five experts	Incumbent-improvement reward
BatchBALD*	Greedy conditional Gaussian variance	Information criterion; fixed features
1-batch VOI*	Expected Top-q posterior-mean increase	Within-batch utility, including chosen points
Rollout*	Improvement plus greedy continuation	Fixed-feature predictor and improvement reward
ALDE*	Bootstrap MLP member sampling	Ensemble recommendation without view gate
EVOLVEpro*	Random-forest mean ranking	Fixed PLM features and greedy selection

D VARIANTS AND DIAGNOSTIC EXPERIMENTS

The internal variants share initialization, measurement schedules, raw-label access, kernel dictionary, and deterministic nomination ties unless a change is specified below. Each follows its own acquired history. Fixed-frozen variants omit unused computations, including adaptation when the recommendation rule does not use adapted features.

D.1 RECOMMENDATION-SELECTION VARIANTS

Figure 3a compares the complete controller with fixed-view and simpler selection rules. The following variants define which recommendation is executed as each campaign acquires its own measurements.

1. **RESOLVE** uses the two-stage rule in Algorithm 1.
2. **Always-Frozen** refits the frozen ten-world pool on current rank-normal targets, derives its threshold, and executes its Top-q. It does not construct an adapted pool, shortlist, or real LoRA. Its nomination is also the shared-immediate rule in Proposition 1.
3. **Always-Tune** constructs the shortlist, real ten-epoch adapter, and both pools as in RESOLVE, but removes the gate. It executes B_T whenever available and B_F otherwise.
4. **Own-Score** constructs the same two proposals but compares $\sum_{i\in B_T} p_T(i)$ with $\sum_{i\in B_F} p_F(i)$, without a future term. It chooses Tune only for a strict advantage and otherwise Frozen, refitting from new real labels at every round. This intervention changes both cross-view scoring and lookahead. Its contrast does not isolate either alone.
5. **Select-Once** runs RESOLVE's complete initial decision at 16 acquired labels, then commits to the chosen view. A frozen commitment refits only the frozen pool thereafter. An adapted commitment repeats the ten-epoch real adaptation and adapted nomination each round, with Frozen fallback when Tune is unavailable but without changing the commitment. It tests repeated reassessment against this particular initial decision, not against every possible fixed-policy selector.

For hindsight best-fixed, we first average the six schedule results within each assay separately for Always-Frozen and Always-Tune, and then choose the larger mean. This is a retrospective comparator selected using completed campaign outcomes. It is not a policy with access to those outcomes during acquisition. Choosing once per assay after schedule averaging differs from selecting the better view separately for each assay–schedule combination.

D.2 PREDICTIVE POOLS AND VIRTUAL-OUTCOME VARIANTS

Figures 4a,b distinguish the complete single-world configuration from a narrower change to the source of virtual outcomes.

1. **Frozen-Single** fits all ten frozen worlds at each update, then retains the finite highest-marginal-likelihood member, with dictionary order breaking ties. It uses that member for its threshold and Top-q, without LoRA or a gate. It is not a fixed linear kernel or a one-fit computational control.
2. **Full-Single** applies the same evidence-based member selection separately in each view at every real and virtual fit. The retained members define current proposals, threshold, fantasy source, and future predictions. The two-stage gate and virtual five-step adaptation remain. Each future view still has mass one-half, and within-world Gaussian fantasy draws remain. This changes the entire prediction-and-planning configuration, including adapted nomination, rather than only removing fantasy diversity.
3. **MAP-Fantasy** retains the live pools, proposals, refitted pools, future mixture, and both evaluation stages. Only the fantasy world index is fixed to the live frozen member with highest marginal likelihood. Gaussian target sampling within that world remains. This separates the use of multiple worlds in imagined outcomes from maintaining them for live predictions.

For the pool-by-planning contrast, let $Y_{P,L}$ denote campaign recall with pool setting $P \in \{\text{single}, \text{pool}\}$ and planning $L \in \{0, 1\}$. We report all four configurations and

$$I = (Y_{\text{pool},1} - Y_{\text{pool},0}) - (Y_{\text{single},1} - Y_{\text{single},0}).$$

Differences are computed within each assay and schedule. The planning configurations also introduce the adapted recommendation, so this interaction is between complete configurations rather than a pure intervention on tail length. Full-Single changes live predictions and simulated updates. MAP-Fantasy is the narrower fantasy-source test. A positive interaction means the planning increment is larger with the pool under these specified interventions, not that the effects are independent or additive.

These interventions retain the stated future-mixture weights, virtual training duration, number of comparisons, and candidate-support rule. Full-Single changes live prediction and simulation together, whereas MAP-Fantasy changes the sampled world while retaining within-world Gaussian draws. The resulting contrasts concern these specified configurations and do not isolate each retained setting or each source of uncertainty separately.

D.3 Virtual-update variants

Figure 4c compares the representation fallback with stopping after the initial head evaluation or repeating a predictor-only evaluation.

1. **Head-Only** executes B_T if the first two-gain head gate passes. Otherwise it immediately executes B_F. No second stage is run.
2. **Head-Repeat** uses the same first-stage stream as RESOLVE. After rejection, it draws fresh fantasies from the continuing generator and repeats both comparisons with fixed real-round features. Pools are fitted afresh. Original embeddings, proposals, and τ are unchanged. There is no virtual LoRA, nor reuse of a preceding branch's fitted posterior.

Real-round adaptation follows the same procedure in RESOLVE, Head-Only, and Head-Repeat. At a fixed checkpoint, both fallback stages evaluate the same original proposals. The representation stage changes the learned features and expands adapted-feature coverage to include its virtual batch, so the contrast measures this combined update. Head-Repeat matches an additional evaluation opportunity, while the representation stage also performs virtual LoRA training. Measurement budgets are matched, but these interventions need not have equal GPU time or FLOPs.

D.4 Shared-state branch evaluation and forecast error

Checkpoints and fixed recommendations. The reference bank uses the prespecified midpoint decision $t = \lfloor (n-1)/2 \rfloor$ from each assay–schedule–seed Always-Frozen campaign. This reference policy is shared and independent of which selector is being diagnosed, though it still defines a particular state distribution. Each checkpoint preserves its acquired history and original frozen/adapted proposals. The primary analysis includes agreements, unavailable alternatives, and rejected overrides. All diagnostic selectors are evaluated on this same Always-Frozen reference bank.

Common continuation. For each distinct original first batch, a cloned state reveals only that batch's actual assay labels and continues to 64 total measurements. At every remaining round, a fresh ten-epoch LoRA starts from the original backbone and both kernel pools are refitted on acquired real labels. For this diagnostic continuation only, Top-q selection uses the half-frozen, half-adapted marginal-likelihood mixture, with that real round's freshly computed threshold and specified frozen-feature fallback. We call this `COMMON-POOL-GREEDY`. Training seeds depend on campaign seed and remaining-step index, giving the two branches the same initialization protocol. No fantasy, gate, or virtual adapter is used. This continuation evaluates the choices under a specified reference policy rather than the future return of the deployed RESOLVE policy. RESOLVE's independent campaign endpoints remain the primary deployed-policy measure.

Repayment and prediction error. For selector s, the branch starting with its chosen batch is compared with the frozen branch. Actual newly discovered elites are counted separately in the first batch

and all remaining batches. Define $\Delta H_{\mathrm{all}} = \Delta H_{\mathrm{first}} + \Delta H_{\mathrm{rest}}$. The primary mean includes every state. Retaining the frozen batch gives a difference of exactly zero. Conditional override summaries have a different denominator from the all-state mean. The raw tune-minus-frozen contrast defines the paired reference for forecast comparison. A predicted current-score sacrifice is not an observed hit loss. Each stage's forecast is the mean of its two simulated gains, not its minimum or agreement margin. Absolute prediction errors are computed per state before averaging. Head and representation forecasts are compared on the same head-rejected subset. The static tail and real adaptive continuation differ, so forecast error includes this approximation.

The shared-state analysis includes 720 midpoint states. Forecast comparisons use the same 352 head-rejected states across 144 assay–schedule combinations. Absolute errors are computed per state before averaging within each subset, then across schedules within assay and finally across assays. Assays receive equal weight despite differing subset sizes. An empty override subset has no conditional mean. Its all-state selected gain is zero because every decision retains the frozen proposal. Campaign endpoints, selected-branch gains, and forecast errors therefore retain their respective denominators.

D.5 Selection of the illustrative variant case

We fix the schedule to six rounds of eight variants and order assays and seeds before examining hits, then take the first checkpoint with distinct proposals. Figure 3c shows both original batches and their elite membership at the resulting ACE2 checkpoint. Table 9 gives their zero-based library indices, nomination scores, and frozen scores for both batches. The latter determine the common current values and their difference δ. Within each selected batch, the reported nomination scores are distinct and positive. This example illustrates how the recommendation choice changes the measured variants and first-batch discoveries. Aggregate performance is assessed by the campaign and shared-state comparisons.

E Additional Results

E.1 Endpoint performance

Table 6: Terminal performance averaged over five seeds, with equal weight for schedules within each assay and for assays overall. Differences subtract each method's result from RESOLVE. The budget is 64 measurements, including initialization. Brackets give 95% paired assay-bootstrap intervals for recall differences, as defined in Appendix C.2.

Method	Recall (%)	Hits	Δ recall (pp)	Δ hits
RESOLVE	23.66	6.47	0.00	0.00
Random	2.87	0.72	20.78 [13.70, 27.99]	5.75
Zero-shot	10.82	3.59	12.84 [7.69, 18.57]	2.88
UCB	6.45	1.93	17.21 [11.19, 23.21]	4.53
LinUCB	10.18	2.91	13.48 [8.93, 18.05]	3.56
Diversity	4.65	1.33	19.01 [11.96, 26.12]	5.14
BatchBALD*	6.08	1.89	17.58 [11.33, 23.90]	4.58
1-batch VOI*	8.39	2.95	15.26 [9.79, 20.81]	3.52
Rollout*	6.27	2.05	17.39 [11.25, 23.57]	4.42
ALDE*	20.48	5.43	3.18 [1.54, 5.08]	1.04
EVOLVEpro*	15.25	4.77	8.40 [4.93, 12.26]	1.70
Always-Frozen	14.38	3.87	9.28 [6.14, 12.49]	2.60
Always-Tune	17.46	4.75	6.20 [4.08, 8.36]	1.72
Own-Score	19.62	5.38	4.04 [2.72, 5.37]	1.08
Select-Once	19.37	5.30	4.29 [2.91, 5.70]	1.17
Frozen-Single	10.51	2.74	13.15 [8.61, 17.79]	3.72
Full-Single	19.07	5.19	4.59 [3.03, 6.17]	1.28
Head-Only	19.94	5.43	3.72 [2.45, 5.01]	1.03
Head-Repeat	21.51	5.87	2.14 [1.45, 2.84]	0.60
MAP-Fantasy	22.37	6.10	1.29 [0.91, 1.67]	0.37

All methods have 144 assay–schedule endpoint records. Own-Score, Head-Only, and Select-Once implement different selection rules and have distinct mean recall values. Their campaign comparisons assess the resulting measurement histories, while the shared-state analysis holds the starting history and reference continuation fixed. For hindsight best-fixed, first average the six schedules for each assay and each fixed view, then select the larger mean. This selects Always-Tune in all 24 assays. Before schedule averaging, Always-Frozen exceeds Always-Tune in 36 cells, Always-Tune exceeds Always-Frozen in 106, and two are tied. These differences across schedules do not demonstrate changes in view preference within a campaign.

E.2 Fixed-view ordering across schedules

Table 7 separates the assay-level hindsight comparison from individual assay–schedule comparisons. The mean recall of Always-Tune is higher under every schedule when averaged over all assays, but particular assays can favor Always-Frozen. Hindsight best-fixed in Figure 3a chooses once per assay after schedule averaging, rather than choosing separately in every cell.

Table 7: Fixed-view comparisons using five-seed mean recall in each assay and schedule. Counts are out of 24 assays per schedule. Schedule-specific preferences differ from choosing the better fixed view after averaging schedules within each assay.

Rounds × batch size	Frozen higher	Tune higher	Tied
2×24	6	18	0
3×16	6	18	0
4×12	6	17	1
6×8	6	17	1
8×6	6	18	0
12×4	6	18	0
Total	36	106	2

E.3 SHARED-STATE DISCOVERY OUTCOMES

Table 8 reports all evaluated selectors on the shared reference states, including those omitted from the main figure for space. These are selected-branch gains over the frozen branch under the same continuation rule. They are not differences between independently evolving campaigns.

Table 8: Selected-branch gains over following Frozen, averaged across all 720 reference states with equal assay and schedule weighting. States that retain the frozen batch contribute zero. First and later gains form the total, up to rounding. Brackets give 95% paired assay-bootstrap intervals for total gains (Appendix C.2).

Selector	First batch	Later batches	Total [95% interval]
RESOLVE	1.404	1.793	3.197 [3.053, 3.342]
Own-Score	1.394	1.743	3.138 [2.986, 3.303]
Head-Only	0.896	1.126	2.022 [1.899, 2.147]
Head-Repeat	1.006	1.229	2.235 [2.113, 2.358]
MAP-Fantasy	1.244	1.581	2.825 [2.721, 2.936]

E.4 CANDIDATE IDENTITIES IN THE ILLUSTRATIVE CASE

Table 9: Candidate scores for Figure 3c at the ACE2 checkpoint after 32 measurements, with six rounds of eight variants and seed 0. Indices are zero-based. p_F evaluates both batches, while p_T ranks the adapted proposal. The displayed scores give $V_0(B_F) = 5.848$, $V_0(B_T) = 1.925$, and $\delta = 3.923$. RESOLVE selects Frozen, finding four elites versus one. Elite labels are hidden before selection.

	Frozen proposal			Adapted proposal			
Rank	Index	p_F	Elite?	Index	p_T	p_F	Elite?
1	1117	0.915	Yes	1431	0.819	0.208	Yes
2	1290	0.846	Yes	1377	0.756	0.139	No
3	1586	0.839	Yes	662	0.702	0.293	No
4	2008	0.822	Yes	1078	0.687	0.371	No
5	1061	0.727	No	1149	0.511	0.240	No
6	832	0.656	No	383	0.447	0.201	No
7	956	0.563	No	1683	0.416	0.427	No
8	454	0.480	No	461	0.384	0.046	No

E.5 GAIN-FORECAST ACCURACY

On the 352 common head-rejected states, the macro MAEs are 1.979 hits for the initial head forecast, 2.423 for Head-Repeat, and 1.125 for the representation forecast. Representation has lower MAE than either head evaluation in all 144 reported assay–schedule cells. After averaging schedules within assay, representation-minus-repeat differences range from -1.504 to -1.076 hits, with a mean of -1.298 hits (Figure 4d). The relative reductions of 53.6% against repetition and 43.1% against the initial head forecast are computed from the macro MAEs. The 95% assay-bootstrap interval for the mean representation-minus-repeat difference is $[-1.354, -1.244]$ hits. Against the initial head evaluation, the mean difference is -0.854 hits with interval $[-0.912, -0.796]$. These intervals summarize assay-level mean errors on the shared reference subset.

Feature coverage and the representation-stage comparison. For 16 assays, $N - 16 \leq 4096$, so the adapted shortlist covers every unmeasured candidate from initialization onward. On these assays, the head and representation stages have identical full-library coverage. RESOLVE exceeds Head-Repeat by 3.13 pp, so coverage expansion alone cannot explain the gain (Table 10). The remaining eight assays use a restricted adapted map and show only a 0.17 pp mean gain, despite lower forecast MAE. Improved prediction on the shared reference states therefore does not ensure comparable gains over an evolving campaign. Restricted nomination and mixed frozen/adapted feature coverage may limit this translation, but the groups also differ in library size and assay composition. This descriptive comparison cannot determine the cause of the smaller gain or isolate feature learning from the other changes in the representation stage.

Table 10: Existing results grouped by adapted-feature coverage. Recall gain is RESOLVE minus Head-Repeat in percentage points. MAEs are in hits on the shared head-rejected states. Seeds and schedules are averaged within assay before giving assays equal weight within each group.

Coverage	Assays	Recall gain	Repeat MAE	Representation MAE
Full library	16	3.13	2.447	1.144
Shortlist	8	0.17	2.375	1.087

E.6 Component effects by feedback schedule

Table 11 provides the component comparisons for each schedule. Each entry averages 24 assay-level five-seed means. The two differences compare the full method with different controls and are not additive contributions.

Table 11: Campaign performance and two matched component contrasts by feedback schedule. All schedules acquire 48 variants after 16 initial measurements. Positive differences favor RESOLVE.

Schedule	RESOLVE recall (%)	RESOLVE − MAP-Fantasy (pp)	RESOLVE − Head-Repeat (pp)
2×24	19.94	1.03	1.76
3×16	22.44	1.30	2.12
4×12	24.05	1.31	2.14
6×8	24.88	1.33	2.22
8×6	25.69	1.42	2.40
12×4	24.96	1.35	2.22