WHEN REUSE REQUIRES NEW DATA: DECIDING WHAT TO MEASURE AND WHEN TO STOP

Anonymous authors
Paper under double-blind review

ABSTRACT

Reusing past data can reduce the need for new data collection, but discovering how to use it in a new environment can itself require costly measurements. We study this as a decision problem whose objective is to minimize the combined burden of prediction error and measurement cost, rather than pursue accuracy regardless of the cost of obtaining it. We develop a procedure that uses selected measurements from the new environment to choose among predictors trained on past data and correct their predictions for other, unmeasured cases. Known relationships between cases guide the choice of what to measure. Before collecting more, the procedure considers possible results and calculates which predictor it would choose and how it would correct the predictions, without retraining the source models. It continues only when the expected improvement justifies the cost, then updates the choice and corrections using the actual result. The decision to acquire more is thus based on the prediction improvement expected from using the results, rather than uncertainty reduction alone. We evaluate using existing records from two sources: the ChEMBL bioactivity database, recording how strongly molecules bind to proteins, and the Hardware-adaptive Efficient Latency Predictor (HELP) study, recording how long neural networks run on devices. In one molecular example, the procedure uses 28.6% fewer measurements than a reference that uses all candidate measurements for the same predictor selection and correction, with an observed increase of about 0.7% in log-scale mean absolute error. In one hardware example, it uses 89.1% fewer measurements than the corresponding reference, with an observed increase of about 26% in that error measure. Mechanism analyses examine how three elements interact within the procedure: relationships that let measurements inform other cases, corrections that match the errors that remain, and judgments of whether further measurement is worth its cost.

1 INTRODUCTION

Collecting evidence in a new environment consumes time, experimental capacity, or human effort. Reusing past data can reduce that burden, but sources differ in relevance (Wang et al., 2014). Recommendation and language-model evaluation illustrate the appeal of earlier records for limiting new questions or tests (Boutilier et al., 2003; Maia Polo et al., 2024). Yet checking what to reuse can itself consume the resources reuse was meant to save.

We ask how much new evidence is worth acquiring for the predictions at hand. We start with predictors trained on past data, called source predictors, and past error estimates that guide their initial selection. Each new measurement checks every predictor and supplies errors that, through relationships between cases, help correct predictions for unmeasured cases. The source predictors remain fixed; their selection and corrections can change.

We study molecular binding using records from the ChEMBL bioactivity database (Gaulton et al., 2012; Zdrazil et al., 2024) and neural-network execution times using records from the HELP latency-prediction study (Lee et al., 2021). An environment is the setting in which responses are measured: a binding experiment (assay) for a protein, or a device. We measure selected molecules or architectures to inform a separate, fixed prediction set whose structures are known but responses remain unobserved.

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 first choose what to measure by proposing a measurement informative about that prediction set. Choosing when to stop requires valuing it before seeing its response. An outcome model—a probability model prepared from past data and updated with current measurements—describes possible measurement responses and how likely they are. For each hypothetical response, we calculate which predictor we would select, how we would correct its outputs, and whether a follow-up measurement would be worthwhile. We acquire the proposed response when the expected improvement justifies the measurement costs, then reassess using its actual value (Kapoor et al., 2007).

In one ChEMBL evaluation, this procedure saves about 29% of measurements with an observed increase of about 0.7% in log-scale mean absolute error, relative to using every candidate measurement with the same prediction rule. We also assess ranking and shortlist quality.

We compare choosing what to measure with random selection at equal counts, and choosing when to stop with a count selected in advance on separate tasks. Their contributions differ between ChEMBL and HELP. Alongside an executable procedure and measurement–prediction trade-offs (Sections 3–4.4), we analyze the mechanisms linking useful corrections to the recognition of their value (Section 4.5).

2 Related Work

Bioactivity studies illustrate the difficulty of choosing past data (Landrum and Riniker, 2024): matching assay metadata improved agreement but excluded training data, without resolving the trade-off in the evaluated multitask models (Schiebroek et al., 2026); AssayMatch selects assays from descriptions (Fan and Barzilay, 2026). These studies address which past data to use for model construction. We ask how much new evidence is worth acquiring to check and correct prepared predictions.

Model Selector buys labels to identify the best pretrained classifier (Okanovic et al., 2025); Active Learning with Model Selection allocates labels between training and validation (Ali et al., 2014). In our procedure, past error estimates initialize selection without a separate validation purchase. Each new response supports both selection and correction, so it can help even when model choice stays unchanged.

Offset-based transfer (Du et al., 2017), molecular correction libraries (Rodgers et al., 2007), and Δ-machine learning (Ramakrishnan et al., 2015) provide correction precedents. We combine correction with decisions about which measurements to acquire and when to stop, for a separate prediction set.

Active transfer learning couples adaptation and acquisition through source reweighting (Chattopadhyay et al., 2013) or source–target offsets and predictive covariance (Wang et al., 2014). Our kernel correction system is sized by the acquired set and shared across source predictors. This makes repeated selection and correction for hypothetical responses possible without retraining those predictors.

Our purchase decision follows cost-sensitive active learning (Kapoor et al., 2007; Diethe et al., 2015), rather than stopping on prediction agreement (Bloodgood and Vijay-Shanker, 2009) or generalization-error bounds (Ishibashi and Hino, 2020). Following Hübotter et al. (2024), our measurement-selection rule, which we call posterior geometry, proposes a case informative about the prediction set. Unlike the cited one-step valuations, we assess the selected-and-corrected predictions after a first response and an optional follow-up (Section 3.6). We then examine whether this calculation recognizes useful correction opportunities (Section 4.5).

3 Theory and Method

3.1 Task, Available Information, and Acquisition Cost

Let $\mathcal{P}$ be the acquisition pool of cases we may measure and $\mathcal{Q}$ the disjoint set whose responses we must predict. Descriptors x_i are known for all $i \in \mathcal{P} \cup \mathcal{Q}$; only responses in $\mathcal{P}$ may be acquired, one at a time and each at most once.

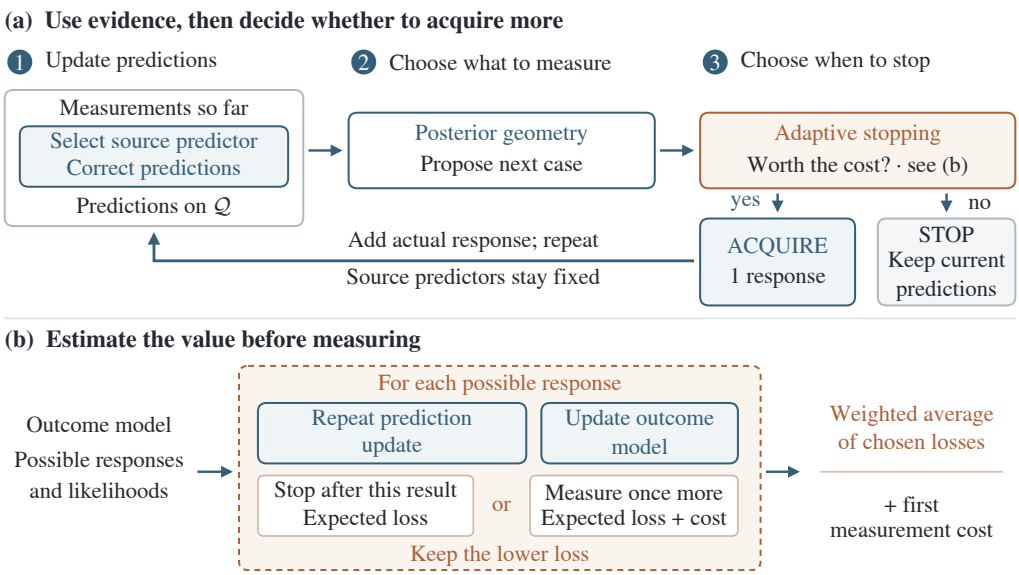

Figure 1: Using measurements and deciding whether to acquire more. (a) Measurements guide source-predictor selection and correction on prediction set $\mathcal{Q}$. Posterior geometry proposes a measurement; adaptive stopping decides whether to buy it. (b) For each hypothetical response from the outcome model, calculate prediction and outcome-model updates, then choose between stopping and measuring once more. Average the lower cost-inclusive losses using response likelihoods and add the first measurement cost. If this is below the expected loss of stopping now, acquire one actual response and reassess (Section 3.6).

An episode is one complete acquisition-and-prediction run. Its history h records the ordered case–response pairs acquired so far; $H = H(h)$ contains their case identities. The vector $F(h)$ gives the selected-and-corrected predictions on $\mathcal{Q}$ (Figure 1).

For a prediction vector f, with prediction f_j and response y_j at case j, mean absolute error (MAE) is $L_{\mathcal{Q}}(f; y_{\mathcal{Q}}) = \frac{1}{|\mathcal{Q}|} \sum_{j \in \mathcal{Q}} |f_j - y_j|$.

The withheld responses $y_{\mathcal{Q}}$ are used only for evaluation. Each measurement has cost $\lambda \geq 0$, expressed in prediction-loss units and fixed within an episode. At stopping history h_{stop}, with $|H_{stop}|$ acquired responses, we evaluate the completed episode by prediction error plus total measurement cost:

$$L_{\mathcal{Q}}(F(h_{\text{stop}}); y_{\mathcal{Q}}) + \lambda |H_{\text{stop}}|. \tag{3.1}$$

3.2 Prepared Predictions and Initial Error Estimates

Using source data from other environments, we train M random-forest predictors $\{b_a\}_{a=1}^{M}$ (Breiman, 2001): a indexes a source predictor (prediction option), and $b_a(x_i)$ is its uncorrected output at case i. We fit the outcome model and select settings on separate development environments, then evaluate on test environments.

ChEMBL forms ten source subsets by accumulating whole same-protein measurement groups ranked by molecular similarity using balanced optimal transport (Peyré and Cuturi, 2019; Appendix B.1.3). HELP uses five source devices.

Initial absolute-error estimates $e_a(x_i)$ use past error experience and available case information, not current responses. Their mean over $\mathcal{Q}$ gives the initial selection score, $e_a^0 = \frac{1}{|\mathcal{Q}|} \sum_{j \in \mathcal{Q}} e_a(x_j)$.

ChEMBL learns these estimates from source predictions on held-out molecules; HELP uses errors on development devices (Appendix A.1). These initial estimates stay fixed during acquisition.

3.3 Selecting and Correcting Predictions

Selecting a source predictor. With k acquired responses and $s > 0$ weighting the initial score, selection uses the errors of uncorrected source predictions:

$$\widehat{e}_a(h) = \frac{se_a^0 + \sum_{i \in H} |y_i - b_a(x_i)|}{s + k}, \qquad \widehat{a}(h) = \arg \min_{a \in \{1,\dots,M\}} \widehat{e}_a(h). \tag{3.2}$$

Initially the score is e_a^0. Each response contributes an error for every source predictor, reducing the initial estimate's relative weight.

Correcting its predictions. Kernel ridge regression (KRR) transfers observed errors through a relationship kernel K. With regularization $\alpha > 0$, the corrected outputs are

$$f_a(h) = b_{a,\mathcal{Q}} + K_{\mathcal{Q},H} \left(K_{H,H} + \alpha I\right)^{-1} \left(y_H - b_{a,H}\right), \qquad F(h) = f_{\widehat{a}(h)}(h). \tag{3.3}$$

Here $K_{\mathcal{Q},H}$ links prediction and measured cases, $K_{H,H}$ links measured cases, and I is the identity matrix. Vectors $b_{a,\mathcal{Q}}$ and $b_{a,H}$ contain source outputs on the indicated sets; y_H contains the acquired responses. The residual $y_H - b_{a,H}$ guides corrections, which are zero before any measurements.

3.4 Modeling Unobserved Outcomes

To value a measurement before observing it, the outcome model assigns probabilities to possible responses for the candidate and cases in $\mathcal{Q}$. We use a mixture of Gaussian distributions, one per source predictor. Each component's initial mean is the predictor's output vector plus a bias estimated from development data (Appendix A.3).

After history h, $U(h)$ contains remaining acquisition candidates and all prediction cases. Their conditional response distribution is

$$\widehat{P}_h\left(Y_{U(h)}\right) = \sum_{r=1}^{R} w_r(h)\, \mathcal{N}\left(\mu_{r,U(h)}(h), C_{r,U(h),U(h)}(h)\right). \tag{3.4}$$

Here $R = M$. Component r has weight $w_r(h)$, mean $\mu_r(h)$, and covariance $C_r(h)$. The covariances describe how far responses may deviate from these means and how deviations vary together across cases; their parameters are fitted on development data and held fixed. Each measurement updates the outcome model in two ways. First, each component uses its covariances to update the means and uncertainties of unmeasured responses. Second, the model updates the component weights using the likelihood of the observed response under each component.

The expected prediction loss of stopping at h is

$$\widehat{R}_F(h) = \mathbb{E}_{\widehat{P}_h}\left[L_{\mathcal{Q}}(F(h); Y_{\mathcal{Q}})\right]. \tag{3.5}$$

The outcome model supplies probabilities for this loss; the corrected predictions $F(h)$ remain the outputs being evaluated (Appendix A.4.1).

3.5 Deciding What to Measure: Posterior Geometry

Posterior geometry favors measurements informative about $\mathcal{Q}$ (Hübotter et al., 2024; Cohn et al., 1996), averaging within-component variance reductions using weights $w_r(h)$. For component r at history h, $C_{jq}^{r,h}$ is the covariance between prediction case j and candidate q, and $C_{qq}^{r,h}$ is the candidate's variance. The score and proposal are

$$G(q \mid h) = \sum_{r=1}^{R} w_r(h)\, \frac{1}{|\mathcal{Q}|} \sum_{j \in \mathcal{Q}} \frac{(C_{jq}^{r,h})^2}{\max\{C_{qq}^{r,h}, \epsilon_{\text{var}}\}}, \qquad q_h \in \arg \max_{q \in \mathcal{P} \setminus H(h)} G(q \mid h). \tag{3.6}$$

Here $\epsilon_{\text{var}} = 10^{-10}$ safeguards division. Within each component, covariance describes how a candidate's response relates to the prediction set. A highly uncertain candidate can therefore receive a low score if its response tells us little about the cases in $\mathcal{Q}$. Measurements update covariances and weights and can change the proposal. All candidates are scored; only the proposed case receives detailed valuation.

3.6 Deciding When to Stop: Adaptive Stopping

Adaptive stopping compares expected prediction benefit with measurement cost (Kapoor et al., 2007; Dimitrakakis and Savu-Krohn, 2008; Diethe et al., 2015). Appending hypothetical response Y_1 for proposal q_h to h gives h_1. Posterior geometry then proposes q_2, whose response Y_2 gives h_2:

$$h_1 = h \oplus (q_h, Y_1)\,, \qquad q_2 = q_{h_1}\,, \qquad h_2 = h_1 \oplus (q_2, Y_2)\,.$$

Let $\widetilde{\mathbb{E}}_{h,q}$ denote numerical expectation over candidate q's modeled responses at h. With at least two candidates remaining, continuation has expected cost-inclusive loss

$$\widehat{A}_2(h;\lambda) = \lambda + \widetilde{\mathbb{E}}_{h,q_h}\left[\min\left\{\widehat{R}_F\left(h_1\right),\ \lambda + \widetilde{\mathbb{E}}_{h_1,q_2}[\widehat{R}_F\left(h_2\right)]\right\}\right]. \tag{3.7}$$

For each hypothetical first response, update both predictions and the outcome model. The minimum compares stopping there with measuring once more; the outer λ pays for the first measurement. Only future costs enter this comparison. This calculation can favor a first measurement even when its immediate expected improvement does not cover its cost, because it also accounts for the benefit and cost of a follow-up chosen after seeing that response.

Acquire the proposed response if expected net gain $\widehat{R}_F(h) - \widehat{A}_2(h;\lambda)$ exceeds numerical tolerance $\epsilon_{\text{stop}} = 10^{-10}$; otherwise return $F(h)$.

Only one actual response is acquired before replanning. With one candidate left, value only that measurement; with none, stop. See Appendix A.5 for numerical integration and Algorithm 1.

4 Experiments

4.1 Tasks and Evaluation Protocols

For each ChEMBL task, one published protein-binding assay is the current environment (Table 1). Source data come from other publications on the same protein, excluding the current publication and every molecule in that environment. Each test protein thus has its own source data, separate from the development proteins used for fitting and setting selection (Appendix B).

Table 1. Data roles and task sizes. Three predefined assignments to acquisition pool P and prediction set Q define separate tasks in each environment, without using responses (Appendix B)**.**

Role or setting	ChEMBL	HELP
Development group	57 proteins	6 devices
Test group	15 proteins	4 devices
Source predictors	10 same-protein subsets	5 source devices
Cases per task ($\mathcal{P}$ / $\mathcal{Q}$)	15 / 15 molecules	42 / 42 architectures
Measurement-cost settings	34	26

ChEMBL v37 responses are pKi, the negative base-10 logarithm of inhibition constant K_i in molar units (Gaulton et al., 2012). HELP uses natural-log latency, with MAE termed log-MAE (Lee et al., 2021).

The relationship kernel uses molecular structure in ChEMBL and edge operations and input–output connectivity in HELP (Appendix A.2.3).

Development cross-validation keeps proteins or devices together when choosing settings; the outcome model is then fitted on all development data before testing (Appendix B.3). HELP losses and costs are converted together to reported log-MAE units (Appendix A.6.4). We report per-measurement costs as c in each application's loss units. The original policy uses $\lambda = c$ in ChEMBL and the corresponding internal-unit cost in HELP.

All-15 and all-42 acquire every response in $\mathcal{P}$ and use the same final prediction rule F. We report MAE and measurement savings relative to them.

Measurement-selection comparison. To test choosing what to measure, posterior geometry and random selection use equal counts and the same prediction rule; random errors average 32 orders per task.

Stopping comparison. We compare adaptive stopping with a count fixed across test tasks at each cost, chosen on development data to minimize error plus measurement cost. Both policies use posterior geometry to choose what to measure and the same prediction rule, but may acquire different counts.

Differences are calculated within tasks and averaged equally over test proteins or devices. Bootstrap resampling keeps each unit's tasks together. We report 90% pointwise confidence intervals (CIs) per setting and approximate 90% simultaneous bands per curve (Appendix B.4).

4.2 ChEMBL: Measurement–Prediction Trade-offs

At $c \approx 0.00158$, the procedure uses 10.71 measurements on average, saving 28.59% relative to all-15. MAE is 0.46817 versus 0.46495 pKi, a within-task increase of 0.00321 $[-0.00281,\ 0.00972]$ (90% pointwise interval).

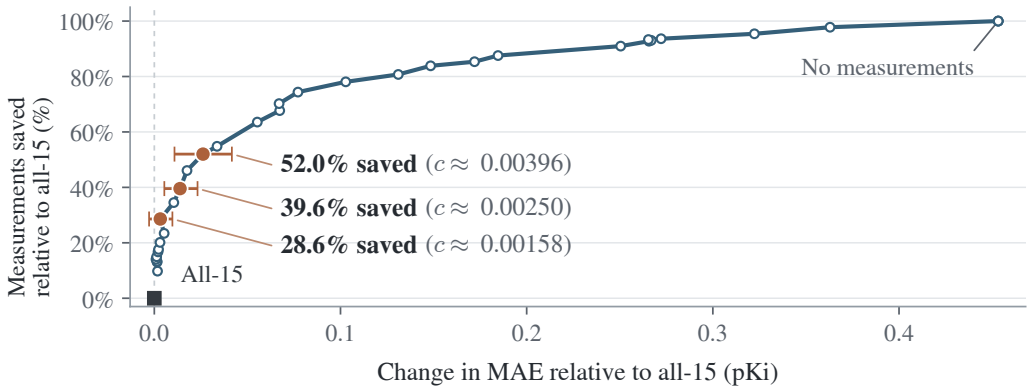

Figure 2: ChEMBL measurement savings and MAE change relative to all-15 using the same prediction rule. The curve connects 34 settings in increasing cost order. Horizontal bars show 90% pointwise protein-bootstrap CIs for mean within-task MAE differences (Appendices B.4 and C.1).

With no measurements, MAE is 0.91822 pKi, an increase of 0.45327 over all-15 (Figure 2).

At the 28.59% saving setting, the predicted top-five shortlist retains a highest-affinity molecule in 62.22% of tasks, as does all-15. However, it retains 57.33% of the actual top five versus 59.11% for all-15 (top-five recall), and Spearman rank correlation is lower (0.4819 versus 0.5072 for all-15). Small mean-error changes therefore need not preserve ranking quality (Appendix C.1).

4.3 ChEMBL: Measurement-Selection and Stopping Comparisons

Choosing what to measure. Posterior geometry has lower mean MAE than random selection at every intermediate count (1–14; Figure 3a; Appendix C.2.1).

Choosing when to stop. We compare completed runs using $J_c = L_{\mathcal{Q}}(F(h_{\text{stop}}); y_{\mathcal{Q}}) + c \mid H_{\text{stop}} \mid$. The evaluation cost c stays fixed when tuning the stopping setting λ.

At $c \approx 0.00158$, adaptive stopping uses 10.71 versus 12 measurements, with error-plus-cost difference +0.00037 [-0.00481, 0.00497] (90% pointwise interval). Thus fixed counts also save measurements. Neither the full curve nor development-tuning of the stopping setting establishes a general adaptive-stopping advantage (Figure 3b; Appendix C.2.3).

Valuing the final prediction rule. The question is whether purchase decisions improve when measurements are valued through the prediction rule that will actually use their results. We compare expected loss of our selected-and-corrected predictions F with minimum expected absolute loss

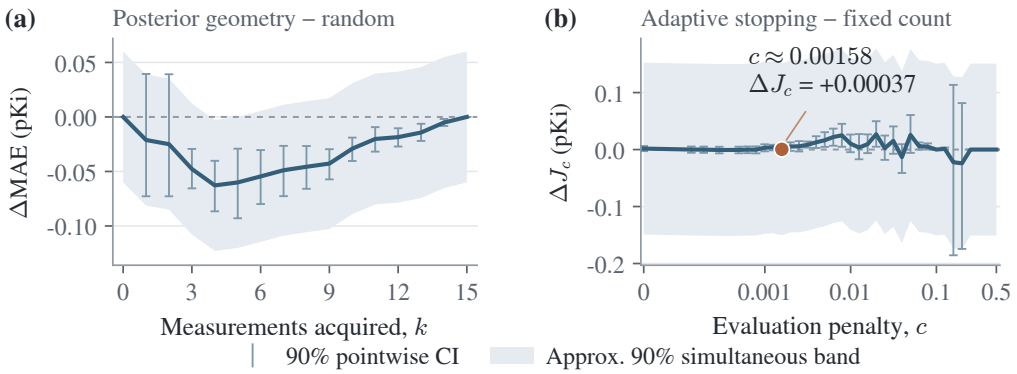

Figure 3: **ChEMBL** measurement-selection and stopping comparisons. Differences are (a) MAE for posterior geometry minus random selection at equal counts and (b) adaptive stopping minus the development-selected fixed-count policy in error-plus-cost J_c (Section 4.3). Negative values favor the proposed rule. Bars are 90% pointwise protein-bootstrap intervals; ribbons are approximate 90% simultaneous bands, below zero only at counts 4–5 in (a).

under the same outcome model, attained by posterior-median predictions. Both procedures use posterior geometry and the same final prediction rule F. Different valuations can change when they stop and how many measurements they acquire, letting us assess the resulting balance between prediction error and measurement cost. With uncalibrated gains and development-selected stopping settings, their mean J_c difference (F-based minus median-based) is +0.00251 [-0.00136, 0.00670] (90% interval), not establishing an overall F-based advantage (Appendix C.5).

4.4 HELP: Hardware Latency Prediction

Figure 4 shows the HELP trade-off and HELP10, the separate predictor of Lee et al. (2021) using ten measured reference architectures. The curve ends at no measurements, with log-MAE 0.48348.

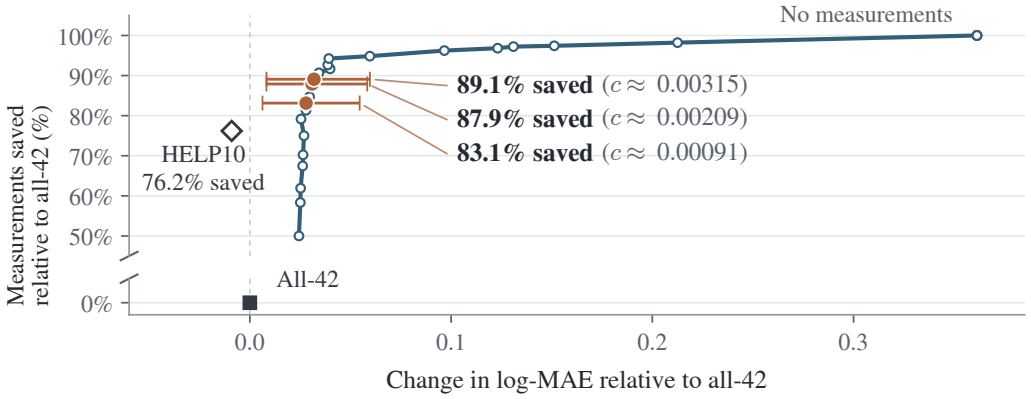

Figure 4: HELP measurement savings and log-MAE change relative to all-42 using the same prediction rule. The curve connects 26 settings in increasing cost order. Horizontal bars show 90% pointwise device-bootstrap CIs for mean within-task log-MAE differences. HELP10 uses a separate predictor (Appendices B.4 and C.3).

At $c \approx 0.00315$, our procedure uses 4.58 measurements on average, saving 89.09% relative to all-42. Log-MAE is 0.15404 versus 0.12218, a within-task increase of +0.03185 [0.00828, 0.05968] (90% pointwise interval).

Comparison with the HELP predictor. Its published network and adaptation rule are retained; reference responses count as paid, and target-device normalization and adaptation use only acquired

responses. At ten measurements, our log-MAE is 0.13801 versus HELP10's 0.11315, but mean absolute relative latency error is lower: 14.29% versus 22.15% (Appendix C.3.2). These metrics weight errors differently. For example, predicting twice or half the true latency gives the same absolute log error, but absolute relative errors of 100% and 50%, respectively. They can therefore rank prediction methods differently.

Choosing what to measure. In HELP, posterior geometry has higher mean error than random selection at the first measurement but lower mean error at some later counts (Appendix C.3.3).

Choosing when to stop. At the same cost $c \approx 0.00315$, the fixed-count policy uses five measurements with log-MAE 0.20879. Adaptive stopping uses 4.58 measurements on average and has lower mean error, giving a J_c difference of -0.05607 [-0.12525, -0.00042] (90% pointwise interval); Appendix C.3.4 gives the full cost curves.

Computation and measurement time. Additional computation can be offset by measurements avoided. Assuming 100 ms per sequential measurement, computation plus measurement time at the 4.58-measurement setting is 0.524 s, versus 1.034 s for HELP10 and 4.204 s for all-42. These illustrative totals exclude offline preparation and loading (Appendices B.5 and C.3).

4.5 Understanding the Measurement-Selection and Stopping Results

Why do measurement selection and stopping contribute differently across applications? Holding fitted models and original policies fixed, we examine correction effects, their usefulness, and the recognition of further gains.

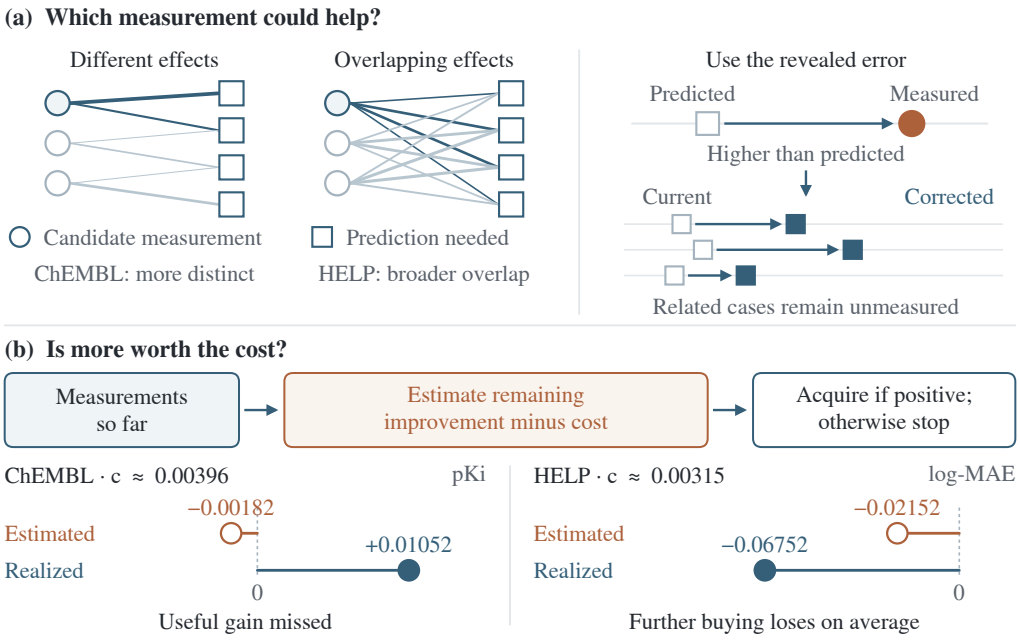

Figure 5: Correction opportunities and their recognition. (a) Conceptual correction effects: arrows show prediction changes. (b) At stopping histories, open and filled markers show expected and realized mean net gains from the proposed measurement and optional follow-up. Net gain is prediction-error reduction minus cost; positive values favor continuing. Segments join means to zero; loss units and scales differ between panels (Appendix C.4).

Choosing what to measure. Candidates may differ in the corrections they provide: an error revealed at one case can guide corrections to related cases (Figure 5a).

Holding the selected source predictor a fixed isolates candidate j's correction (Appendix A.2.4):

$$\Delta_{a,j}(h) = f_a\left(h \oplus (j, y_j)\right) - f_a(h) = A_{h,j}\, z_{a,j}(h).$$

Here $z_{a,j}(h)$ is the response minus its current corrected prediction; $A_{h,j}$ maps a unit residual to changes across prediction cases, using the kernel, regularization, and acquired cases.

Over counts 1–14, correction effects are more distinct in ChEMBL than HELP (mean signed cosine similarity 0.474 versus 0.743). Greater variation in ChEMBL correction strength accompanies larger gains over random selection (Appendix C.6.2). Correction strength describes how much a unit residual can move predictions, not whether those movements reduce error. Actual improvement depends on the residual revealed at the measured case and the errors remaining in the prediction set.

To test whether the residual revealed at a particular candidate makes its correction useful, we compare using each candidate's own residual with reassigning the same residual values among candidates, each value used once. We keep the correction coefficients and current prediction-set errors fixed. Using each candidate's own residual gives greater mean error reduction than the exact average over all such reassignments, by 0.00965 in ChEMBL and 0.00272 in HELP after normalization by each task's initial MAE (Appendix C.6.3).

Choosing when to stop. We examine whether the stopping calculation recognizes when further measurements would improve predictions enough to cover their cost. From each history of acquired case–response pairs, we retrospectively evaluate the proposed measurement and optional follow-up defined in Section 3.6. Realized net gain is prediction-error reduction minus measurement cost.

The ChEMBL checks identify useful correction opportunities missed by adaptive stopping. At measurement costs $c \approx 0.00250$ and $c \approx 0.00396$, carrying out the rejected measurement and optional follow-up gives mean net gains of 0.00623 and 0.01052 pKi, respectively, despite negative estimates. Reassigning initial prediction-set errors while retaining predictions and decisions shows that later fixed-count predictions captured useful case-specific corrections. The stopping calculation underestimated gains from measurements it had already evaluated (Appendices C.4 and C.6.6).

In HELP, comparing histories at the same device, count, and cost shows that the updated net-gain score distinguishes larger from smaller realized gains better than its initial value (Appendix C.6.5).

At $c \approx 0.00315$, the continuation rejected by adaptive stopping has mean net gain -0.06752 log-MAE units [-0.14157, -0.00494] (90% pointwise interval), so stopping avoids this mean loss (Figure 5b).

5 Discussion and Limitations

The findings suggest assessing source predictions by how much limited evidence can correct them, not only their initial accuracy. Case relationships enable correction, but deciding whether to acquire more also requires anticipating its value. Because both valuation rules rely on the same outcome model (Section 4.3), improving purchase decisions calls for examining how it represents remaining errors together with the corrections measurements can provide.

Use also determines what constitutes a good trade-off: candidate prioritization requires evaluating ranking and shortlist quality, and resource accounting must include preparation and computation costs. Our retrospective evidence covers 15 test proteins and four devices. Extension requires checking whether new environments support useful corrections and informative stopping judgments, and whether prediction quality suits the task.

Investing in a task-specific model rather than limited correction likewise requires weighing preparation cost against predictive benefit and anticipated use, as in model-lifetime active-learning costs (Pullar-Strecker et al., 2024) and cost-aware retraining (Mahadevan and Mathioudakis, 2024).

6 Conclusion

We presented a procedure for reusing past data that selects and corrects source predictions using purchased measurements and values further measurements through the same updates. ChEMBL and HELP quantify measurement–prediction trade-offs and different contributions of measurement selection and stopping. Controlled analyses distinguish correction opportunities, their usefulness, and their recognition.

AI USE STATEMENT

The authors directed the research, checked the mathematical claims and proofs, and made the methodological, interpretive, and presentation decisions. Generative AI tools assisted with conceptual and mathematical development (including proof assistance), hypothesis refinement, and experimental design. They also supported method implementation, data preparation, analysis and interpretation of results, literature review, and manuscript preparation, including drafting, translation, revision, figures, and supporting materials.

The ChEMBL and HELP evaluations used existing measurement records, not AI-generated measurements. The authors reviewed and revised the manuscript and take responsibility for the entire submission, including all AI-assisted content.

REFERENCES

Alnur Ali, Rich Caruana, and Ashish Kapoor. 2014. Active Learning with Model Selection. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1673–1679*.

Michael Bloodgood and K. Vijay-Shanker. 2009. A Method for Stopping Active Learning Based on Stabilizing Predictions and the Need for User-Adjustable Stopping. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), pp. 39–47*.

Craig Boutilier, Richard S. Zemel, and Benjamin Marlin. 2003. Active Collaborative Filtering. Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, pp. 98–106.

Leo Breiman. 2001. Random Forests. Machine Learning, 45:5–32.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A Limited Memory Algorithm for Bound Constrained Optimization. SIAM Journal on Scientific Computing, 16(5):1190–1208.

Rita Chattopadhyay, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. 2013. Joint Transfer and Batch-mode Active Learning. *Proceedings of the 30th International Conference on Machine Learning, PMLR 28(3):253–261*.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active Learning with Statistical Models. *Journal of Artificial Intelligence Research*, 4:129–145.

Tom Diethe, Niall Twomey, and Peter Flach. 2015. Bayesian Active Transfer Learning in Smart Homes. *ICML Workshop on Advances in Active Learning: Bridging Theory and Practice*.

Christos Dimitrakakis and Christian Savu-Krohn. 2008. Cost-Minimising Strategies for Data Labelling: Optimal Stopping and Active Learning. In *Foundations of Information and Knowledge Systems*, Lecture Notes in Computer Science, 4932:96–111. Springer.

Xuanyi Dong and Yi Yang. 2020. NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search. *Proceedings of the Eighth International Conference on Learning Representations*.

Simon S. Du, Jayanth Koushik, Aarti Singh, and Barnabás Póczos. 2017. Hypothesis Transfer Learning via Transformation Functions. *Advances in Neural Information Processing Systems, 30:574–584*.

Vincent Fan and Regina Barzilay. 2026. AssayMatch: Learning To Select Data for Molecular Activity Models. Journal of Chemical Information and Modeling, 66(7):3540–3549.

C. A. Field and A. H. Welsh. 2007. Bootstrapping clustered data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):369–390.

Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. 2012. ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Research, 40(D1):D1100–D1107.

Stephen R. Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. 2015. InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics*, 7:23.

Jonas Hübotter, Bhavya Sukhija, Lenart Treven, Yarden As, and Andreas Krause. 2024. Transductive Active Learning: Theory and Applications. *Advances in Neural Information Processing Systems, 37:124686–124755.*

Hideaki Ishibashi and Hideitsu Hino. 2020. Stopping criterion for active learning based on deterministic generalization bounds. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, PMLR 108:386–397*.

Ashish Kapoor, Eric Horvitz, and Sumit Basu. 2007. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, pp. 877–882.

Gregory A. Landrum and Sereina Riniker. 2024. Combining IC50 or Ki Values from Different Sources Is a Source of Significant Noise. *Journal of Chemical Information and Modeling*, 64(5):1560–1567.

Hayeon Lee, Sewoong Lee, Song Chong, and Sung Ju Hwang. 2021. Hardware-adaptive Efficient Latency Prediction for NAS via Meta-Learning. Advances in Neural Information Processing Systems, 34:27016–27028.

Ananth Mahadevan and Michael Mathioudakis. 2024. Cost-aware retraining for machine learning. Knowledge-Based Systems, 293:111610.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinyBenchmarks: evaluating LLMs with fewer examples. Proceedings of the 41st International Conference on Machine Learning, PMLR 235:34303–34326.

Patrik Okanovic, Andreas Kirsch, Jannes Kasper, Torsten Hoefler, Andreas Krause, and Nezihe Merve Gürel. 2025. All models are wrong, some are useful: Model Selection with Limited Labels. *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics, PMLR 258:2035–2043*.

Gabriel Peyré and Marco Cuturi. 2019. Computational Optimal Transport: With Applications to Data Science. Foundations and Trends in Machine Learning, 11(5–6):355–607.

Zac Pullar-Strecker, Katharina Dost, Eibe Frank, and Jörg Wicker. 2024. Hitting the target: stopping active learning at the cost-based optimum. Machine Learning, 113:1529–1547.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. 2015. Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach. *Journal of Chemical Theory and Computation, 11(5):2087–2096*.

Sarah L. Rodgers, Andrew M. Davis, Nick P. Tomkinson, and Han van de Waterbeemd. 2007. QSAR Modeling Using Automatically Updating Correction Libraries: Application to a Human Plasma Protein Binding Model. *Journal of Chemical Information and Modeling, 47:2401–2407*.

David Rogers and Mathew Hahn. 2010. Extended-Connectivity Fingerprints. Journal of Chemical Information and Modeling, 50(5):742–754.

Carl C. G. Schiebroek, Gregory A. Landrum, and Sereina Riniker. 2026. Balancing Data Quantity and Quality: Evaluating Curation Strategies for Bioactivity Prediction in Lead Optimization. *Journal of Chemical Information and Modeling, 66(13):7446–7452*.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. 2001. A Generalized Representer Theorem. In Computational Learning Theory, Lecture Notes in Computer Science, 2111:416–426. Springer.

Xuezhi Wang, Tzu-Kuo Huang, and Jeff Schneider. 2014. Active Transfer Learning under Model Shift. Proceedings of the 31st International Conference on Machine Learning, PMLR 32(2):1305–1313.

David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36.

Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J. Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F. Mosquera, Maria Paula Maga-

rinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A. Patrícia Bento, Melissa F. Adasme, Peter Monecke, Gregory A. Landrum, and Andrew R. Leach. 2024. The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192.

A Mathematical and Computational Details

This appendix provides the derivations and numerical details of the calculations in Section 3. Throughout, $n = |\mathcal{P}|$, $m = |\mathcal{Q}|$, and $k = |H(h)|$. Matrix blocks and response vectors indexed by $H(h)$ use the same acquisition-history order.

A.1 Preparing Error Estimates and Updating the Selection Score

A.1.1 ChEMBL: source error prediction

For each prepared source membership, the activity model is a random forest (RF) regressor with 300 trees, square-root feature subsampling, unrestricted tree depth, and Morgan-count descriptors. A separate 300-tree forest predicts absolute errors, using all three error-feature coordinates at each split. Both forests use squared-error splitting and bootstrap sampling, with minimum leaf size one and minimum split size two. The error-regression targets are obtained by shuffled five-fold source cross-fitting grouped by molecule, so the prediction used to measure a source molecule's error is produced without that molecule in the fitting fold.

For a source example i, let $b_a^{(-v(i))}(x_i)$ be the out-of-fold (OOF) activity prediction. The error-regression target is

$$d_{a,i}^{\mathrm{OOF}} = \left| y_i - b_a^{(-v(i))}(x_i) \right|. \tag{A.1}$$

Activity forests use unscaled, 2048-dimensional chiral Morgan count fingerprints of radius two (Rogers and Hahn, 2010), stored as sparse `float32` inputs. Each error-regression input has three coordinates, in order: the population standard deviation across the 300 activity-tree predictions, their mean prediction, and the maximum binary-Morgan Tanimoto similarity to a reference source molecule. The tree standard deviation uses divisor 300 (`ddof=0`). For an out-of-fold source example, both the forest and the similarity reference use only its training fold; all assay records sharing the same standardized molecular identifier are held out together, and the reference contains the unique training-fold molecules. The identifier is the InChIKey, a hash derived from the International Chemical Identifier (InChI; Heller et al., 2015).

For a current molecule, the final activity forest and similarity reference use the complete selected source membership. Current and source molecule identities are required to be disjoint. The error forest is fitted to the out-of-fold absolute errors in Equation (A.1) and applied directly to the three current-input features to produce $e_a(x_i)$. Its output is neither clipped nor standardized; negative or nonfinite values are rejected. Its out-of-bag score is a diagnostic, not the initial absolute-error estimate.

The raw error estimates, averaged over the prediction set, supply e_a^0. The separate error floor used in A.3.1 does not modify this selection score. Source-side cross-fitting prepares the initial estimate; it does not make subsequently acquired current labels an independent validation sample.

The ChEMBL source subsets are constructed by accumulating the ranked blocks as described in Section 3.2. Each nominal fraction refers to the number of source records, not the number of blocks. Because a block is kept intact, a subset can exceed its nominal fraction, and two nominal fractions can produce the same subset; identical subsets can reuse the same source-model fit.

A.1.2 HELP: initial errors from development devices

The HELP evaluation uses 15 devices in three disjoint groups with fixed roles: five source devices, six development devices, and four test devices. Source devices supply the prepared prediction bank.

Development devices supply the initial error estimates and the data used to fit the outcome model and choose settings. Test devices remain outside these preparation steps.

The HELP bank consists of five source-device RF predictors. Each architecture is represented by 30 binary operation indicators: five indicators for each of the six edge positions in its cell. The source forests use 128 trees, minimum leaf size two, and all 30 indicators, with `max_features=1.0`. They are fitted in source-normalized log-latency units. The retained evaluation reuses this prepared bank; the forests are not refitted after a current-device measurement.

For a given architecture, we estimate each option's error by comparing its prediction with that architecture's measured latency on the allowed development devices. Let $\mathcal{D}$ be the set of devices supplying these errors. The initial estimate is

$$e_a^{\mathcal{D}}(x) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} |y_d(x) - b_a(x)|\,, \tag{A.2}$$

where $y_d(x)$ is the source-normalized log latency of architecture x on development device d, and predictions use the same scale. Errors are computed separately at each architecture. The prior-error calculation applies no clipping. The covariance construction in A.3.1 subsequently floors each case-specific error scale at 0.02, while initial selection scores and mixture weights use the unfloored errors.

Development validation and final preparation. Within the six-device development set, one device is held out at a time to evaluate the selection and correction settings. Each validation fit uses the other five devices. After selecting the settings, the final outcome-model parameters are fitted using all six development devices and used for all four test devices. The devices supplying the error estimates in Equation (A.2) are:

Stage	Episode receiving an error estimate	Devices in $\mathcal{D}$
Development validation	The held-out development device	The other five development devices
Fitting a development-validation model	One of that fit's five development devices	The other four devices in that fitting set
Fitting the final model	One of the six development devices	The other five development devices
Test evaluation	Any of the four fixed test devices	All six development devices

Thus a fitting episode never supplies its own error estimate, and no test-device response enters any of these error estimates or preparation fits. During test acquisition, only the requested responses in $\mathcal{P}$ update selection, correction, and conditioning; the prepared initial error estimates remain fixed. Appendix A.3.1 gives the fitting populations.

A.1.3 The role of the initial score

For $k > 0$, define the observed base-error average

$$\overline{d}_a(h) = \frac{1}{k} \sum_{i \in H(h)} |y_i - b_a(x_i)|\,.$$

Then Equation (3.2) is equivalently

$$\widehat{e}_a(h) = \frac{s}{s+k} e_a^0 + \frac{k}{s+k} \overline{d}_a(h). \tag{A.3}$$

Thus s controls the weight of the fixed initial error estimate relative to errors revealed in the current history. For an empty history, we set $\widehat{e}_a = e_a^0$ directly. Ties are resolved by the fixed source-predictor order. The ChEMBL settings used on test proteins are $s = 0.25$, $\alpha = 0.1$; the HELP settings selected on the six development devices and held fixed for all four test devices are $s = 0.25$, $\alpha = 0.01$. The acquisition cost is a separate input.

Since the acquired cases are selected adaptively, $\overline{d}_a(h)$ need not be an unbiased estimate of error on the prediction set. Its role here is to update the specified selection score. The modeled mean absolute error (MAE) used to decide acquisition is computed for the resulting corrected output, as described in A.4.

A.2 Kernel Residual Correction

A.2.1 Objective and solution

For a fixed option a, define $r_a = y_H - b_{a,H}$. Kernel ridge regression (KRR) fits a correction function g_a in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ associated with K. The objective is the sum of squared observed residual errors plus a penalty on the function norm:

$$\sum_{i \in H} (r_{a,i} - g(x_i))^2 + \alpha \|g\|^2_{\mathcal{H}_K}. \tag{A.4}$$

This objective explains the correction formula; the deployed calculation is the corresponding small linear solve. Using the representer theorem (Schölkopf et al., 2001), we write $g(x) = K_{x,H}c$. The coefficient objective is

$$\|r_a - K_{H,H}c\|_2^2 + \alpha c^\top K_{H,H} c.$$

Its stationarity condition is

$$K_{H,H}\left[(K_{H,H} + \alpha I)\, c - r_a\right] = 0.$$

For positive-semidefinite K and $\alpha > 0$, the choice

$$c_a = (K_{H,H} + \alpha I)^{-1} r_a \tag{A.5}$$

is a minimizing representation and yields Equation (3.3). If $K_{H,H}$ is singular, coefficient representations may not be unique; the regularized fitted function is the object of interest. The objective uses a **sum**, rather than a mean, of squared residual errors, consistent with the displayed $K_{H,H} + \alpha I$ solve. Averaging that objective without adjusting α would change the correction.

The correction uses a squared-residual fitting objective, while option selection and acquisition use their separately defined absolute-error criteria.

A.2.2 Shared solve across prediction options

Let $B_H \in \mathbb{R}^{k \times M}$ collect the base predictions at the acquired cases. Define

$$D_H = y_H \mathbf{1}_M^\top - B_H, \qquad C_H^{\text{corr}} = (K_{H,H} + \alpha I)^{-1} D_H. \tag{A.6}$$

The matrix of corrected predictions on the prediction set is

$$B_{\mathcal{Q}} + K_{\mathcal{Q},H} C_H^{\text{corr}}.$$

The selected column is $F(h)$. The scalar implementation solves one $k \times k$ system simultaneously for all prediction options. HELP instead computes the kernel transfer matrix to the prediction set

$$T_H^{\text{corr}} = \left[\text{solve}\left(K_{H,H} + \alpha I, K_{H,\mathcal{Q}}\right)\right]^\top$$

and applies it to the residuals of all options and response vectors in a batch. Both calculations use the stated α, without scaling it by the number of measurements or adding further regularization. Under a mean-squared-residual convention, the penalty coefficient equivalent to the sum objective

in Equation (A.4) would be α/k for $k > 0$. Neither path uses a pseudoinverse, an additional-jitter retry, or a fallback solve. An empty history returns the base predictions directly.

Only the base residuals are used on the right-hand side. Residuals of an already corrected predictor are not recursively substituted for them. Recomputing the correction from the acquired set therefore avoids counting the same residual update twice.

A.2.3 Relationship kernels and outcome covariances

ChEMBL. Let $z(x) \in \{0,1\}^{2048}$ be the chiral binary Morgan fingerprint of radius two. The correction kernel is

$$K(x, x') = \frac{z(x)^\top z(x')}{\|z(x)\|_1 + \|z(x')\|_1 - z(x)^\top z(x')}. \tag{A.7a}$$

The assembled matrix is symmetrized and its diagonal is set to one. No additional distance transformation, bandwidth, normalization, centering, eigenvalue projection, or kernel nugget is applied. This binary-fingerprint kernel is distinct from the count-fingerprint input to the activity forest.

Source blocks are ordered using a balanced optimal transport (OT) distance between empirical structural distributions, with $1 - K$ as the ground cost. That comparison uses the full current representative assay's molecular structure pool and each source block, rather than only the current episode's scoring molecules. Uniform mass is assigned to model rows, retaining the multiplicity of a molecule that occurs in several source rows; compressed source atoms carry the corresponding row-count mass. This distributional distance is not the pairwise correction kernel in Equation (A.7a).

Empty binary fingerprints are rejected during preparation. On nonempty active-bit sets, the exact Tanimoto kernel is positive semidefinite: it is the probability that the first active bit under a shared random permutation agrees for the two sets, equivalently an average of inner products of one-hot vectors. This supports the RKHS interpretation in A.2.1. The floating-point matrix is checked for symmetry, range, and unit diagonal; its preparation does not perform an eigenvalue certification.

HELP. The retained kernel uses the number of differing edge operations $d_{\text{op}}(x, x')$ and the input-to-output connectivity regime $r(x)$:

$$K_0(x, x') = \frac{1}{2} + \frac{1}{2}\exp\{-d_{\text{op}}(x, x')/2\},$$

$$K(x, x') = [\gamma + (1-\gamma)\mathbf{1}\{r(x) = r(x')\}]\, K_0(x, x'), \qquad \gamma = 0.5. \tag{A.7}$$

The regime $r(x) \in \{0,1\}$ indicates whether at least one live path connects the cell input to its output; a `none` operation is an absent edge. This is a connectivity indicator, not a declaration about compiler behavior or the numerical value of the cell output. The operation distance counts disagreements across the six corresponding edges. Within-regime similarities are unchanged, while cross-regime similarities are multiplied by $\gamma = 0.5$. The kernel diagonal remains one, with no further normalization.

The regime factor acts on $\rho D_r K D_r$ in Equation (A.8a). It does not multiply the global offset covariance $\tau^2 \mathbf{1}\mathbf{1}^\top$, the independent diagonal term $(1-\rho)D_r^2$, or the fitted and numerical diagonal floors. Thus it does not attenuate every form of dependence between regimes in the outcome model. The exact kernel is positive semidefinite: the exponential of the operation Hamming distance is a product of categorical edge kernels, and the constant addition and positive-semidefinite regime factor preserve this property.

The coefficient α regularizes the residual correction. The outcome covariance uses its own fitted variance terms, specified in Appendix A.3.1.

A.2.4 How one new measurement changes other predictions

We examine how the error revealed by a new measurement changes predictions for other cases, and how much those predictions improve. To separate correction from a change of prediction option, we

keep option a fixed for this one-measurement comparison. The kernel and regularization coefficient remain those of Equation (3.3).

The error revealed at the new case. Let $H = H(h)$ be the acquired cases, $j \in \mathcal{P}\backslash H$ a remaining candidate, and $S_H = K_{H,H} + \alpha I$. Before acquiring the response at j, we can already predict its value by applying the correction based on h to option a. Denote this current corrected prediction by $\widetilde{b}_{a,j}(h)$. Once y_j is revealed, we subtract that prediction from the observed value:

$$\widetilde{b}_{a,j}(h) = b_a\left(x_j\right) + K_{j,H} S_H^{-1}\left(y_H - b_{a,H}\right), \qquad z_{a,j}(h) = y_j - \widetilde{b}_{a,j}(h). \tag{A.7b}$$

The difference $z_{a,j}(h)$ is the residual relative to the current corrected prediction. It is positive when the observed value is higher than predicted and negative when it is lower. This is not the residual against the original source prediction: earlier measurements have already contributed to the current prediction.

How this residual changes other predictions. For each case in $\mathcal{Q}$, a kernel-based coefficient specifies how much its prediction changes per unit of the newly revealed residual. Collect these coefficients into a vector $A_{h,j} \in \mathbb{R}^m$:

$$\begin{aligned} d_{h,j} &= K_{j,j} + \alpha - K_{j,H} S_H^{-1} K_{H,j}, \\ A_{h,j} &= \frac{K_{\mathcal{Q},j} - K_{\mathcal{Q},H} S_H^{-1} K_{H,j}}{d_{h,j}}. \end{aligned} \tag{A.7c}$$

Write $h^+ = h \oplus (j, y_j)$ for the enlarged history. The change in predictions when we update the same option is exactly

$$\Delta_{a,j}(h) = f_a\left(h^+\right) - f_a(h) = A_{h,j}\, z_{a,j}(h). \tag{A.7d}$$

Each prediction in $\mathcal{Q}$ therefore changes by the residual multiplied by its corresponding coefficient. These coefficients can differ in both sign and size, so the adjustment need not be the same at every case.

Collecting the coefficient vectors for all remaining candidates gives $A_h \in \mathbb{R}^{m \times (n-k)}$. Using the same candidate order, their prediction changes form $\Delta_a(h) = A_h diag\left(z_a(h)\right)$. At fixed H, A_h depends only on K and α; the acquired positions themselves may depend on earlier responses. These are coefficients of the correction rule, distinct from the outcome-model covariances $C_r(h)$ and the posterior geometry score in Equation (3.6).

Derivation. For positive-semidefinite K and $\alpha > 0$, the denominator in Equation (A.7c) is the positive Schur complement of the enlarged regularized kernel matrix. Let $c_a = S_H^{-1}\left(y_H - b_{a,H}\right)$. In the enlarged correction solve, write c_H^+ for the coefficients on the previous measurements and c_j^+ for the new coefficient. Eliminating c_H^+ gives

$$c_j^+ = \frac{z_{a,j}(h)}{d_{h,j}}, \qquad c_H^+ = c_a - S_H^{-1} K_{H,j} c_j^+.$$

Substituting into $b_{a,\mathcal{Q}} + K_{\mathcal{Q},H} c_H^+ + K_{\mathcal{Q},j} c_j^+$ yields Equation (A.7d). With H empty, terms involving H vanish: $z_{a,j} = y_j - b_a\left(x_j\right)$ and $A_{h,j} = K_{\mathcal{Q},j} / \left(K_{j,j} + \alpha\right)$. This identity compares two applications of Equation (3.3). Each application still solves for the correction using residuals from the base predictions; the identity does not replace them with residuals from previously corrected predictions.

Checking whether the other predictions improved. We compare prediction error on $\mathcal{Q}$ before and after adding the measurement. Let $\eta_a(h) = y_{\mathcal{Q}} - f_a(h)$ be the signed error before the update. This uses the withheld responses, unlike the prepared absolute-error estimates $e_a(x)$. After the update, the error is $\eta_a(h) - \Delta_{a,j}(h)$. Subtracting the new absolute error from the old one gives the reduction at each case and its mean:

$$\begin{aligned} u_{a,j}(h) &= \left|\eta_a(h)\right| - \left|\eta_a(h) - \Delta_{a,j}(h)\right|, \\ \overline{u}_{a,j}(h) &= \tfrac{1}{m} \mathbf{1}_m^\top u_{a,j}(h) \\ &= L_{\mathcal{Q}}\left(f_a(h); y_{\mathcal{Q}}\right) - L_{\mathcal{Q}}\left(f_a\left(h^+\right); y_{\mathcal{Q}}\right). \end{aligned} \tag{A.7e}$$

Absolute values are taken separately at each case. A positive $\overline{u}_{a,j}(h)$ means that the mean absolute error fell; a negative value means that it rose. This comparison measures whether the adjustments brought the predictions closer to the corresponding withheld responses.

The coefficients A_h are available before the remaining responses are revealed. The actual residual and the resulting prediction change become known once the candidate's response is obtained. Checking the error reduction additionally requires the withheld responses in $\mathcal{Q}$. The all-candidate diagnostics therefore use these withheld values retrospectively; they do not supply them to the acquisition policy.

For the current-option comparison, we set $a = \widehat{a}(h)$ and hold it fixed. The deployed procedure can select another option after the reveal, so its full change in F can also include reselection. The calculation here isolates one correction, before acquisition cost or a second measurement is included. Section 4.5 uses it to analyze the procedure; acquisition decisions still use Section 3.6.

A.3 Joint Outcome Model and Conditioning

A.3.1 Prepared episode model

Let $\mathcal{I} = (\mathcal{P}, \mathcal{Q})$ contain the $N = n + m$ episode cases, with purchasable cases first. For each prepared prediction option r, let $B_r \in \mathbb{R}^N$ contain its predictions and $E_r \in \mathbb{R}^N_+$ its nonnegative case-specific error estimates. Thus $B, E \in \mathbb{R}^{R\times N}$, with options as rows; this is the transpose convention of the option-column matrices B_H and $B_{\mathcal{Q}}$ in A.2.2. There is one outcome component per option, in the same fixed order: $R = M = 10, N = 30$ in ChEMBL and $R = M = 5, N = 84$ in HELP. Consequently, $B_{ri} = b_r(x_i)$ and $E_{ri} = e_r(x_i)$; the component and option indices refer to the same bank but have distinct roles. The joint response model is

$$\widehat{P}_0(Y_{\mathcal{I}}) = \sum_{r=1}^{R} w_r^0 \mathcal{N}_N\left(\mu_r^0, \Sigma_r^0\right), \qquad \sum_r w_r^0 = 1. \tag{A.8}$$

Component means and covariances. For each component, a fitted scalar bias β_r is added to every prediction. A fitted positive multiplier κ_r scales its error estimates. The bias has the observed-minus-base-prediction sign. Define

$$\bar{E}_{ri} = \max(E_{ri}, 0.02), \qquad D_r = diag\left(\kappa_r \bar{E}_{r1}, \ldots, \kappa_r \bar{E}_{rN}\right).$$

The prepared moments are

$$\mu_r^0 = B_r + \beta_r \mathbf{1}, \qquad \Sigma_r^0 = \tau^2 \mathbf{1}\mathbf{1}^\top + \rho D_r K D_r + (1-\rho) D_r^2 + \left(\sigma^2 + 10^{-7}\right) I. \tag{A.8a}$$

Here K is the domain kernel in A.2.3. The first covariance term represents a shared environment offset. The next two terms represent correlated and independent case-specific variation, respectively. The fitted floor $\sigma^2 I$ and fixed numerical term $10^{-7} I$ complete the covariance. The scalars τ, ρ, σ are shared across components within a fitted model; β_r and κ_r are component-specific. The component describes the responses themselves, so conditioning does not add another observation-noise term. The correction coefficient α is separate from these covariance parameters. The error threshold 0.02 is in internal response units; the fixed numerical variance 10^{-7} is in squared internal response units.

Initial component weights. The weights use the unfloored initial error estimates on the prediction set:

$$e_r^0 = \frac{1}{m}\sum_{i\in\mathcal{Q}} E_{ri}, \qquad z_r = -\frac{e_r^0 - \min_s e_s^0}{\max(\vartheta, 10^{-5})}, \qquad w_r^0 = \frac{\exp\{clip(z_r - \max_s z_s, -60, 0)\}}{\sum_t \exp\{clip(z_t - \max_s z_s, -60, 0)\}}. \tag{A.8b}$$

The temperature ϑ is fitted during preparation and has the same units as the error estimates. The clipped logits give positive initial weights. The error floor in Equation (A.8a) does not alter either

these weight inputs or the initial option-selection scores. The weights describe the joint response model, rather than weights for averaging the final corrected predictions. Likewise, the fitted bias in μ_r^0 does not replace the base predictions in the selection-and-correction rule.

Preparation fit. Let j index the development episodes retained for a given fit. All responses in both the acquisition pool and prediction set of these model-preparation episodes enter the fit. These episodes come only from the development data permitted for that fit; its held-out validation or test tasks are excluded. Within each protein or device, episode weights sum to one: $\omega_j = 1/c_{t(j)}$, where $c_{t(j)}$ is its number of retained episodes. These weights are not further normalized across proteins. Their total therefore equals the number of fitting proteins or devices, not one; this convention fixes the scale of the data term relative to the 0.05 penalty below.

Step 1: bias. The bias is first computed as the weighted mean observed-minus-predicted response:

$$\beta_r = \frac{\sum_j \omega_j \, N^{-1} \sum_{i=1}^{N} (y_{ji} - B_{jri})}{\sum_j \omega_j}, \qquad v_{jr} = y_j - B_{jr} - \beta_r \mathbf{1}. \tag{A.8c}$$

The denominator is numerically protected by a 10^{-10} floor, which is inactive for these nonempty development-fitting sets.

Step 2: covariance. With the biases fixed, we fit the covariance parameters by minimizing

$$\begin{aligned} J(\theta) \quad = \sum_j \tfrac{\omega_j}{R} \sum_{r=1}^{R} \tfrac{1}{2} \left[N \log(2\pi) + \log \det \Sigma_{jr}^0(\theta) + v_{jr}^\top \Sigma_{jr}^0(\theta)^{-1} v_{jr} \right] \\ +0.05 \sum_r \left(\log \kappa_r - \tfrac{1}{R} \sum_s \log \kappa_s \right)^2, \end{aligned} \tag{A.8d}$$

where $\theta = (\log \kappa_1, \ldots, \log \kappa_R, \log \tau, \text{logit}\rho, \log \sigma)$. Thus covariance fitting averages the component Gaussian negative log densities; it does not minimize the negative log density of their mixture. Biases are fixed at Step 1, and the initial-weight temperature is not a variable in this optimization. Positive parameters use the exponential transform, and ρ uses the logistic transform.

We use the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm with bound constraints (L-BFGS-B; Byrd et al., 1995), with bounds $[-2, 2]$ on each $\log \kappa_r$, $[-4, 1]$ on $\log \tau$, $[-4, 4]$ on $\text{logit}\rho$, and $[-4.5, 0]$ on $\log \sigma$. The initial values are $\tau = 0.35, \rho = 0.5, \sigma = 0.08$, and

$$\kappa_r^{\text{init}} = clip_{[0.25,4]} \sqrt{\frac{\sum_j \omega_j \, N^{-1} \|v_{jr}\|_2^2}{\max\{\sum_j \omega_j \, N^{-1} \|\bar{E}_{jr}\|_2^2, 10^{-10}\}}}. \tag{A.8e}$$

The optimizer uses numerical gradients, at most 120 iterations, relative objective tolerance 10^{-8}, and at most 30 line-search steps per iteration. Gaussian log densities are evaluated by Cholesky factorization; a failed factorization during fitting receives objective value 10^{100}.

Step 3: initial-weight temperature. Holding the fitted biases and covariances fixed, we choose the temperature on the same development-fitting episodes:

$$\vartheta^* = \arg \max_{\vartheta \in \{0.05, 0.1, 0.2, 0.4, 0.8, 1.6\}} \sum_j \omega_j \log \sum_{r=1}^{R} w_{jr}^0(\vartheta) \, \mathcal{N}_N \left(v_{jr}; 0, \Sigma_{jr}^0 \right). \tag{A.8f}$$

This score is evaluated by log-sum-exp, with a 10^{-300} floor before taking the logarithm of a weight; exact score ties select the smaller temperature. This final step uses the mixture density, unlike Step 2. It uses the same fitting episodes rather than a separate temperature-validation holdout. The bias, covariance parameters, and initial-weight temperature are therefore fitted in sequence.

Fitting populations. In ChEMBL, tasks are sorted by protein, environment, and assignment, retaining one deterministic assignment per environment. Of the 513 development tasks, the final fit uses 171 tasks, one from each environment of the 57 development proteins, with 30 response entries per task and weight $1/3$ per environment. The 15 test proteins are excluded from this fit.

In HELP, each device–assignment pair defines a distinct task, so all three tasks per development device are retained. Each development-validation fit uses five devices and 15 tasks; the final fit uses all six development devices and 18 tasks. Every task contains 42 acquisition cases and 42 prediction cases, giving 84 response entries and weight $1/3$ per task. The error estimates supplied to a fitting episode exclude that episode's own device, as specified in A.1.2.

The four test devices share the same final HELP preparation parameters and selection/correction settings. Each test episode assembles its outcome model from those parameters and its prepared predictions, error estimates, and kernel. Acquired responses condition only that episode's model; they do not refit the shared preparation parameters or carry over to another test episode.

These fits supply the complete tuple in Equation (A.8) before the evaluated environment is queried. Acquiring a response conditions this fixed model; it does not refit its preparation parameters or the source forests.

A.3.2 Conditioning a component on the acquired responses

Partition each component into acquired indices H and remaining indices $U = \mathcal{I}\backslash H$. The following formulas are the exact Gaussian-conditioning algebra for the declared component, with an invertible observed block. Any model-defined observation variance is part of that component's response covariance.

$$\mu_{r,U}(h) = \mu^0_{r,U} + \Sigma^0_{r,UH}\left(\Sigma^0_{r,HH}\right)^{-1}\left(y_H - \mu^0_{r,H}\right), \tag{A.9}$$

$$C_{r,UU}(h) = \Sigma^0_{r,UU} - \Sigma^0_{r,UH}\left(\Sigma^0_{r,HH}\right)^{-1}\Sigma^0_{r,HU}. \tag{A.10}$$

The weight becomes

$$w_r(h) = \frac{w^0_r\,\mathcal{N}_k\left(y_H; \mu^0_{r,H}, \Sigma^0_{r,HH}\right)}{\sum_{s=1}^{R} w^0_s\,\mathcal{N}_k\left(y_H; \mu^0_{s,H}, \Sigma^0_{s,HH}\right)}. \tag{A.11}$$

The implementation factors the prepared observed block $\Sigma^0_{r,HH}$ by Cholesky, without adding further observation variance or jitter. It computes the Schur complement over the full episode vector, replaces it by its symmetric part, and clips negative diagonal entries to zero. Off-diagonal entries are not clipped, and there is no eigenvalue projection or positive conditional-variance floor. The unacquired block of this matrix supplies the covariances used in Equation (3.4).

Posterior weights are evaluated in log space using log-sum-exp (LSE):

$$\begin{aligned} \ell_r &= \log\max\left(w^0_r, 10^{-300}\right) \\ &\quad -\tfrac{1}{2}\left[k\log(2\pi) + \log\det\Sigma^0_{r,HH} + \left(y_H - \mu^0_{r,H}\right)^\top\left(\Sigma^0_{r,HH}\right)^{-1}\left(y_H - \mu^0_{r,H}\right)\right], \\ w_r(h) &= \exp\{\ell_r - \mathrm{LSE}\left(\ell_1, \ldots, \ell_R\right)\}. \end{aligned}$$

Here $\mathrm{LSE}\left(u_1, \ldots, u_R\right) = \log\sum_r \exp\left(u_r\right)$. The log determinant is twice the sum of the logarithms of the Cholesky diagonal entries. There is no subsequent positive floor on posterior weights, so numerical underflow can produce a zero weight. The prepared-prior floor is inactive for the positive clipped-softmax weights in Equation (A.8b). For an empty history, the likelihood contribution is omitted and the prepared moments are returned.

Every actual or hypothetical history is evaluated jointly from the prepared tuple. At fixed acquired indices, the component covariances depend only on those indices, while the conditional means and weights also depend on the observed values.

In exact Gaussian algebra, conditioning on one further response y at q can also be written sequentially as follows, for $C_{r,qq} > 0$. These identities explain the update; the implementation instead evaluates the complete enlarged history from the prepared tuple, with the numerical treatment described above. For remaining indices $U' = U\backslash\{q\}$,

$$\mu^{+}_{r,U'} = \mu_{r,U'} + \frac{C_{r,U'q}}{C_{r,qq}}\left(y - \mu_{r,q}\right), \qquad C^{+}_{r,U'U'} = C_{r,U'U'} - \frac{C_{r,U'q}C_{r,qU'}}{C_{r,qq}}, \tag{A.12}$$

$$w^{+}_{r} = \frac{w_r\,\mathcal{N}\left(y;\mu_{r,q},C_{r,qq}\right)}{\sum_{s=1}^{R} w_s\,\mathcal{N}\left(y;\mu_{s,q},C_{s,qq}\right)}. \tag{A.13}$$

These equations condition on a revealed response variable; they do not assert that a physical measurement is noise-free. Equation (A.8a) already defines the response covariance, including its fitted and numerical diagonal terms. Those terms are not added again to the observed block, and the correction parameter α is not used as observation noise.

A.3.3 Hypothetical responses do not reveal a component identity

A quadrature node may be generated from component r's marginal response distribution, but its value is treated only as a hypothetical observation $Y_q = y$. The child state is obtained by conditioning the **whole mixture** on that value. The generating component is not supplied as an additional observation and is not forced to have posterior weight one.

This distinction matters because overlapping component distributions may assign substantial likelihood to the same value. The hypothetical response updates the likelihoods and weights of all components using the complete enlarged history, and thereby changes the next proposal and risk evaluation. The final predictor is still the selected-and-corrected F, not the mixture mean.

A.4 Expected Absolute Loss and the Posterior-Geometry Score

A.4.1 Absolute loss under a Gaussian marginal

Let $Y \sim \mathcal{N}(\mu, v)$, let z be a fixed prediction, and set $d = z - \mu$. For $v > 0$, writing ϕ and Φ for the standard normal density and cumulative distribution function,

$$\psi(d,v) = \mathbb{E}|z - Y| = 2\sqrt{v}\,\phi\left(d/\sqrt{v}\right) + d\left[2\Phi\left(d/\sqrt{v}\right) - 1\right]. \tag{A.14}$$

To obtain this expression, write $Y = \mu + \sqrt{v}Z$ and split the integral of $|d - \sqrt{v}Z|$ at $Z = d/\sqrt{v}$. The two truncated first moments are $-\phi\left(d/\sqrt{v}\right)$ and $\phi\left(d/\sqrt{v}\right)$. Their substitution yields Equation (A.14). The continuous zero-variance limit is $\psi(d,0) = |d|$. Numerically, the variance is first replaced by $\max(v,0)$. If its square root is at most 10^{-10} in internal response units, the implementation uses $|d|$; otherwise it evaluates the Gaussian formula. This standard-deviation cutoff and the variance-unit denominator floor in the posterior geometry calculation are distinct numerical rules.

Consequently, the current modeled loss is

$$\widehat{R}_F(h) = \frac{1}{m}\sum_{j\in\mathcal{Q}}\sum_{r=1}^{R} w_r(h)\,\psi\left(F_j(h) - \mu_{r,j}(h), C_{r,jj}(h)\right). \tag{A.15}$$

This evaluates the absolute error of the specified output $F(h)$ under the fitted mixture. It is not the risk of an optimal mixture-median predictor. Linearity of expectation permits the sum over marginal losses even when the responses in the prediction set are dependent. By contrast, acquisition needs the joint law because a revealed response changes those marginals.

For fixed model parameters and a fixed history, Equation (A.15) is an analytic model expectation, subject to the stated small-variance convention and floating-point evaluation. The expectations over **future acquired responses**, whose values also change the selected option and correction, are evaluated by the separate quadrature in A.5.

A.4.2 Within-component variance reduction

From Equation (A.12), revealing Y_q changes the variance of response in the prediction set Y_j within component r by

$$C_{jj}^{r,h} - C_{jj}^{r,h\oplus(q,y)} = \frac{\left(C_{jq}^{r,h}\right)^2}{C_{qq}^{r,h}}, \tag{A.16}$$

when $C_{qq}^{r,h} > 0$. This reduction does not depend on the response value within that fixed component. Averaging it over prediction-set indices and the **current** component weights gives the unregularized score in Equation (3.6). The implementation uses $\max\left(C_{qq}^{r,h}, 10^{-10}\right)$ in the denominator, with the threshold expressed in squared internal response units, and considers only unacquired candidates. Exact posterior geometry-score ties use lexical case identifiers: molecular keys in ChEMBL and zero-padded `arch_00000`-format identifiers in HELP, equivalent to increasing architecture identifier. Nearby but unequal scores are not treated as ties. The denominator floor is not a score-tie tolerance.

The mixture's full marginal variance also contains between-component mean variation:

$$\mathrm{Var}_{\widehat{P}_h}\left(Y_j\right) = \sum_r w_r(h)\left[C_{jj}^{r,h} + \left(\mu_{r,j}(h) - \overline{\mu}_j(h)\right)^2\right], \quad \overline{\mu}_j(h) = \sum_r w_r(h)\mu_{r,j}(h). \tag{A.17}$$

Equation (3.6) is not an integral of the change in this entire expression. It averages the within-component reduction while using current weights. Response-dependent changes in component weights and means are subsequently reflected in the detailed hypothetical-history evaluation, not folded into an assertion that posterior geometry is exact downstream acquisition value.

Once conditional covariances and weights are available, computing the posterior geometry scores requires no option reselection or corrected-output calculation at hypothetical responses. Those calculations are reserved for the selected proposal. This is the computational division between proposal and valuation.

A.5 Quadrature and the Optional Second Acquisition

A.5.1 Three-node integration of a response marginal

At history h, the modeled marginal response of candidate q is

$$\sum_{r=1}^{R} w_r(h)\,\mathcal{N}\left(\mu_{r,q}(h), C_{r,qq}(h)\right).$$

The three-node Gauss–Hermite rule expressed in standard-normal coordinates uses

$$(t_1, t_2, t_3) = \left(-\sqrt{3}, 0, \sqrt{3}\right), \qquad (\omega_1, \omega_2, \omega_3) = (1/6, 2/3, 1/6). \tag{A.18}$$

These are the normalized nodes and weights for a standard normal marginal, not the unscaled physicists' Hermite convention. For each component, set

$$y_{r\ell}(h,q) = \mu_{r,q}(h) + \sqrt{\max\left(C_{r,qq}(h), 0\right)}\, t_\ell, \qquad \pi_{r\ell}(h,q) = w_r(h)\omega_\ell.$$

For a function u of the response, the numerical expectation is

$$\widetilde{\mathbb{E}}_{h,q}\left[u\left(Y_q\right)\right] = \sum_{r=1}^{R}\sum_{\ell=1}^{3} \pi_{r\ell}(h,q)\, u\left(y_{r\ell}(h,q)\right). \tag{A.19}$$

The implementation retains all $3R$ mixture atoms, including coincident response values and zero-mass atoms, sorts them stably by response value, and normalizes their masses to sum to one before the numerical expectation. Equation (A.19) uses those normalized masses; in exact arithmetic, the

displayed $w_r(h)\omega_\ell$ already sum to one. No atom compression is applied in the retained primary calculation. There are 30 first-response evaluations in ChEMBL and 15 in HELP, with the same construction at the second step. Every atom conditions the whole mixture on the enlarged history as in A.3.3; its generating component is not treated as observed. These are computational evaluations, not acquired response labels.

Selection boundaries and the minimum between stopping and continuing make the integrand nonlinear and potentially nonsmooth. The three-node calculation is therefore a numerical approximation to the fitted-model expectation, not an exact integral of all continuous response values.

A.5.2 Matching the second-stage decision to the first-stage score

Define the one-further-measurement loss for an arbitrary available proposal q by

$$\widehat{A}_1(h, q; \lambda) = \lambda + \widetilde{\mathbb{E}}_{h,q}\left[\widehat{R}_F\left(h \oplus (q, Y_q)\right)\right]. \tag{A.20}$$

Let the remaining one-step continuation be

$$B_1(h; \lambda) = \begin{cases} \widehat{R}_F(h), & \mathcal{P}\backslash H(h) = \varnothing, \\ \min\{\widehat{R}_F(h), \widehat{A}_1\left(h, q_h; \lambda\right)\}, & \text{otherwise.} \end{cases} \tag{A.21}$$

The implemented two-step continuation loss, including acquisition costs, is then

$$\widehat{A}_2(h; \lambda) = \lambda + \widetilde{\mathbb{E}}_{h,q_h}\left[B_1\left(h \oplus (q_h, Y_q)\,; \lambda\right)\right]. \tag{A.22}$$

This is Equation (3.7) with the empty-pool case included. When only one candidate remains, it reduces to the one-further-measurement loss in Equation (A.20), evaluated at the sole remaining proposal.

At each first-response node, the internal second-acquisition decision is the strict comparison

$$\widehat{A}_1\left(h_1, q_{h_1}; \lambda\right) < \widehat{R}_F\left(h_1\right). \tag{A.23}$$

An exact tie chooses STOP. The outer, actually executed acquisition uses the separate 10^{-10} tolerance in internal loss units. ChEMBL implements it as $\widehat{R}_F(h) > \widehat{A}_2(h; \lambda) + 10^{-10}$, whereas HELP implements it as $\widehat{A}_2(h; \lambda) < \widehat{R}_F(h) - 10^{-10}$, with $\widehat{A}_1$ replacing $\widehat{A}_2$ when only one candidate remains. These are algebraically the same rule, but their floating-point operation order differs. Scores, risks, and prices are not rounded before comparison. Algorithm 1 gives the common mathematical rule rather than prescribing either operation order.

Applying this outer tolerance to the internal comparison would define a slightly different continuation; the continuation loss uses the minimum in Equation (A.21).

The expected second cost is incurred only on branches where the second acquisition is chosen. In the quadrature sum, total acquisition cost is thus λ plus λ times the weighted fraction of continuing branches, not invariably 2λ.

A.5.3 The value of considering an optional second measurement

For the same first proposal and the same quadrature rule, algebra gives

$$\begin{aligned} &\widehat{A}_1\left(h, q_h; \lambda\right) - \widehat{A}_2(h; \lambda) \\ &= \widetilde{\mathbb{E}}_{h,q_h}\left[\left(\widehat{R}_F\left(h_1\right) - \widehat{A}_1\left(h_1, q_{h_1}; \lambda\right)\right)_+\right] \\ &\geq 0, \end{aligned} \tag{A.24}$$

with the integrand set to zero when no second candidate remains. The identity follows from $x - \min(x, y) = (x - y)_+$ and the nonnegative quadrature weights. It describes the value **inside the**

calculation of being allowed to acquire again without being required to do so. It does not assert that the second acquisition improves realized loss in every episode, or that the complete two-step-replanning policy dominates a separately executed one-step policy.

The expectation remains outside the minimum over the two first-response-dependent choices. Moving that minimum outside the expectation would remove the ability to change the second decision after seeing the first response.

A.5.4 Replanning and the estimand used in value diagnostics

The runtime controller acquires at most one response before replanning. A second acquisition anticipated by $B_1(h_1; \lambda)$ is therefore not a commitment made at the previous step. Once an actual first response arrives, the runtime controller can again look two steps ahead.

For a diagnostic of Equation (A.22), however, the matching realized continuation is different: acquire the proposal, apply the **one-step remaining** comparison in Equation (A.23) using its actual response, acquire a second response only if that rule elects to do so, and then stop. Its endpoint h_{end} is fixed by model calculations before responses in the prediction set are used for scoring. The corresponding realized net gain is

$$L_{\mathcal{Q}}(F(h); y_{\mathcal{Q}}) - L_{\mathcal{Q}}(F(h_{\text{end}}); y_{\mathcal{Q}}) - \lambda(|H(h_{\text{end}})| - |H(h)|). \tag{A.25}$$

This evaluates the same local continuation as the modeled two-step continuation loss. It is not the terminal loss of the full runtime policy after unrestricted repeated replanning. In the retrospective diagnostic, the same construction can also evaluate a proposed continuation rejected at a voluntary STOP. A pool-exhausted state has no such next proposal and is not assigned an artificial zero-gain measurement.

A.5.5 Complete acquisition procedure

An episode is one complete acquisition-and-prediction run. At step t, we measure case q_t from the acquisition pool and obtain response y_{q_t}. The measurement history h_t records these case–response pairs, while H_t contains their case identities:

$$h_t = ((q_1, y_{q_1}), \ldots, (q_t, y_{q_t})), \qquad H_t = \{q_1, \ldots, q_t\} \subseteq \mathcal{P}.$$

Algorithm 1. Selecting, correcting, and acquiring measurements

```
Inputs: acquisition pool P; prediction set Q; visible descriptors;
        prepared predictions b and initial errors e0; correction kernel
        ↪ K;
        fitted outcome model; correction/selection settings alpha and s;
        per-response acquisition penalty lambda;
        numerical stopping tolerance epsilon_stop = 1e-10.

h <- empty history
repeat:
    Select an option and correct its outputs to obtain F(h).
    Condition the outcome model on h.
    stop_loss <- modeled expected loss of F(h).
    If every candidate in P has been acquired:
        return F(h), POOL_EXHAUSTED.
    Let q be the remaining candidate with the largest posterior geometry
    ↪ score.
    If q is the last remaining candidate:
        continuation_loss <- A1(h; lambda).
    Otherwise:
        for each quadrature response of q:
            update the hypothetical history and the whole outcome
            ↪ mixture;
            recompute F and the next posterior geometry proposal;
```

```
        compare stopping there with acquiring that proposal once
        ↪ more;
    continuation_loss <- A2(h; lambda), including acquisition costs.
estimated_net_gain <- stop_loss - continuation_loss.
If estimated_net_gain <= epsilon_stop:
    return F(h), VOLUNTARY_STOP.
Reveal the actual response of q and append it to h.
```

A.6 Online Computation, Caching, and Response Units

A.6.1 Fixed objects and history-dependent calculations

The source prediction bank, initial error estimates, relationship kernel, fitted outcome-model parameters, correction and selection settings, and acquisition price are fixed during an episode. Acquired values change the selection scores, residual corrections, and component means and weights. The acquired indices determine the component conditional covariances. Together these quantities determine the proposal and expected continuation loss.

Both implementations retain an episode-local dictionary of location-dependent Gaussian conditioning terms. Its keys are ordered tuples of acquired indices, without sorting the tuple, and each entry contains the component transfer matrices, full conditional covariances, Cholesky factors, and log determinants. These entries are shared across response values with the same ordered indices. There is no size limit or eviction rule; the entries remain until the episode world is discarded.

ChEMBL recomputes its small correction solve at every nonempty terminal evaluation, sharing the system across options. HELP additionally batches hypothetical responses with common acquired indices and shares a kernel transfer solve to the prediction set within each batch and correction setting. First-response branches are grouped by their next proposal to form the second-stage batches. Neither implementation caches KRR factorizations across calls.

A.6.2 Algebraic work, rather than a timing claim

For dense matrices and prepared kernel blocks, evaluating all corrections with a common $k \times k$ solve has a direct arithmetic cost of order

$$O\left(k^3 + Mk^2 + mkM\right). \tag{A.26}$$

This expression counts factorization, multiple right-hand-side solves, and correction on the prediction set. It is an algebraic description of the direct scalar calculation in Equation (A.6), not a measured speed comparison or a claim about a particular solver backend.

For a batch of B_{batch} response vectors with common acquired indices, HELP computes T_H^{corr} once per correction setting and applies it to all $B_{\text{batch}}M$ residual vectors. The direct dense correction cost is

$$O\left(k^3 + k^2 m + B_{\text{batch}} M k m\right).$$

This changes the organization of numerical work without changing the prediction formula. Once component conditional moments are available, evaluating $\widehat{R}_F(h)$ uses Rm marginal absolute-loss terms, while posterior geometry over $n-k$ candidates uses $Rm(n-k)$ covariance-reduction terms. Preparing and conditioning the component covariances requires additional computation and is not hidden inside the acquisition penalty.

Let $B = 3R$ denote the number of first-response quadrature evaluations. A two-step valuation can evaluate B first-child predictions and up to B^2 second-child predictions, in addition to the current prediction. This gives 30 and up to 900 branch evaluations in ChEMBL, or 15 and up to 225 in HELP, before any reuse of equivalent numerical work. No source RF is fitted at these branches. The actual correction dimension is at most 15 in the ChEMBL episodes and 42 in the HELP episodes.

The construction reduces repeated fitting and confines detailed outcome evaluation to the recommended candidate. It still spends computation to decide whether to acquire information. Prepara-

tion work, online computation, and acquired label count must therefore remain separately reported resources.

A.6.3 SHARED POSTERIOR GEOMETRY PATHS ACROSS PRICES

For a fixed outcome model and an acquired history, the posterior geometry proposal does not use λ or the terminal selection/correction settings. A retrospective evaluator can consequently reuse the same potential sequential posterior geometry path across acquisition prices. At each price, it chooses the **first** state at which the estimated net gain, given by modeled stopping loss minus continuation loss, fails the acquisition comparison. Later response values or realized prediction-set losses are not used to move that stopping point.

A stored path may continue beyond a particular policy's STOP because it supports other prices and post hoc diagnostics. That storage does not give the earlier decision access to later labels. The operational policy remains the one-response-at-a-time procedure in Algorithm 1.

A.6.4 HELP RESPONSE AND PRICE UNITS

HELP responses use natural-log latency normalized by constants computed from the five source devices alone. With latency t in the reported native unit, the internal response is

$$y = \frac{\log t - c_{\text{src}}}{s_{\text{src}}}, \quad c_{\text{src}} = 2.412847587151337, \quad s_{\text{src}} = 0.6103144578607641. \tag{A.27}$$

The center and scale are the mean and population standard deviation (`ddof=0`) of the natural-log latencies pooled across all architectures and the five source devices. No clipping or scale floor is applied; responses are checked to be finite and positive before transformation, and the source scale is checked to be positive. The source predictors are fitted in these normalized response units. The current device's unacquired extrema or full latency profile are not used to compute these constants. Internal MAE, modeled loss, and acquisition cost are expressed on this same standardized scale. Conversion to natural-log-latency error uses

$$L_{\log} = s_{\text{src}} L_{\text{internal}}, \qquad \lambda_{\log} = s_{\text{src}} \lambda_{\text{internal}}. \tag{A.28}$$

Multiplying both loss and cost by the same positive scale preserves the underlying exact comparison. The outer tolerance is specified in internal units; its corresponding log-latency value is $6.103144578607641 \times 10^{-11}$. Biases, predicted errors, τ, σ, and the initial-weight temperature are in internal response units. The covariance's 10^{-7} numerical term and posterior geometry's 10^{-10} denominator floor are in squared internal response units. The scale multipliers κ_r, correlation parameter ρ, and dimensionless-kernel correction parameter α have separate roles. Conversion to millisecond prediction error is nonlinear and is not obtained by multiplying log-MAE by the source scale.

The predicted latency, the computation time required for an acquisition decision, and the time required to obtain a latency label are distinct quantities.

B DATA AND EVALUATION PROTOCOLS

Both experiments ask how to obtain predictions in a new environment while paying for only some of its measurements. This appendix explains how the data are divided, how the predictors and settings are prepared, and how the resulting predictions and acquisition decisions are evaluated. Sections B.1–B.2 describe the two applications; B.3 defines settings and comparisons; B.4 covers metrics and uncertainty; B.5 covers computation time; and B.6 specifies the explanatory analyses.

B.1 ChEMBL: molecular binding measurements

B.1.1 Responses and eligible environments

Source data supply the starting predictors. Development data are used to estimate how they behave in other environments, fit a distribution over possible unobserved responses (the outcome model), and choose settings. Testing then applies these fixed preparation choices to separate proteins or devices.

In ChEMBL, a case is a molecule and an environment is a binding-assay setting for a particular protein. An assay is an experimental procedure used to measure binding. We use version 37 of the ChEMBL database (Gaulton et al., 2012), retaining numerical inhibition constants, K_i, from eligible human single-protein binding assays. The response is pKi: the negative base-10 logarithm of K_i expressed in molar units. Larger pKi therefore represents stronger binding.

The database's log-affinity field, pChEMBL, supplies the response for this exact-Ki subset. A record must agree with 9 - $\log_{10}(K_i$ in nM) to within 0.02. This check verifies consistency between two database fields; it does not standardize responses across assays. The remaining eligibility rules are summarized in Table B.1.

Table B.1. Eligibility rules for the ChEMBL study.

Object	Retained records
Protein	Human (Homo sapiens; taxonomic identifier 9606), annotated as a single protein rather than a species group; exactly one protein component, with a matching accession and a human sequence.
Assay assignment	Binding assay (assay_type B), direct target relationship (D), confidence score 9, human organism annotation, source identifier 1, and an associated publication (the paper reporting the measurements).
Assay format	BioAssay Ontology (BAO) formats: single-protein (BAO_0000357) or cell-based (BAO_0000219). Variant-sequence and tissue fields must be absent.
Measurement	Standard type Ki, equality relation, nM units, standard flag 1, positive finite standard value and finite pChEMBL value; no potential-duplicate flag, validity comment, variant accession or mutation.
Description	Evidence of binding or Ki is required. Descriptions indicating variants, ambiguous target identities or functional rather than binding contexts are excluded by the fixed text-screening rules.
Molecular structure	A canonical structure encoded in the Simplified Molecular Input Line Entry System (SMILES; Weininger, 1988) must be present and yield a valid standardized structure. Repeated records are grouped using the standardized molecular identifier, as described below.

For each protein, we group eligible assays by their source publication and select the assay containing the largest number of distinct standardized molecules; ties are resolved by assay identifier in lexical order. It defines one candidate environment. A selected environment must contain at least 30 distinct molecules: 15 for the acquisition pool and 15 for the separate prediction set.

B.1.2 Molecular identity and source measurements

Molecular identity is standardized before constructing tasks or source subsets. We apply RDKit ChargeParent standardization and generate canonical isomeric SMILES, a textual representation of molecular structure that retains stereochemistry. Records are grouped by the resulting InChIKey, a standardized molecular identifier. Within an assay, repeated measurements of the same standardized molecule are replaced by their median pKi. Measurements of that molecule in different assays remain separate source rows.

For each selected environment, the source data used to train the starting predictors consist of eligible measurements for the same protein in other publications. We exclude the current publication, every molecule appearing anywhere in the current representative assay, and publications reserved from source use by the fixed study partition. The molecule exclusion applies to the entire representative assay, not just to the 15 candidate measurements and 15 prediction cases assigned to a task. We re-

quire a minimum source-data size and coverage across assays and publications: at least 200 distinct molecules, three assays and three publications after the exclusions.

For a test protein, its source measurements concern that test protein. They are not the measurements used to form development tasks for the 57 preparation proteins. Development data instead determine how the prepared predictions will be selected, corrected and used to assess further measurements.

B.1.3 Preparing the ten prediction options

To construct comparable choices from the source data, we group source rows with exactly the same assay description. Each group may include more than one assay or publication. We rank these whole groups by how closely their molecular-structure distributions resemble that of the current representative assay, using balanced optimal transport: a distributional comparison that matches their molecular structures while preserving the weight of each collection. The ground cost between two molecules is one minus their binary-fingerprint Tanimoto similarity. Each source model row has equal mass, so repeated molecules in different assay rows retain their multiplicity.

Starting with the closest group, we add complete groups until reaching each nominal fraction from 10% to 100% of the available source model rows, in steps of 10%. Because a group is not split, a subset may exceed its nominal fraction. Distances within 10^{-12} of a group minimum are treated as ties and ordered by the source-group identifier. Ranking uses structures from the full current representative assay, not its response values.

The forests use molecular fingerprints: numerical vectors representing molecular structure. The activity model uses counts, while the similarity calculations use binary indicators of the represented features (Table B.2).

Table B.2. ChEMBL source-prediction and initial-error preparation.

Setting	Specification
Source predictor	Random forest regression; 300 trees; square-root feature subsampling at a split; squared-error criterion; bootstrap sampling; unrestricted depth; minimum leaf size 1 and minimum split size 2.
Molecular input	Unscaled chiral Morgan count fingerprints, radius 2 and 2,048 components; sparse float32 representation.
Error regressor	Random forest regression with 300 trees and all three error-prediction features available at each split; otherwise the forest settings above.
Error-prediction features	Population standard deviation across source trees, their mean predicted pKi, and maximum binary-Morgan Tanimoto similarity to a unique source training molecule.
Error training targets	Absolute prediction errors obtained from five shuffled folds that keep all measurements of the same standardized molecule together.
Correction kernel	Chiral binary Morgan fingerprints, radius 2 and 2,048 bits; Tanimoto similarity, symmetrized with diagonal one. This differs from the count-fingerprint input to the source forest.

The error regressor is trained on predictions for molecules withheld from the source forest used to predict them. All source measurements of a withheld molecule are excluded together, including measurements in other assays. After source preparation, the error regressor supplies one initial absolute-error estimate for each prediction option and current case. These estimates remain fixed during acquisition; the selection score combines them with errors revealed by purchased responses. Appendix A.1.1 gives the construction, and Section 3.3 gives the selection and correction rules.

B.1.4 Protein groups and prediction tasks

The 57 development proteins each contribute three selected environments. The 15 test proteins each contribute one. For every selected environment, three predefined assignments of molecules to measurement and prediction give three separate tasks. Thus the development group contains 171 environments and 513 tasks, and the test group contains 15 environments and 45 tasks. The protein groups are disjoint; their identifiers are given in Table B.3.

Table B.3. Fixed ChEMBL protein groups.

Group	ChEMBL protein identifiers
Development: 57 proteins	CHEMBL1800, CHEMBL1833, CHEMBL1855, CHEMBL1871, CHEMBL1889, CHEMBL1899, CHEMBL1946, CHEMBL2001, CHEMBL2014, CHEMBL2034, CHEMBL2035, CHEMBL204, CHEMBL205, CHEMBL2056, CHEMBL210, CHEMBL211, CHEMBL214, CHEMBL216, CHEMBL217, CHEMBL218, CHEMBL219, CHEMBL222, CHEMBL223, CHEMBL224, CHEMBL225, CHEMBL226, CHEMBL228, CHEMBL231, CHEMBL2327, CHEMBL233, CHEMBL234, CHEMBL236, CHEMBL237, CHEMBL238, CHEMBL240, CHEMBL244, CHEMBL245, CHEMBL2492, CHEMBL251, CHEMBL253, CHEMBL255, CHEMBL256, CHEMBL259, CHEMBL261, CHEMBL264, CHEMBL287, CHEMBL3155, CHEMBL3227, CHEMBL3242, CHEMBL3371, CHEMBL344, CHEMBL3473, CHEMBL3594, CHEMBL3759, CHEMBL3798, CHEMBL4308, CHEMBL5071
Test: 15 proteins	CHEMBL1821, CHEMBL1836, CHEMBL1867, CHEMBL1875, CHEMBL1898, CHEMBL1945, CHEMBL2049, CHEMBL229, CHEMBL249, CHEMBL3764, CHEMBL4315, CHEMBL4561, CHEMBL4644, CHEMBL4777, CHEMBL4792

Within each environment, we hash an assignment-specific fixed string together with the environment identifier and each molecule's InChIKey. The resulting Secure Hash Algorithm 256-bit (SHA-256) hash values define a deterministic molecular order, with lexical InChIKey tie-breaking. We take the first 30 molecules: positions 1–15 form the prediction set Q, and positions 16–30 form the acquisition pool P. The three predefined strings give the three assignments. The same molecular identities can recur between tasks, but P and Q do not overlap within a task.

The acquisition pool P contains molecules whose responses (pKi values) the procedure may request. The prediction set Q contains molecules whose pKi values it must predict. Molecular descriptors are visible in both sets, while current-environment responses begin hidden. A task starts with no acquired responses, and every requested P response is charged once. The Q responses are used only for evaluation after the predictions and decisions have been made.

The three assignments provide repeated evaluations of a protein's environment. Aggregation first keeps these related tasks together, as specified in B.4.

B.2 HELP: HARDWARE LATENCY PREDICTION

B.2.1 DEVICE GROUPS AND ARCHITECTURE REPRESENTATION

In the hardware experiment, a case is a neural architecture and an environment is a device. The response is the architecture's execution latency on that device. We use the latency data for the neural architecture search (NAS) benchmark NAS-Bench-201 (Dong and Yang, 2020) distributed with the HELP latency-prediction study (Lee et al., 2021). Source measurements cover 15,625 architectures. The source, development and test devices have fixed, separate roles (Table B.4).

Table B.4. HELP device groups and their roles.

Group	Device identifiers	Use
Source: 5 devices	1080ti_1, titanx_1, silver_4114, essential_ph_1, pixel3	One prepared random forest predictor per device; source-only response normalization.
Development: 6 devices	2080ti_1, titanxp_1, silver_4210r, gold_6240, samsung_a50, samsung_s7	Initial error estimates, outcome-model fitting and selection of settings and comparator counts.
Test: 4 devices	titan_rtx_1, gold_6226, pixel2, raspi4	Evaluation after the preparation choices are fixed.

Our source predictors use 30 binary operation indicators: five indicators at each of the six edge positions in the architecture's cell. An indicator records which operation occupies an edge. Each source device supplies one random forest with 128 trees, a minimum leaf size of two and all 30 indicators available at each split. The five source forests are fixed during acquisition.

Latency is first expressed on a natural-log scale. We then center and scale it using only the five source-device response profiles. If t is a latency in milliseconds, the internal response is

$$y_{internal} = \frac{ln(t) - \mu_{src}}{s_{src}}, \quad \mu_{src} = 2.412847587151337, \quad s_{src} = 0.6103144578607641. \text{ (B.1)}$$

The two constants are the mean and population standard deviation of the source log latencies. They remain the same for development and test devices. An internal prediction is returned to milliseconds by reversing the scaling and exponentiating. Mean absolute error (MAE) averages the absolute prediction–response differences. On the natural-log-latency scale it is called log-MAE. Converting an internal MAE to log-MAE requires multiplication by the source scale but no addition of the source mean.

B.2.2 Relationships used for correction

The correction kernel compares the operations at corresponding edges. Let d be the number of edge positions with different operations. Its base similarity is 0.5 + 0.5 exp(-d/2). We also record whether at least one live path connects the cell input to its output; a "none" operation is an absent edge. Base similarity is unchanged when two architectures have the same connectivity status and is multiplied by 0.5 otherwise. Appendix A.2.3 gives the full kernel and identifies the covariance terms to which this factor applies.

B.2.3 Common measurement and prediction cases

Every development and test device is evaluated on the same three architecture assignments. Each task contains 42 architectures available for measurement and a separate set of 42 architectures whose latencies are to be predicted. The measurement pool includes the ten reference architectures required by the HELP comparator. Our procedure may acquire any of the 42 candidates and is not required to start with those references.

The assignments are fixed without using current-device latencies. A fixed random permutation of the 15,625 architecture indices defines the assignments. Three consecutive blocks of 84 provide the initial assignments. Their final 42 entries form each prediction set. The measurement set starts with the ten reference indices and is filled with 32 distinct non-reference candidates in the predefined order, excluding prediction cases. Any reference collision in a prediction set is replaced by the first available index from the same fixed permutation; none occurs in the assignments used here.

The ten reference indices, in order, are 11982, 431, 55, 9, 14451, 8636, 9881, 13479, 6196, 1462. HELP uses these ten reference latencies to describe the device to its predictor and also includes them among the measurements used to adjust its predictions. Each reference response is counted once, even though it serves both purposes. Every task starts afresh; responses are not transferred between its three repeated assignments.

Table B.5. Exact HELP architecture assignments.

Assignment	Candidate measurements: 42 indices	Prediction cases: 42 indices
0	11982, 431, 55, 9, 14451, 8636, 9881, 13479, 6196, 1462, 4105, 14034, 11968, 14996, 2052, 12940, 1175, 6639, 11288, 7179, 2767, 7326, 10083, 7334, 2048, 12027, 13015, 3722, 14729, 3794, 8769, 7583, 11252, 11422, 5149, 938, 5674, 10428, 4867, 9466, 4123, 15012	12542, 5036, 11045, 11569, 11457, 6006, 5797, 2678, 11681, 13870, 15098, 1668, 6442, 9796, 1429, 14561, 13726, 8420, 3726, 11541, 8918, 5812, 2330, 6951, 10629, 5714, 3039, 4670, 5458, 3834, 8162, 5997, 15138, 13440, 11349, 1230, 8396, 8501, 5553, 2579, 1643, 6562
1	11982, 431, 55, 9, 14451, 8636, 9881, 13479, 6196, 1462, 3998, 5093, 9079, 15162, 10175, 7198, 10908, 11775, 2960, 8868, 2573, 15061, 10998, 1616, 11695, 4707, 2499, 10322, 14163, 11231, 14052, 1042, 1679, 12446, 4162, 8508, 54, 282, 2836, 2839, 11078, 14376	13562, 7640, 13515, 11209, 2803, 3674, 9345, 7752, 4991, 11317, 2332, 11491, 15549, 8697, 13096, 3226, 621, 12155, 2270, 14267, 5438, 510, 1312, 3891, 8693, 8307, 285, 7572, 5421, 10120, 9382, 5876, 571, 1173, 13661, 3505, 6007, 8034, 2121, 4932, 12297, 345

Assignment	Candidate measurements: 42 indices	Prediction cases: 42 indices
2	11982, 431, 55, 9, 14451, 8636, 9881, 13479, 6196, 1462, 9234, 11719, 6993, 11557, 7834, 14631, 3395, 12582, 9875, 9668, 12399, 15587, 1432, 7759, 7097, 1129, 6705, 8256, 3851, 1232, 4092, 13512, 13736, 6273, 13431, 1449, 1609, 2127, 13286, 700, 11697, 4177	15318, 3290, 9718, 6985, 11124, 12571, 4588, 15435, 1007, 5098, 11445, 10868, 2764, 10482, 856, 1655, 1056, 15614, 4016, 6, 6470, 4785, 11753, 14779, 4256, 1255, 3690, 9206, 14358, 11721, 8973, 5856, 1942, 1079, 2099, 8000, 8154, 9735, 5501, 12944, 4285, 14938

These are zero-based indices in the architecture order of the HELP NAS-Bench-201 data. The first ten measurement indices are the reference architectures listed above. Each row is reused across all six development and four test devices; responses remain device-specific.

B.2.4 Initial error estimates and development exclusions

Before acquiring a current-device response, we estimate how much each source predictor may err at each architecture. For a given predictor–architecture pair, we compare its prediction with that same architecture's latency on the allowed development devices and average the absolute differences in the internal units of Equation (B.1). This yields a case-specific initial error estimate, not a single error value for the entire device.

A development device must not supply the response against which its own initial error estimate is assessed. The exclusions also apply when fitting the outcome model, which represents possible unobserved responses for the acquisition calculation. Table B.6 makes these stages explicit. Appendix A.1.2 gives the formula.

Table B.6. Which devices supply HELP initial error estimates?

Stage	Device whose task is evaluated	Devices supplying its error estimate
Choosing settings	One held-out development device	The other five development devices.
Fitting a validation outcome model	One of the five devices included in that fit	The other four devices in that fitting group.
Fitting the final outcome model	One of the six development devices	The other five development devices.
Testing	Any of the four test devices	All six development devices.

The six development validation fits each contain five devices and 15 tasks. Once settings are chosen, one final outcome model uses all six development devices and 18 tasks. That final five-component model, with one component for each source prediction option, and one selected pair of correction/selection settings are used for every test task. The source forests and prepared error estimates stay fixed; acquired responses update the chosen prediction, its correction and the conditional outcome distribution.

B.2.5 HELP comparator under the same paid-information task

The HELP comparator learns from the same five source devices, but predicts with the architecture of Lee et al. (2021): a graph network represents each architecture, and a hardware encoder uses ten measured reference latencies to represent the device. Its inference network estimates a task-specific adjustment from the purchased responses, followed by two parameter updates with step sizes learned during source training. We average ten sampled predictions. This comparator is separate from our five-forest selection-and-correction procedure.

At test time it purchases 10, 12, 16, 20, 28, 36 or 42 responses, always beginning with the ten references and then extending the same measurement-pool prefix. All seven counts are evaluated. The ten-response version is denoted HELP10 in the figures. Its references are part of its ten paid measurements, not additional free measurements.

Only those ten purchased reference responses determine its device-specific normalization. Their minimum and maximum, t_{min} and t_{max}, are fixed throughout the task. The response transformation is $0.05 + 0.9(t - t_{min})/(t_{max} - t_{min})$; the reference-vector hardware representation uses the corresponding [0, 1] scale. Values outside the reference range are not clipped, and unpurchased response ranges or outcome-based filtering are not used. The comparison therefore uses HELP with this explicit paid-information adapter.

Table B.7. Preparation and evaluation settings of the HELP comparator.

Setting	Value and role
Architecture network	Four graph-convolution layers, width 100; the hardware encoder and inference network follow the preserved HELP implementation.
Source training	2,000 meta-training iterations; five source-device tasks per iteration; 200 prediction architectures disjoint from the purchased set in each task.
Training measurement counts	Uniformly sampled from 10, 12, 16, 20, 28, 36 and 42, always including the ten references.
Optimization and adaptation	Meta-training learning rate 10^{-4}; initial learned adaptation step size 10^{-3}; two adaptation steps; task-specific modulation scale 0.01; Kullback–Leibler regularization weight 0.1.
Model selection	Checkpoints evaluated every 100 iterations using mean development log-MAE, equally across six devices, three tasks and all seven counts. The selected checkpoint is at iteration 2,000.
Test prediction	At each acquisition count, we average ten sampled predictions. The same selected checkpoint is used for all four test devices.
Scoring linear outputs	Raw millisecond predictions are used for relative error, millisecond error and ranking. For log scoring only, values are floored at 10^{-6} ms.

The adapter retains each raw linear prediction. Applying a positivity floor only when taking logarithms makes the log metric defined without changing the other scores. During meta-training, the five task prediction losses are averaged, and 0.1 times the sum of their separately computed Kullback–Leibler regularization terms is added; the implementation does not reuse the final task's regularization term for all tasks.

B.3 Settings and acquisition comparisons

B.3.1 What is selected during development?

Two settings govern how acquired responses are used. The correction coefficient α regularizes the kernel regression that adjusts a source predictor. The selection weight s determines how strongly the prepared initial error estimate is weighted relative to errors observed so far. These roles are defined in Section 3.3. They are distinct from the per-measurement acquisition penalty, which controls whether to obtain more responses.

Table B.8. Candidate and selected settings for our procedure.

Setting	ChEMBL	HELP
Correction coefficient α	0.05, 0.1, 0.25, 0.5, 1, 2, 4	0.01, 0.1, 1
Initial-error weight s	0.25, 0.5, 1, 2, 3, 5, 10	0.25, 5
Candidate pairs	All 49 combinations	All 6 combinations
Counts used to score settings	Every count from 0 to 15	0, 2, 4, 8, 12, 16, 20, 28, 36, 42
Settings used on every test task	$\alpha = 0.1$; s = 0.25	$\alpha = 0.01$; s = 0.25
Outcome-model components	10, one per prediction option	5, one per prediction option

B.3.2 Validation splits and final preparation

We choose ChEMBL settings using two levels of protein-wise validation. The development proteins are partitioned into six outer folds. For each outer fold, its held-out proteins are not used to fit that

fold's outcome model. Within the remaining proteins, five inner folds are used to choose α and s. A candidate setting is scored by the actual MAE of its corrected predictions, averaged equally over the declared measurement counts, then over tasks within environment, environments within protein and proteins. Table B.9 records the resulting outer-fold choices.

Table B.9. ChEMBL development splits and settings used on held-out proteins.

Outer split index	Fitting proteins	Held-out proteins	α	s
0	47	10	0.1	0.25
1	47	10	0.1	1.0
2	47	10	0.1	1.0
3	48	9	0.1	0.25
4	48	9	0.1	0.25
5	48	9	0.1	0.25

The outer-held predictions provide development results without fitting the outcome model on the proteins being scored. To choose the final test settings, we aggregate candidate scores from the outer-held development predictions and take the smallest mean MAE, resolving exact ties by smaller α, then smaller s, then the setting identifier. The final choice is $\alpha = 0.1$ and s = 0.25. One final outcome model is fitted using all 57 development proteins.

For that final ChEMBL outcome-model fit, tasks are sorted by protein, environment and assignment, and one deterministic assignment per environment is retained. The fit uses all 30 responses in each of 171 retained tasks, with weight one-third per environment. The three environments for each protein therefore contribute equal total protein weight. The larger set of 513 development tasks remains the evaluation and setting-selection population. These two counts serve different purposes.

HELP holds out one of the six development devices at a time. The outcome model is fitted on the other five devices, with the initial-error exclusions in Table B.6. Each candidate pair is scored on the held-out device's three tasks at the counts in Table B.8. We average counts within task, tasks within device and then the six devices equally. Exact score ties are resolved by smaller α and then smaller s. The selected pair is $\alpha = 0.01$ and s = 0.25; the final outcome model is then fitted using all six development devices.

The outcome model contains the fitted biases, scales, dependence terms and initial mixture-weight temperature described in Appendix A.3.1. Its parameters remain fixed during testing. Test responses do not choose α, s or those fitted parameters. During acquisition, the distribution is conditioned on acquired responses rather than refitted.

B.3.3 Acquisition penalties and their units

The controller compares the expected benefit of further measurements with a penalty for each response acquired. We use λ for the penalty in the controller's calculation and c for the penalty used to evaluate final outcomes. In the original ChEMBL policy, $c = \lambda$. HELP performs its calculation in standardized log-latency units, so the reported penalty c is its internal λ multiplied by the source scale in Equation (B.1). Losses and penalties are converted by the same factor.

Table B.10 lists every penalty evaluated: 34 for ChEMBL and 26 for HELP. The ChEMBL grid contains 31 equally spaced points on a logarithmic scale from 0.00025 to 0.25, with 0, 0.35 and 0.5 added. HELP uses the complete values in the table. Both complete grids are evaluated for every test task, and every resulting curve point is retained.

Table B.10. Complete acquisition-penalty grids.

Index within each grid	ChEMBL $c = \lambda$ pKi per response	HELP internal λ standardized log units	HELP c log-MAE units per response
0	0	0	0
1	0.00025	8.339478544e-05	5.089704326e-05
2	0.0003147313529	0.0001259935982	7.689571455e-05
3	0.0003962232981	0.0001903522707	0.0001161747429

Index within each grid	ChEMBL c = λ pKi per response	HELP internal λ standardized log units	HELP c log-MAE units per response
4	0.0004988155787	0.000287585937	0.0001755178552
5	0.0006279716079	0.0004344874417	0.0002651739674
6	0.000790569415	0.0006564275673	0.0004006272349
7	0.0009952679264	0.0009917367219	0.0006052712598
8	0.001252968084	0.001498324834	0.0009144493086
9	0.001577393361	0.002263682747	0.001381558309
10	0.001985820587	0.003419992424	0.002087270822
11	0.0025	0.005166955571	0.003153467688
12	0.003147313529	0.007806283336	0.004764287582
13	0.003962232981	0.01179380366	0.007197928887
14	0.004988155787	0.01781818553	0.01087469624
15	0.006279716079	0.02691987629	0.0164295897
16	0.00790569415	0.04067079324	0.02482197313
17	0.009952679264	0.06144580327	0.03750126211
18	0.01252968084	0.09283287684	0.0566572469
19	0.01577393361	0.1402527522	0.08559828245
20	0.01985820587	0.2118951301	0.1293226614
21	0.025	0.3201330842	0.1953818497
22	0.03147313529	0.4836599669	0.2951846705
23	0.03962232981	0.7307178644	0.4459676773
24	0.04988155787	1.103975176	0.6737720113
25	0.06279716079	1.667895709	1.017940865
26	0.0790569415	—	—
27	0.09952679264	—	—
28	0.1252968084	—	—
29	0.1577393361	—	—
30	0.1985820587	—	—
31	0.25	—	—
32	0.35	—	—
33	0.5	—	—

The indices identify positions within two different grids; equal indices do not imply matched costs across applications. A dash means that the HELP grid has no entry at that index. Values are displayed to ten significant digits from the retained settings. The internal and reported HELP costs describe the same original policy setting.

B.3.4 Measurement selection and stopping comparisons

To test which measurement to choose, we compare posterior geometry with random selection at each fixed count: 0–15 for ChEMBL and 0–42 for HELP. Posterior geometry is the proposal rule in Section 3.5. The random comparator uses the mean over all 32 predefined whole-pool permutations for each task, rather than its best permutation. Each count uses a prefix of that same order. Both rules use the same prepared predictions, option-selection rule and correction. No stopping decision is applied in this equal-count comparison.

To test when to stop, both policies use posterior geometry. At each evaluation cost c, the fixed-count comparator chooses one integer count k using development data and minimizes prediction MAE plus c times the number acquired. The same chosen count is then applied to every test task. The adaptive policy instead uses its measurements to decide when to stop. Their final outcomes are evaluated with the same loss:

$$J_c \;=\; L_Q \;+\; c\,k. \quad \text{(B.2)}$$

Here L_Q is the actual prediction-set MAE in pKi units for ChEMBL or natural-log-latency units for HELP. Comparing policies at the same c gives them the same cost per response; it does not force them to buy the same number of responses.

A supplementary adaptive comparison also allows the controller penalty to be chosen from development data. For each c, it selects one of the original 34 ChEMBL or 26 HELP adaptive settings by the same development objective. Its controller penalty may differ from c, but its test outcome is still

scored at c. It selects an existing adaptive setting, rather than fitting a new stopping rule. We keep it separate from the original-penalty adaptive comparison.

Fixed counts and adaptive settings are selected by the smallest development objective, then fewer mean acquisitions, then the smaller count or original penalty index. All comparisons use the fixed development groups. The separate valuation-rule comparison in Section 4.3 uses the protocol and results in Appendix C.5, including its distinct controller grid and uncertainty calculations. For HELP, the held-device development paths used here are selection data: α and s were themselves chosen using development results. These development scores are not an independent test of the complete selection process.

Table B.11. Comparisons and the decision each one examines.

Comparison	**What changes**	**What stays the same**
Adaptive versus all-pool	Number and identity of acquired responses and the stopping point	Prepared options, final selection/correction rule and prediction cases.
Posterior geometry versus random, at a fixed count	The identities and order of the acquired responses	Acquisition count and final prediction rule; all 32 random orders are averaged.
Adaptive versus development-selected fixed count	Whether the number of measurements is decided in advance or from the task's measurements	Posterior-geometry proposal rule, final prediction rule and evaluation penalty c.
Development-selected adaptive setting versus fixed count	Adaptive is also given a development-based choice of its controller penalty	Evaluation remains at the same c; test responses do not choose either setting.
Our fixed-count predictions versus HELP	Prediction procedure, including how purchased responses are used	Same measurement pool, prediction cases and integer count: 10, 12, 16, 20, 28, 36 or 42.

Appendix A.5 specifies the numerical purchase rule. To decide whether to acquire the proposed response now, the procedure requires an estimated net gain greater than 10^{-10} in internal loss units. Within that value calculation, each possible first response is followed by a comparison between stopping and acquiring once more. This second comparison chooses another measurement whenever its cost-inclusive expected loss is strictly lower; it does not apply the 10^{-10} tolerance. After an actual response is acquired, the procedure plans again from the updated history. Exact proposal ties use the lexical case identifier; exact option-selection ties use the fixed order of prediction options.

B.4 Prediction quality, aggregation and uncertainty

B.4.1 Prediction error and measurement savings

Mean absolute error is calculated on the prediction cases, not on the responses acquired for correction. For each task, we average the absolute difference between its prediction and response over the 15 ChEMBL or 42 HELP prediction cases. The score uses the final output selected by the procedure.

ChEMBL uses pKi differences. HELP's principal metric, log-MAE, is the mean absolute difference between natural-log latencies; it is not the logarithm of an MAE. We also report mean absolute millisecond error and mean absolute relative latency error, $|\hat{t}/t - 1|$. HELP predictions from our procedure are transformed back from the internal units before these quantities are calculated. The source mean cancels from log-error differences, while the source scale does not.

A zero acquisition penalty does not force the adaptive procedure to acquire every response: it can stop when its modeled continuation does not improve on stopping now. The all-pool reference is therefore evaluated separately by acquiring all 15 ChEMBL or all 42 HELP responses and applying the same final prediction rule.

Measurement savings are 100 $\times$ (1 - mean acquired count / pool size), with pool size 15 for ChEMBL and 42 for HELP. Error differences against the all-pool reference subtract its error on the same task before aggregation. Negative differences indicate lower error than the reference. Savings describe response counts, not measured expenditure or elapsed-time savings.

B.4.2 ChEMBL ranking and molecule selection

An accurate average prediction need not identify the most strongly binding molecule. We therefore retain the molecule-selection metrics in Table B.12. Candidates are ranked by descending predicted pKi; truth rankings use descending measured pKi. Lexical molecular identifiers resolve ties when a fixed-size selection is required.

Table B.12. Additional ChEMBL quality metrics.

Metric	Task-level definition
Spearman rank correlation	Correlation between average ranks of predictions and responses. The implementation returns 0 when either rank standard deviation is at most 10^{-15}.
Top-1 hit	An indicator that the highest-predicted molecule belongs to the measured best set. This set contains responses within 10^{-12} pKi of the maximum.
Top-5 recall	Number shared by the predicted top five and the deterministically selected measured top five, divided by five.
Best molecule in the predicted top five	An indicator that the five-molecule shortlist contains any member of the measured best set.
Top-1 selection loss	Maximum measured pKi minus the measured pKi of the highest-predicted molecule.
Severe top-1 miss	An indicator that the top-1 selection loss is strictly greater than 0.5 pKi.
Same-history option-selection regret	MAE of the selected corrected option minus the smallest actual MAE among all corrected options using exactly the same acquired history. This is a retrospective option comparison, not a molecule-selection loss.

B.4.3 HELP ranking, shortlist and tail errors

For latency prediction, lower values are preferable. We rank architectures by predicted latency and resolve selection ties using the visible architecture identifier. In the following definitions, t_{best} is the smallest measured latency in the prediction set, t_{sel} is the measured latency of the predicted-fastest architecture, and S_5 and T_5 are the predicted and measured five-fastest sets.

Table B.13. Additional HELP quality metrics.

Metric	Task-level definition
Spearman rank correlation	Correlation of average tied ranks. A constant prediction or response ranking gives a missing value. A missing task value propagates to its device and aggregate summary.
Relative penalty of the predicted-fastest architecture	$t_{\text{sel}} / t_{\text{best}} - 1$.
Extra milliseconds of the selected architecture	$t_{\text{sel}} - t_{\text{best}}$.
Log penalty of the selected architecture	$\ln(t_{\text{sel}} / t_{\text{best}})$.
Five-architecture shortlist: mean latency penalty	Mean measured latency in S_5 divided by mean measured latency in T_5, minus one.
Five-architecture shortlist: best latency penalty	Smallest measured latency in S_5 divided by t_{best}, minus one.
Relative-error 95th percentile	Within task: the 95th percentile of 42 absolute relative errors. Within device: the percentile of its 126 prediction errors across the three tasks.
Worst relative error	The largest absolute relative error among all 504 scored prediction cases. A paired comparison of mean device maxima is reported separately when used.

The latency-penalty quantities are bounded below at zero in the scoring implementation. The curve's mean-device 95th percentile averages the four device percentiles; it is not the percentile of a single pooled 504-case distribution. HELP's raw linear outputs are used for ranking and relative/millisecond errors, with the positivity floor in Table B.7 used only for logarithms. These speed-based selection metrics evaluate latency, not neural-network accuracy or a complete architecture-search objective.

B.4.4 KEEPING RELATED TASKS TOGETHER

For ChEMBL, we first average the 32 random orders when they are present, then tasks within environment, environments within protein, and finally proteins with equal weight. Test results therefore give each of the 15 proteins equal weight. HELP averages its three tasks within each device and then the four test devices equally. Development selection uses the analogous hierarchy for its own fixed group.

A paired comparison matches the same task and setting in two methods, calculates their difference there, and then uses the same hierarchy. A protein or device remains one group even when it appears at many measurement counts or penalties. We do not treat those repeated results as new independent proteins or devices.

The definitions of a missing value are preserved for each analysis. In particular, a zero denominator in a diagnostic ratio stays missing. HELP's constant-rank convention differs from the ChEMBL convention in Table B.12; it is not replaced by zero or silently dropped during aggregation.

B.4.5 POINTWISE INTERVALS AND SIMULTANEOUS BANDS

For the performance comparisons, bootstrap resampling draws entire proteins or devices with replacement and keeps all of each selected group's tasks together (Field and Welsh, 2007). A pointwise 90% interval describes one specified mean or paired difference. It uses the 5th and 95th percentiles of the bootstrap means. The original sampling rules are listed in Table B.14.

Table B.14. Resampling rules and the comparison families covered.

Result	Resampling	Reported uncertainty
ChEMBL main trade-off and secondary quality	100,000 paired whole-protein resamples	90% pointwise percentile intervals for the retained contrasts; no main-curve simultaneous band.
ChEMBL posterior geometry versus random	20,000 paired whole-protein resamples	90% pointwise intervals and an approximate 90% simultaneous band over all 16 measurement counts.
ChEMBL adaptive versus fixed count	The same 20,000-resample convention	90% pointwise intervals; a separate approximate simultaneous band over the 34 cost-inclusive-loss differences for each adaptive comparison.
HELP main, fixed-count and native-HELP comparisons	All $4^4 = 256$ ordered resamples of four device means; linear percentile interpolation	90% pointwise intervals. Separate simultaneous families cover 26-cost adaptive–fixed loss curves, 26-cost adaptive–all-pool/no-acquisition log-error curves, and our fixed-count procedure versus the HELP predictor at 10, 12, 16, 20, 28, 36 and 42 measurements (log-MAE).
HELP posterior geometry versus random	All 256 ordered device resamples	90% pointwise intervals and an approximate simultaneous band over all 43 measurement counts.
HELP additional quality contrasts	All 256 ordered device resamples	90% pointwise intervals. For log-MAE only: separate bands for adaptive–all-pool, adaptive–HELP10, each adaptive–fixed comparison, and our fixed-count procedure versus the HELP predictor at 10, 12, 16, 20, 28, 36 and 42 measurements.
HELP local value and score-ordering diagnostics	Whole-device resampling over the participating devices	90% pointwise intervals. With four participating devices, all 256 resamples are enumerated; the zero-cost actual-stop subset with three devices uses all $3^3 = 27$.

A simultaneous band accounts for looking across the whole declared curve. For the unstandardized curve bands summarized in Table B.14, we subtract, for each bootstrap sample, the observed mean-difference curve, take the largest absolute deviation over its counts or costs, and use the 90th percentile of these maxima as a common radius around the observed curve. Bands are constructed sep-

arately for their stated comparison families. A log-MAE band is not a joint band for error and acquisition savings, or for every secondary metric. Appendix C.5.3 specifies the standardized-deviation bands used for the separate valuation-rule comparison.

The bootstrap resamples protein or device groups while keeping the fitted models, selected settings, task assignments and acquisition orders fixed. Fitting and setting selection are not repeated within a bootstrap sample.

The deletion analyses used in C.6 answer a different question: how much does a reported summary change when one entire protein or device is omitted? Their brackets give the smallest and largest resulting summaries. They are sensitivity ranges, not confidence intervals. Counts of favorable costs or measurement depths describe the curve and are not counts of independent successful tests.

B.5 Online computation and acquisition-time scenarios

The time spent selecting measurements and producing predictions is different from the time needed to obtain the measurements. We measure the computation on the same central processing unit (CPU) host for our procedure and the HELP predictor, using the four test devices and three tasks per device. Both methods use their existing fitted models and settings. The HELP latency response describes an architecture's execution on a device; obtaining a measurement can additionally require setup, compilation, transfer, warm-up or repeated executions.

Table B.15. Same-host timing protocol for our procedure and the HELP predictor.

Item	**Measured scope**
Computing environment	CPU execution on one host, with one worker and one thread per numerical library; PyTorch computation also uses one thread. Methods run separately, with no concurrent training.
Starting point	Models, source predictions and public input representations are already in memory. Each measurement pass starts a fresh interpreter and warms up the method before timed calls.
Adaptive run	Start with an empty measurement history; evaluate one of the 26 original penalties; compute proposals and stopping decisions as measurements are acquired; return the final selected-and-corrected prediction at the first stop or pool exhaustion.
Fixed-count run	Use posterior geometry to choose each next measurement during the timed run, then predict after 10, 12, 16, 20, 28, 36 or 42 acquisitions. No stopping-value calculation is needed.
HELP predictor	Use the same seven acquisition counts, including its ten reference responses. Time input checks, normalization from purchased responses, graph indexing, two adaptation steps for each of ten prediction samples, and final prediction.
All-pool reference	Acquire all 42 responses directly and produce the final prediction, without choosing an acquisition order or evaluating stopping. The no-acquisition reference is also timed.
Repetitions and aggregation	Two passes with three repeats per task and setting in each pass: six repeats in total. Average all six durations within task, then the three tasks within device, and then the four devices equally. Task medians are summarized separately.
Measurement order	HELP, our procedure, our procedure, HELP; reverse task and setting order in the second pass. Retain every repeat, including slow ones.
Recorded family	26 adaptive penalties; seven fixed counts for each of the two methods; our all-pool and no-acquisition references. These 42 settings give 504 task–setting combinations and 3,024 timed procedure calls.
Included / excluded	Procedure time includes the method's measurement/prediction calculation and conversion of outputs to milliseconds. Both methods exclude offline training, model/file loading, public graph preprocessing, source prediction preparation, scoring, verification, serialization and physical measurement waits.

We measure elapsed wall-clock time with a high-resolution timer. For each penalty, we time a separate adaptive run from an empty measurement history to its stopping point and final prediction. Each call performs its own measurement choices and calculations. After timing, we check its acquired identities, acquisition count and predictions against the corresponding original result. Timing summaries are descriptive; no confidence intervals are calculated for them.

We report two computation times. Procedure time starts with the visible task inputs already prepared. Preparation-inclusive time adds our procedure's per-task calculation of initial error estimates, lookup of prepared source predictions, kernel construction and input checks. The HELP predictor's corresponding input processing and normalization are already included in its procedure time. Preparation-inclusive times are used for the primary timing comparisons; both scopes are retained in the detailed results.

The two methods retain their existing software environments. They share the CPU host and thread limits, but their library versions differ. Appendix D records the processor and software configuration.

For a sequential scenario, let a be the assumed time required to collect one response, $T_{compute}$ the measured computation time under the stated accounting scope, and k the acquired count. Total time is

$$T_{total} = T_{compute} + k\,a. \quad \text{(B.3)}$$

For the 100 ms scenario, we set a = 0.1 seconds for every response and every method. We assume that measurements are collected one at a time and that computation and measurement do not run in parallel. Total time is therefore the measured computation time plus k times the assumed time per measurement. Changing a changes this total, not the acquisition count or prediction error evaluated for each policy.

For comparisons with HELP10, a break-even duration is calculated only when our procedure acquires fewer responses. We subtract HELP10's mean computation time from ours and divide by the mean number of measurements saved. A positive value gives equal total times at that duration and a shorter total time above it. If our computation is already faster, the total is shorter even at zero acquisition time. These thresholds use differences of aggregate means. Prediction error is reported alongside the time comparison.

These are conditional time scenarios, not measurements of physical profiling savings. The acquisition penalty c remains a charge in prediction-loss units, separate from the assumed duration a.

B.6 Protocols for the explanatory analyses

B.6.1 Histories and prediction subsets

The analyses in Section 4.5 and Appendix C.6 examine the outcomes of the existing procedure rather than choose a new policy. A history contains the identities and responses acquired so far. We retain the complete posterior-geometry path, which continues until the acquisition pool is exhausted, as well as the stopping point of each original cost setting. Some analyzed histories can therefore occur after that setting would already have stopped.

Let k denote the number of responses already acquired. The common comparison covers k = 1–14 in both applications. HELP is additionally summarized over k = 1–41. The initial state k = 0 is separate. When restricting to histories the adaptive policy reaches, we include its stopping point; the separate stopping subset contains only histories where it stops with candidates remaining. A matched pair is included only when both histories belong to the stated subset.

The main prediction sets contain all 15 ChEMBL molecules or all 42 HELP architectures. An additional HELP comparison uses a fixed 15-architecture prediction subset to match the number of prediction cases in ChEMBL. Architecture identifiers are ordered by their SHA-256 keys, with lexical tie-breaking, and the first 15 are retained. For each full or restricted prediction set, alternating entries of its predetermined order give disjoint subsets A and B, as evenly sized as possible. Only description or scoring uses these subsets; the original acquisition pool and full-prediction-set decisions are unchanged.

B.6.2 What each controlled comparison changes

A candidate residual is its response minus the prediction corrected using the current history. The frozen-initial score is the score computed for the same task and cost before any response was acquired; the current score uses the measurements acquired so far. Table B.16 separates the comparisons that use these quantities.

Table B.16. Explanatory comparisons and their fixed inputs.

Question	Fixed quantities	Diagnostic calculation
Where can one response change predictions?	Kernel, correction regularization and acquired identities; one prediction option for the one-response update.	Calculate the correction coefficients before revealing any remaining response. Their breadth, overlap and strength variation are defined in C.6.1.
Does a response's residual fit the correction it triggers?	Coefficient vectors, current prediction errors and the multiset of candidate residual values.	Pair the same residual values one-to-one with different candidate coefficient vectors; compare the original pairing with the exact mean over all reassignments (C.6.3–C.6.4).
Does the current state help assess another purchase?	Device, measurement count, cost, original candidate proposal and conditional second-purchase rule.	Compare current and frozen-initial score orderings against realized gains across task histories; separately compare gains on opposite prediction subsets (C.6.5).
Which useful corrections were missed when ChEMBL stopped?	Initial Q-error values, acquired responses, entire prediction path, option choices, scores, purchase counts and costs.	Reassign initial errors among prediction cases, with the same reassignment across depths in a task; score the fixed predictions and compare policy endpoints (C.6.6).

B.6.3 Local gains, final outcomes and averaging

A candidate residual becomes available only when that response is revealed. Checking whether a correction reduces error additionally requires the withheld prediction responses. All-candidate residual and error comparisons use those outcomes retrospectively, after fixing the original decisions. They do not feed hidden responses into the acquisition rule.

For the local value diagnostic, we evaluate the proposed measurement followed by one further measurement only when the original one-further-measurement rule requests it after the first response. The realized net gain subtracts the costs of the responses actually included in this conditional plan from its prediction-error reduction. This is the same plan used for the expected-value calculation, not a forced two-response purchase or the policy's entire later path. Appendix C.4 gives the definitions and C.6.5 the score-ordering statistic.

To relate correction structure to acquisition performance, we calculate the descriptor on one prediction subset and the performance contrast on the other, and then reverse their roles. Comparisons match the measurement count. For the stopping-score analysis, they also match the device and cost before comparing histories. A gain-difference-weighted ordering statistic and a Spearman correlation are separate summaries, as defined in C.6; neither is silently replaced by classification accuracy.

The two reassignment controls in Table B.16 change different objects. Candidate-residual reassignment is a local calculation holding the current prediction option fixed. ChEMBL initial-error reassignment changes which original Q error is attached to each prediction case while retaining the full prediction and decision paths. Its exact average is computed separately within each task and evaluation subset. It is not a single relabelled database shared across overlapping tasks, and no policy is executed on those synthetic responses.

In HELP's final comparison, the observed difference between adaptive and fixed-count policies is divided into prediction correction, option reselection and acquisition cost along the original path between their endpoints. This ordered accounting is distinct from the local residual-pairing calculation and is reported in C.4 and C.6.7. It does not estimate the performance of a new policy that removes reselection.

When normalizing a local gain, we divide it by the initial MAE of that same task and prediction subset before averaging. Variances use the square of that initial MAE. The initial denominator is not recomputed using an option selected at a later history. Random orders are averaged first when present, then included counts within task, tasks within environment or device, and proteins or devices equally. Scope-specific ratios keep their declared numerator and denominator aggregation, rather than averaging ratios from individual matched pairs.

Undefined comparisons remain undefined: for example, a cosine overlap needs at least two nonzero candidate vectors, a rank statistic needs variation, and a ratio needs a nonzero denominator. A

single remaining candidate has zero candidate-gain variance but does not supply a candidate-pair correlation. The table notes in C.6 identify exclusions and participating counts. Sensitivity analyses omit complete proteins or devices, and the bootstrap intervals retain the rules in B.4.

Appendix C.4 reports the estimated and realized gains of the additional-measurement plan. Appendix C.6 connects correction structure, useful residual patterns, the stopping score and final policy outcomes, retaining the detailed comparisons and sensitivity results for each question.

C Full Results and Additional Analyses

C.1 ChEMBL: Full Prediction and Selection Results

This section gives the complete prediction-quality results behind Section 4.2. Each test task predicts pKi for 15 molecules in Q while acquiring responses from a separate pool P of 15 molecules. Results average the three tasks within each protein and then give the 15 test proteins equal weight. Appendix B specifies preparation and evaluation.

The results below use the reported measurement cost c; the original ChEMBL policy uses $\lambda = c$ in its stopping calculation (Section 4.1; Appendix B.3.3). All-15 acquires the whole pool and applies the same prediction-option selection and residual correction. Changes below subtract all-15 within the same tasks. The 90% pointwise intervals use 100,000 paired whole-protein resamples, holding fitted models, selected settings and tasks fixed.

Table C1.1. Full test curve: measurements and prediction error.

Index	c (pKi/response)	Mean k	Measurement reduction (%)	90% CI for reduction (%)	MAE (pKi)	ΔMAE (pKi)	90% CI for ΔMAE
All-15	—	15.00	0.00	—	0.46495	—	—
0	0	13.53	9.78	[4.44, 17.04]	0.46668	+0.00172	[-0.00268, 0.00620]
1	0.00025	13.04	13.04	[7.56, 20.00]	0.46645	+0.00149	[-0.00303, 0.00609]
2	0.00031473135	13.00	13.33	[8.00, 20.15]	0.46640	+0.00145	[-0.00304, 0.00609]
3	0.0003962233	12.89	14.07	[8.59, 20.89]	0.46586	+0.00090	[-0.00391, 0.00600]
4	0.00049881558	12.76	14.96	[9.48, 21.78]	0.46620	+0.00124	[-0.00362, 0.00628]
5	0.00062797161	12.49	16.74	[11.41, 23.26]	0.46695	+0.00200	[-0.00326, 0.00722]
6	0.00079056942	12.36	17.63	[12.30, 24.15]	0.46722	+0.00226	[-0.00302, 0.00754]
7	0.00099526793	11.98	20.15	[14.81, 26.52]	0.46802	+0.00306	[-0.00259, 0.00865]
8	0.0012529681	11.49	23.41	[18.52, 29.19]	0.47023	+0.00528	[-0.00148, 0.01225]
9	0.0015773934	10.71	28.59	[23.70, 34.07]	0.46817	+0.00321	[-0.00281, 0.00972]
10	0.0019858206	9.82	34.52	[28.89, 40.59]	0.47542	+0.01047	[0.00314, 0.01901]
11	0.0025	9.07	39.56	[34.81, 44.89]	0.47877	+0.01382	[0.00538, 0.02321]
12	0.0031473135	8.09	46.07	[41.77, 50.81]	0.48251	+0.01756	[0.00727, 0.02855]
13	0.003962233	7.20	52.00	[48.00, 56.15]	0.49110	+0.02615	[0.01083, 0.04166]
14	0.0049881558	6.78	54.81	[50.81, 58.81]	0.49865	+0.03370	[0.01626, 0.05329]
15	0.0062797161	5.47	63.56	[60.89, 66.22]	0.52027	+0.05531	[0.03537, 0.07590]
16	0.0079056942	4.84	67.70	[65.04, 70.22]	0.53231	+0.06736	[0.03983, 0.09567]
17	0.0099526793	4.47	70.22	[67.56, 72.89]	0.53197	+0.06702	[0.03962, 0.09479]
18	0.012529681	3.84	74.37	[72.30, 76.30]	0.54210	+0.07715	[0.04814, 0.10583]
19	0.015773934	3.29	78.07	[77.04, 79.11]	0.56777	+0.10281	[0.06752, 0.14128]
20	0.019858206	2.89	80.74	[79.70, 81.78]	0.59600	+0.13105	[0.08914, 0.17604]
21	0.025	2.42	83.85	[82.52, 85.04]	0.61336	+0.14840	[0.10590, 0.19355]
22	0.031473135	2.20	85.33	[84.44, 86.07]	0.63695	+0.17199	[0.12413, 0.22241]
23	0.03962233	1.87	87.56	[86.52, 88.59]	0.64957	+0.18462	[0.12940, 0.24172]
24	0.049881558	1.36	90.96	[89.78, 92.00]	0.71544	+0.25049	[0.17435, 0.33147]
25	0.062797161	1.09	92.74	[92.30, 93.19]	0.73064	+0.26569	[0.19142, 0.34190]
26	0.079056942	1.04	93.04	[92.74, 93.33]	0.73185	+0.26689	[0.19126, 0.34528]
27	0.099526793	1.00	93.33	[93.33, 93.33]	0.73036	+0.26540	[0.19052, 0.34342]
28	0.12529681	0.96	93.63	[93.33, 94.22]	0.73714	+0.27219	[0.19611, 0.35031]
29	0.15773934	0.69	95.41	[94.22, 96.59]	0.78728	+0.32232	[0.22993, 0.42399]
30	0.19858206	0.33	97.78	[96.44, 98.96]	0.82789	+0.36294	[0.25188, 0.47748]
31	0.25	0.00	100.00	[100.00, 100.00]	0.91822	+0.45327	[0.28872, 0.64776]

Index	c (pKi/response)	Mean k	Measurement reduction (%)	90% CI for reduction (%)	MAE (pKi)	ΔMAE (pKi)	90% CI for ΔMAE
32	0.35	0.00	100.00	[100.00, 100.00]	0.91822	+0.45327	[0.28838, 0.64568]
33	0.5	0.00	100.00	[100.00, 100.00]	0.91822	+0.45327	[0.28903, 0.64562]

Indices 0–33 identify the complete penalty grid; the same indices are used in later tables. Measurement reduction is 100 × (1 - mean k / 15). All-15 is the reference, so its contrast intervals are not applicable. At indices 31–33, the procedure uses no current responses.

C.1.1 Ranking and molecule selection

Seven additional metrics assess molecular ranking and the identification of high-affinity molecules in Q, either individually or in a five-molecule shortlist. They also assess the choice among prepared prediction options by comparing their corrected prediction errors at the same measurement history. Tables C1.2–C1.5 report every penalty setting and its paired change from all-15. Appendix B.4.2 gives the exact definitions and tie rules. All brackets in these tables are 90% pointwise intervals; differences between rates are percentage points.

Spearman correlation measures the ordering of all 15 prediction molecules. A top-1 hit occurs when the highest-predicted molecule is one of the molecules with the highest measured affinity. Higher values are preferable for both metrics.

Table C1.2. Full-grid ranking and top-1 hit results.

Index	Spearman	Δ from all-15	90% CI for Δ	Top-1 hit (%)	Δ from all-15	90% CI for Δ
All-15	0.5072	—	—	24.44	—	—
0	0.5000	-0.0072	[-0.0255, 0.0078]	24.44	0.00	[0.00, 0.00]
1	0.5007	-0.0066	[-0.0243, 0.0073]	26.67	+2.22	[0.00, 6.67]
2	0.5048	-0.0024	[-0.0220, 0.0144]	26.67	+2.22	[0.00, 6.67]
3	0.5062	-0.0010	[-0.0182, 0.0147]	26.67	+2.22	[0.00, 6.67]
4	0.5008	-0.0064	[-0.0236, 0.0094]	26.67	+2.22	[0.00, 6.67]
5	0.4962	-0.0110	[-0.0286, 0.0058]	26.67	+2.22	[0.00, 6.67]
6	0.4909	-0.0163	[-0.0368, 0.0033]	26.67	+2.22	[0.00, 6.67]
7	0.4893	-0.0179	[-0.0388, 0.0018]	26.67	+2.22	[0.00, 6.67]
8	0.4786	-0.0286	[-0.0524, -0.0065]	24.44	0.00	[0.00, 0.00]
9	0.4819	-0.0254	[-0.0478, -0.0047]	26.67	+2.22	[0.00, 6.67]
10	0.4459	-0.0614	[-0.1084, -0.0207]	28.89	+4.44	[0.00, 8.89]
11	0.4451	-0.0622	[-0.1106, -0.0200]	33.33	+8.89	[2.22, 17.78]
12	0.4543	-0.0530	[-0.0979, -0.0149]	24.44	0.00	[-4.44, 4.44]
13	0.4460	-0.0612	[-0.1234, -0.0113]	26.67	+2.22	[-6.67, 13.33]
14	0.4329	-0.0744	[-0.1403, -0.0226]	26.67	+2.22	[-6.67, 13.33]
15	0.3857	-0.1215	[-0.1982, -0.0589]	24.44	0.00	[-11.11, 11.11]
16	0.3665	-0.1407	[-0.2242, -0.0688]	22.22	-2.22	[-11.11, 6.67]
17	0.3614	-0.1459	[-0.2321, -0.0702]	22.22	-2.22	[-11.11, 6.67]
18	0.3689	-0.1383	[-0.2289, -0.0555]	20.00	-4.44	[-13.33, 4.44]
19	0.3497	-0.1576	[-0.2486, -0.0746]	13.33	-11.11	[-22.22, 0.00]
20	0.3332	-0.1740	[-0.2667, -0.0872]	17.78	-6.67	[-17.78, 4.44]
21	0.3145	-0.1927	[-0.2833, -0.1057]	17.78	-6.67	[-20.00, 6.67]
22	0.3167	-0.1906	[-0.2743, -0.1093]	17.78	-6.67	[-20.00, 6.67]
23	0.3060	-0.2012	[-0.2628, -0.1390]	17.78	-6.67	[-20.00, 6.67]
24	0.2589	-0.2484	[-0.3076, -0.1902]	15.56	-8.89	[-22.22, 4.44]
25	0.2599	-0.2473	[-0.3138, -0.1819]	15.56	-8.89	[-22.22, 4.44]
26	0.2619	-0.2453	[-0.3124, -0.1800]	15.56	-8.89	[-22.22, 4.44]
27	0.2543	-0.2529	[-0.3225, -0.1849]	15.56	-8.89	[-22.22, 4.44]
28	0.2551	-0.2521	[-0.3214, -0.1839]	15.56	-8.89	[-22.22, 4.44]
29	0.2340	-0.2732	[-0.3476, -0.2005]	15.56	-8.89	[-22.22, 4.44]
30	0.2450	-0.2622	[-0.3314, -0.1953]	17.78	-6.67	[-17.78, 4.44]

Index	Spearman	Δ from all-15	90% CI for Δ	Top-1 hit (%)	Δ from all-15	90% CI for Δ
31	0.2355	-0.2717	[-0.3523, -0.1968]	15.56	-8.89	[-22.22, 4.44]
32	0.2355	-0.2717	[-0.3532, -0.1975]	15.56	-8.89	[-22.22, 4.44]
33	0.2355	-0.2717	[-0.3532, -0.1970]	15.56	-8.89	[-22.22, 4.44]

Five-molecule shortlist quality Top-five recall is the fraction of the five highest-affinity molecules included in the predicted five-molecule shortlist. "Best in five" records whether that shortlist contains any molecule with the highest measured affinity. Higher values are preferable.

Table C1.3. Full-grid five-molecule shortlist results.

Index	Top-5 recall (%)	Δ from all-15	90% CI for Δ	Best in five (%)	Δ from all-15	90% CI for Δ
All-15	59.11	—	—	62.22	—	—
0	59.11	0.00	[-1.33, 1.33]	62.22	0.00	[-4.44, 4.44]
1	59.11	0.00	[-1.78, 1.78]	62.22	0.00	[-4.44, 4.44]
2	59.56	+0.44	[-1.78, 2.67]	64.44	+2.22	[-4.44, 8.89]
3	60.00	+0.89	[-0.89, 3.11]	64.44	+2.22	[-4.44, 8.89]
4	60.00	+0.89	[-0.89, 3.11]	64.44	+2.22	[-4.44, 8.89]
5	60.00	+0.89	[-0.89, 3.11]	64.44	+2.22	[-4.44, 8.89]
6	60.00	+0.89	[-0.89, 3.11]	64.44	+2.22	[-4.44, 8.89]
7	59.56	+0.44	[-1.78, 2.67]	64.44	+2.22	[-4.44, 8.89]
8	59.11	0.00	[-1.78, 1.78]	64.44	+2.22	[-4.44, 8.89]
9	57.33	-1.78	[-3.56, 0.00]	62.22	0.00	[-6.67, 6.67]
10	54.67	-4.44	[-7.11, -1.78]	57.78	-4.44	[-11.11, 2.22]
11	55.56	-3.56	[-6.67, -0.44]	60.00	-2.22	[-11.11, 6.67]
12	57.33	-1.78	[-4.00, 0.89]	60.00	-2.22	[-11.11, 6.67]
13	56.00	-3.11	[-7.11, 0.44]	60.00	-2.22	[-11.11, 6.67]
14	56.00	-3.11	[-7.56, 0.89]	57.78	-4.44	[-15.56, 6.67]
15	53.33	-5.78	[-10.67, -1.33]	53.33	-8.89	[-17.78, 0.00]
16	52.89	-6.22	[-11.56, -1.33]	57.78	-4.44	[-15.56, 4.44]
17	52.44	-6.67	[-12.44, -0.89]	60.00	-2.22	[-13.33, 8.89]
18	50.67	-8.44	[-14.22, -2.67]	57.78	-4.44	[-13.33, 4.44]
19	48.89	-10.22	[-15.11, -4.89]	55.56	-6.67	[-15.56, 2.22]
20	47.11	-12.00	[-17.33, -6.22]	55.56	-6.67	[-15.56, 2.22]
21	44.89	-14.22	[-19.11, -9.33]	48.89	-13.33	[-22.22, -4.44]
22	46.67	-12.44	[-17.33, -7.56]	55.56	-6.67	[-13.33, 0.00]
23	45.33	-13.78	[-18.67, -8.44]	48.89	-13.33	[-22.22, -4.44]
24	43.56	-15.56	[-20.89, -10.22]	44.44	-17.78	[-28.89, -6.67]
25	42.67	-16.44	[-21.33, -11.56]	46.67	-15.56	[-26.67, -4.44]
26	43.56	-15.56	[-20.89, -10.22]	46.67	-15.56	[-26.67, -4.44]
27	43.56	-15.56	[-20.89, -10.67]	46.67	-15.56	[-26.67, -6.67]
28	43.56	-15.56	[-20.89, -10.67]	46.67	-15.56	[-26.67, -4.44]
29	40.44	-18.67	[-24.89, -12.44]	44.44	-17.78	[-28.89, -6.67]
30	41.78	-17.33	[-23.11, -12.00]	44.44	-17.78	[-28.89, -6.67]
31	42.67	-16.44	[-22.67, -11.11]	46.67	-15.56	[-28.89, -2.22]
32	42.67	-16.44	[-22.67, -11.11]	46.67	-15.56	[-28.89, -2.22]
33	42.67	-16.44	[-22.67, -11.11]	46.67	-15.56	[-28.89, -2.22]

Δ and its interval are in percentage points. An interval concerns the paired change, not either method's absolute rate. Equality of one observed rate is not an equivalence result.

Affinity loss and severe misses Top-1 selection loss is the measured affinity gap between the best molecule in Q and the molecule selected by the prediction. A severe top-1 miss means that this gap exceeds 0.5 pKi. Lower values are preferable.

Table C1.4. Full-grid top-1 affinity loss and severe-miss rates.

Index	Top-1 loss (pKi)	Δ from all-15	90% CI for Δ	Severe miss (%)	Δ from all-15	90% CI for Δ
All-15	0.41778	—	—	31.11	—	—
0	0.41200	-0.00578	[-0.01911, 0.00178]	31.11	0.00	[0.00, 0.00]
1	0.40711	-0.01067	[-0.02356, -0.00044]	31.11	0.00	[0.00, 0.00]
2	0.40711	-0.01067	[-0.02356, -0.00044]	31.11	0.00	[0.00, 0.00]
3	0.40711	-0.01067	[-0.02356, -0.00044]	31.11	0.00	[0.00, 0.00]
4	0.40711	-0.01067	[-0.02356, -0.00044]	31.11	0.00	[0.00, 0.00]
5	0.40711	-0.01067	[-0.02311, -0.00044]	31.11	0.00	[0.00, 0.00]
6	0.40956	-0.00822	[-0.02200, 0.00333]	31.11	0.00	[0.00, 0.00]
7	0.43422	+0.01644	[-0.01756, 0.06578]	33.33	+2.22	[0.00, 6.67]
8	0.45378	+0.03600	[-0.03444, 0.10978]	37.78	+6.67	[2.22, 13.33]
9	0.44267	+0.02489	[-0.04689, 0.10133]	35.56	+4.44	[-2.22, 11.11]
10	0.43911	+0.02133	[-0.04978, 0.09667]	35.56	+4.44	[-2.22, 11.11]
11	0.42800	+0.01022	[-0.07889, 0.09933]	37.78	+6.67	[0.00, 13.33]
12	0.51644	+0.09867	[0.02910, 0.17489]	44.44	+13.33	[6.67, 20.00]
13	0.45067	+0.03289	[-0.06667, 0.12889]	37.78	+6.67	[-2.22, 15.56]
14	0.45089	+0.03311	[-0.06600, 0.12933]	37.78	+6.67	[-2.22, 15.56]
15	0.38889	-0.02889	[-0.13778, 0.07667]	33.33	+2.22	[-6.67, 11.11]
16	0.43089	+0.01311	[-0.10334, 0.12578]	35.56	+4.44	[-6.67, 15.56]
17	0.44711	+0.02933	[-0.09422, 0.15111]	35.56	+4.44	[-6.67, 13.33]
18	0.44733	+0.02956	[-0.08711, 0.14444]	35.56	+4.44	[-4.44, 13.33]
19	0.57489	+0.15711	[-0.02067, 0.34712]	42.22	+11.11	[0.00, 22.22]
20	0.60622	+0.18844	[-0.03267, 0.44244]	44.44	+13.33	[0.00, 26.67]
21	0.66844	+0.25067	[0.04733, 0.48623]	48.89	+17.78	[6.67, 31.11]
22	0.70733	+0.28956	[0.08267, 0.52667]	53.33	+22.22	[8.89, 35.56]
23	0.73689	+0.31911	[0.11178, 0.55378]	55.56	+24.44	[11.11, 37.78]
24	0.78311	+0.36533	[0.15244, 0.60511]	57.78	+26.67	[15.56, 37.78]
25	0.78467	+0.36689	[0.15933, 0.60133]	60.00	+28.89	[15.56, 42.22]
26	0.78467	+0.36689	[0.15956, 0.59979]	60.00	+28.89	[15.56, 42.22]
27	0.79800	+0.38022	[0.16711, 0.61778]	60.00	+28.89	[15.56, 42.22]
28	0.79800	+0.38022	[0.16911, 0.61644]	60.00	+28.89	[15.56, 42.22]
29	0.97822	+0.56044	[0.19644, 1.01401]	60.00	+28.89	[15.56, 42.22]
30	0.95622	+0.53844	[0.18889, 0.99044]	57.78	+26.67	[15.56, 37.78]
31	1.01911	+0.60133	[0.24956, 1.04156]	55.56	+24.44	[15.56, 33.33]
32	1.01911	+0.60133	[0.25067, 1.04578]	55.56	+24.44	[15.56, 33.33]
33	1.01911	+0.60133	[0.25044, 1.04067]	55.56	+24.44	[15.56, 33.33]

For top-1 loss, Δ and its interval are in pKi. For the severe-miss rate, they are in percentage points. The 0.5 pKi threshold is the one used in the original evaluation.

Selecting a prediction option at the same history Same-history option-selection regret compares the selected corrected prediction with the lowest-error corrected option among the ten prepared prediction options, using exactly the same acquired responses. It evaluates the choice of a prediction option, not the choice of a molecule. The comparison uses Q responses retrospectively for scoring; those responses are not available to the selection rule. Lower regret is preferable.

Table C1.5. Full-grid same-history option-selection regret (pKi).

Index	c (pKi/response)	Mean regret	Δ from all-15	90% CI for Δ
All-15	—	0.01900	—	—
0	0	0.01775	-0.00124	[-0.00430, 0.00090]
1	0.00025	0.01767	-0.00132	[-0.00439, 0.00096]
2	0.00031473135	0.01760	-0.00140	[-0.00444, 0.00090]
3	0.0003962233	0.01743	-0.00157	[-0.00460, 0.00079]
4	0.00049881558	0.01794	-0.00106	[-0.00427, 0.00151]
5	0.00062797161	0.01787	-0.00112	[-0.00432, 0.00147]
6	0.00079056942	0.01792	-0.00107	[-0.00424, 0.00150]
7	0.00099526793	0.01828	-0.00072	[-0.00398, 0.00195]

Index	c (pKi/response)	Mean regret	Δ from all-15	90% CI for Δ
8	0.0012529681	0.01792	-0.00108	[-0.00430, 0.00158]
9	0.0015773934	0.01747	-0.00153	[-0.00482, 0.00122]
10	0.0019858206	0.01821	-0.00079	[-0.00433, 0.00224]
11	0.0025	0.01814	-0.00086	[-0.00464, 0.00246]
12	0.0031473135	0.01962	+0.00063	[-0.00407, 0.00529]
13	0.003962233	0.01977	+0.00077	[-0.00420, 0.00591]
14	0.0049881558	0.02079	+0.00180	[-0.00339, 0.00716]
15	0.0062797161	0.02106	+0.00206	[-0.00499, 0.00950]
16	0.0079056942	0.01984	+0.00084	[-0.00574, 0.00738]
17	0.0099526793	0.01963	+0.00063	[-0.00581, 0.00706]
18	0.012529681	0.01965	+0.00065	[-0.00646, 0.00800]
19	0.015773934	0.02476	+0.00576	[-0.00342, 0.01557]
20	0.019858206	0.03244	+0.01345	[0.00305, 0.02390]
21	0.025	0.03395	+0.01496	[0.00144, 0.02751]
22	0.031473135	0.03818	+0.01919	[0.00520, 0.03191]
23	0.03962233	0.04327	+0.02427	[0.01129, 0.03653]
24	0.049881558	0.05907	+0.04007	[0.02065, 0.06039]
25	0.062797161	0.06584	+0.04684	[0.02584, 0.06863]
26	0.079056942	0.06503	+0.04603	[0.02510, 0.06764]
27	0.099526793	0.06461	+0.04562	[0.02472, 0.06712]
28	0.12529681	0.07267	+0.05368	[0.03062, 0.07696]
29	0.15773934	0.09064	+0.07164	[0.03454, 0.11356]
30	0.19858206	0.08543	+0.06643	[0.02841, 0.10927]
31	0.25	0.11064	+0.09164	[0.04510, 0.14117]
32	0.35	0.11064	+0.09164	[0.04497, 0.14130]
33	0.5	0.11064	+0.09164	[0.04523, 0.14144]

This retrospective comparison does not measure the benefit of the complete acquisition or stopping policy.

C.1.2 Results for individual test proteins

Tables C1.6–C1.8 show the complete acquired-count and MAE curves for all 15 test proteins. Each entry averages that protein's three prediction tasks. The two columns under each protein give the mean number of pKi responses acquired from P (mean k) and the prediction MAE on Q (pKi). The penalty index is the same as in Table C1.1.

Table C1.6. Acquired counts and MAE for five test proteins.

Penalty	CHEMBL1821		CHEMBL1836		CHEMBL1867		CHEMBL1875		CHEMBL1898	
	Mean		Mean		Mean		Mean		Mean	
Index	k	MAE	k	MAE	k	MAE	k	MAE	k	MAE
All-15	15.00	0.33812	15.00	0.46060	15.00	0.19884	15.00	0.25329	15.00	0.35696
0	15.00	0.33812	15.00	0.46060	14.00	0.19543	15.00	0.25329	14.67	0.35604
1	15.00	0.33812	15.00	0.46060	13.67	0.19784	13.00	0.24555	14.33	0.35466
2	15.00	0.33812	15.00	0.46060	13.67	0.19784	13.00	0.24555	13.67	0.35395
3	15.00	0.33812	15.00	0.46060	13.67	0.19784	13.00	0.24555	13.67	0.35395
4	15.00	0.33812	15.00	0.46060	12.67	0.20188	12.67	0.24640	13.67	0.35395
5	15.00	0.33812	14.67	0.46132	12.67	0.20188	11.33	0.25316	13.67	0.35395
6	14.67	0.33955	14.00	0.46170	12.67	0.20188	11.00	0.25118	13.67	0.35395
7	14.67	0.33955	13.33	0.45769	12.33	0.19945	11.00	0.25118	13.33	0.35650
8	14.67	0.33955	12.33	0.46456	12.00	0.20299	10.67	0.25117	12.00	0.35687
9	13.67	0.33503	12.33	0.46456	12.00	0.20299	9.67	0.24276	10.67	0.35146
10	13.67	0.33503	11.00	0.47274	11.67	0.20029	8.33	0.26377	9.67	0.35346
11	9.67	0.37811	11.00	0.47274	9.67	0.20021	8.33	0.26377	8.00	0.35031
12	8.67	0.41225	10.00	0.47066	9.00	0.18572	6.67	0.26613	7.67	0.35346
13	7.67	0.44726	8.33	0.48300	7.67	0.18132	5.67	0.29462	7.00	0.33651
14	5.33	0.50105	8.00	0.49079	6.67	0.20689	5.33	0.30612	6.67	0.34565
15	5.33	0.50105	6.67	0.54172	5.33	0.22100	4.67	0.32719	5.00	0.36716
16	4.67	0.54105	5.33	0.61763	5.33	0.22100	4.67	0.32719	4.33	0.37305

Penalty	CHEMBL1821		CHEMBL1836		CHEMBL1867		CHEMBL1875		CHEMBL1898	
	Mean		Mean		Mean		Mean		Mean	
Index	k	MAE	k	MAE	k	MAE	k	MAE	k	MAE
17	4.67	0.54105	5.00	0.60435	5.33	0.22100	3.67	0.32402	4.00	0.36423
18	3.67	0.53540	4.33	0.59246	4.33	0.22239	3.33	0.33120	3.33	0.34275
19	3.00	0.54201	3.33	0.55932	4.00	0.22427	3.00	0.33981	3.33	0.34275
20	2.33	0.55573	3.00	0.57408	3.00	0.23507	3.00	0.33981	3.00	0.37858
21	2.33	0.55573	3.00	0.57408	2.67	0.25759	2.33	0.33748	2.00	0.41654
22	2.00	0.53896	2.67	0.60159	2.33	0.28158	2.00	0.34569	2.00	0.41654
23	2.00	0.53896	2.33	0.63525	2.00	0.33353	2.00	0.34569	2.00	0.41654
24	1.33	0.57160	1.00	0.87356	2.00	0.33353	2.00	0.34569	1.00	0.57590
25	1.00	0.60440	1.00	0.87356	1.00	0.47030	1.00	0.35010	1.00	0.57590
26	1.00	0.60440	1.00	0.87356	1.00	0.47030	1.00	0.35010	1.00	0.57590
27	1.00	0.60440	1.00	0.87356	1.00	0.47030	1.00	0.35010	1.00	0.57590
28	1.00	0.60440	1.00	0.87356	1.00	0.47030	1.00	0.35010	1.00	0.57590
29	1.00	0.60440	1.00	0.87356	0.67	0.56492	1.00	0.35010	1.00	0.57590
30	1.00	0.60440	1.00	0.87356	0.00	0.88655	1.00	0.35010	0.67	0.46390
31	0.00	0.59696	0.00	2.18661	0.00	0.88655	0.00	0.45362	0.00	0.33158
32	0.00	0.59696	0.00	2.18661	0.00	0.88655	0.00	0.45362	0.00	0.33158
33	0.00	0.59696	0.00	2.18661	0.00	0.88655	0.00	0.45362	0.00	0.33158

All-15 uses 15 measurements for every task. The final three penalty settings make no acquisitions for any test protein. No protein-level confidence interval is assigned to these descriptive means.

Individual-protein measurement and error curves (continued)

Table C1.7. Acquired counts and MAE for five test proteins.

Penalty	CHEMBL1945		CHEMBL2049		CHEMBL229		CHEMBL249		CHEMBL3764	
	Mean		Mean		Mean		Mean		Mean	
Index	k	MAE	k	MAE	k	MAE	k	MAE	k	MAE
All-15	15.00	0.70559	15.00	0.59052	15.00	0.44406	15.00	0.42805	15.00	0.62815
0	13.33	0.70111	5.67	0.56740	12.67	0.45227	13.33	0.44373	15.00	0.62815
1	13.33	0.70111	5.67	0.56740	11.33	0.44811	13.33	0.44373	14.00	0.62682
2	13.33	0.70111	5.67	0.56740	11.33	0.44811	13.33	0.44373	14.00	0.62682
3	13.33	0.70111	5.67	0.56740	11.33	0.44811	13.33	0.44373	13.33	0.61641
4	13.33	0.70111	5.67	0.56740	11.33	0.44811	13.33	0.44373	13.33	0.61641
5	13.00	0.69977	5.67	0.56740	11.33	0.44811	13.33	0.44373	13.00	0.60864
6	13.00	0.69977	5.67	0.56740	10.67	0.45228	13.33	0.44373	13.00	0.60864
7	13.00	0.69977	5.67	0.56740	9.33	0.46154	12.33	0.44404	13.00	0.60864
8	11.67	0.71629	5.67	0.56740	9.33	0.46154	11.67	0.45400	12.67	0.60597
9	10.67	0.70395	5.67	0.56740	9.00	0.46054	11.33	0.45762	12.00	0.62055
10	10.00	0.69815	4.33	0.61803	9.00	0.46054	11.33	0.45762	11.67	0.62049
11	10.00	0.69815	4.00	0.62148	8.33	0.46272	11.33	0.45762	10.67	0.62039
12	9.00	0.72750	4.00	0.62148	6.00	0.46413	10.00	0.45219	9.67	0.60109
13	8.67	0.73327	4.00	0.62148	5.67	0.46789	9.33	0.45597	9.00	0.58549
14	8.67	0.73327	4.00	0.62148	5.67	0.46789	9.00	0.45643	8.33	0.58546
15	6.67	0.82595	4.00	0.62148	4.33	0.44408	6.33	0.49131	7.00	0.58631
16	6.00	0.83318	3.33	0.70247	4.33	0.44408	6.33	0.49131	5.67	0.57610
17	6.00	0.83318	3.33	0.70247	3.00	0.45733	6.33	0.49131	5.33	0.57481
18	5.67	0.83618	2.67	0.74602	3.00	0.45733	4.33	0.54194	4.33	0.58211
19	3.67	1.03587	2.67	0.74602	3.00	0.45733	3.67	0.57923	3.67	0.65442
20	3.33	1.08911	2.67	0.74602	2.33	0.46029	3.00	0.64986	3.67	0.65442
21	3.33	1.08911	2.00	0.70592	2.00	0.45512	3.00	0.64986	3.00	0.72305
22	3.00	1.12527	2.00	0.70592	2.00	0.45512	2.33	0.65806	2.33	0.97377
23	2.67	1.19946	1.33	0.68715	1.67	0.48367	1.67	0.68488	2.33	0.97377
24	1.67	1.38017	1.00	0.69173	1.00	0.49019	1.33	0.71034	2.00	1.19016
25	1.33	1.33185	1.00	0.69173	1.00	0.49019	1.33	0.71034	1.33	1.19706
26	1.00	1.36551	1.00	0.69173	1.00	0.49019	1.33	0.71034	1.33	1.19706
27	1.00	1.36551	1.00	0.69173	1.00	0.49019	1.00	0.70531	1.00	1.17969
28	1.00	1.36551	1.00	0.69173	1.00	0.49019	0.33	0.80713	1.00	1.17969
29	0.00	1.66206	0.00	1.03855	1.00	0.49019	0.00	0.80617	1.00	1.17969

Penalty	CHEMBL1945		CHEMBL2049		CHEMBL229		CHEMBL249		CHEMBL3764	
	Mean		Mean		Mean		Mean		Mean	
Index	k	MAE	k	MAE	k	MAE	k	MAE	k	MAE
30	0.00	1.66206	0.00	1.03855	0.00	0.44666	0.00	0.80617	0.33	1.04990
31	0.00	1.66206	0.00	1.03855	0.00	0.44666	0.00	0.80617	0.00	0.88490
32	0.00	1.66206	0.00	1.03855	0.00	0.44666	0.00	0.80617	0.00	0.88490
33	0.00	1.66206	0.00	1.03855	0.00	0.44666	0.00	0.80617	0.00	0.88490

All-15 uses 15 measurements for every task. The final three penalty settings make no acquisitions for any test protein. No protein-level confidence interval is assigned to these descriptive means.

Individual-protein measurement and error curves (continued)

Table C1.8. Acquired counts and MAE for five test proteins.

Penalty	CHEMBL4315		CHEMBL4561		CHEMBL4644		CHEMBL4777		CHEMBL4792	
	Mean		Mean		Mean		Mean		Mean	
Index	k	MAE	k	MAE	k	MAE	k	MAE	k	MAE
All-15	15.00	0.49083	15.00	0.81145	15.00	0.42721	15.00	0.47734	15.00	0.36330
0	12.33	0.51843	14.67	0.81670	13.67	0.42739	14.00	0.47452	14.67	0.36699
1	12.33	0.51843	14.67	0.81670	12.33	0.43578	14.00	0.47452	13.67	0.36736
2	12.33	0.51843	14.67	0.81670	12.33	0.43578	14.00	0.47452	13.67	0.36736
3	11.33	0.52072	14.67	0.81670	12.33	0.43578	14.00	0.47452	13.67	0.36736
4	11.33	0.52072	14.67	0.81670	12.00	0.43581	14.00	0.47452	13.33	0.36754
5	11.33	0.52072	14.00	0.82214	12.00	0.43581	13.00	0.48200	13.33	0.36754
6	11.33	0.52072	14.00	0.82214	12.00	0.43581	13.00	0.48200	13.33	0.36754
7	11.00	0.52112	13.33	0.82163	12.00	0.43581	12.33	0.48846	13.00	0.36747
8	10.67	0.53536	12.00	0.80887	12.00	0.43581	12.33	0.48846	12.67	0.36461
9	10.33	0.52853	10.00	0.80896	9.33	0.43443	11.67	0.47912	12.33	0.36463
10	8.33	0.55668	9.00	0.80372	9.33	0.43443	9.00	0.47938	11.00	0.37701
11	7.00	0.56214	9.00	0.80372	9.00	0.43380	9.00	0.47938	11.00	0.37701
12	6.33	0.55553	8.33	0.82371	8.67	0.44060	9.00	0.47938	8.33	0.38382
13	6.33	0.55553	7.00	0.85518	5.67	0.49047	7.67	0.47473	8.33	0.38382
14	6.33	0.55553	6.67	0.85326	5.67	0.49047	7.00	0.48166	8.33	0.38382
15	4.67	0.58060	5.67	0.87567	4.67	0.48394	5.00	0.52797	6.67	0.40853
16	3.67	0.60229	4.00	0.80659	4.33	0.51966	4.33	0.52339	6.33	0.40569
17	3.67	0.60229	3.67	0.80740	4.00	0.53311	4.33	0.52339	4.67	0.39967
18	3.67	0.60229	3.67	0.80740	3.67	0.54525	4.00	0.51844	3.67	0.47037
19	2.67	0.64476	3.33	0.81831	3.33	0.54803	3.00	0.55398	3.67	0.47037
20	2.33	0.66277	3.00	0.83652	2.67	0.65657	3.00	0.55398	3.00	0.54720
21	2.00	0.66696	2.33	0.81520	2.00	0.73181	2.00	0.68730	2.33	0.53461
22	2.00	0.66696	2.33	0.81520	2.00	0.73181	2.00	0.68730	2.00	0.55042
23	1.33	0.72720	1.33	0.75265	1.67	0.75553	1.67	0.65889	2.00	0.55042
24	1.00	0.73494	1.00	0.75096	1.33	0.74856	1.00	0.71836	1.67	0.61600
25	1.00	0.73494	1.00	0.75096	1.33	0.74856	1.00	0.71836	1.00	0.71141
26	1.00	0.73494	1.00	0.75096	1.00	0.73297	1.00	0.71836	1.00	0.71141
27	1.00	0.73494	1.00	0.75096	1.00	0.73297	1.00	0.71836	1.00	0.71141
28	1.00	0.73494	1.00	0.75096	1.00	0.73297	1.00	0.71836	1.00	0.71141
29	0.67	0.71200	1.00	0.75096	1.00	0.73297	1.00	0.71836	0.00	0.74932
30	0.00	0.68500	1.00	0.75096	0.00	0.91087	0.00	1.14039	0.00	0.74932
31	0.00	0.68500	0.00	0.99411	0.00	0.91087	0.00	1.14039	0.00	0.74932
32	0.00	0.68500	0.00	0.99411	0.00	0.91087	0.00	1.14039	0.00	0.74932
33	0.00	0.68500	0.00	0.99411	0.00	0.91087	0.00	1.14039	0.00	0.74932

All-15 uses 15 measurements for every task. The final three penalty settings make no acquisitions for any test protein. No protein-level confidence interval is assigned to these descriptive means.

Per-protein ranking and selection at the reported examples The following tables give all seven additional metrics for all-15 and the three settings labeled in Figure 2. The full-grid averages and

intervals are in Tables C1.2–C1.5. Each protein row is the mean of its three tasks; rate columns are percentages.

Table C1.9. Per-protein quality with all 15 responses.

Protein	Spearman	Top-1 hit (%)	Top-5 recall (%)	Best in five (%)	Top-1 loss (pKi)	Severe miss (%)	Option regret (pKi)
CHEMBL1821	0.7993	33.33	80.00	66.67	0.01333	0.00	0.00480
CHEMBL1836	0.3952	33.33	53.33	33.33	0.34333	33.33	0.01985
CHEMBL1867	0.4100	0.00	46.67	33.33	0.18000	0.00	0.01885
CHEMBL1875	0.3421	0.00	46.67	66.67	0.26000	33.33	0.00888
CHEMBL1898	0.2262	33.33	26.67	66.67	0.43333	33.33	0.02449
CHEMBL1945	0.8493	33.33	73.33	66.67	0.27333	33.33	0.00000
CHEMBL2049	0.3483	0.00	53.33	33.33	0.46667	0.00	0.09097
CHEMBL229	0.4935	33.33	46.67	66.67	0.57000	66.67	0.00787
CHEMBL249	0.8917	33.33	80.00	100.00	0.09333	0.00	0.04050
CHEMBL3764	0.3962	0.00	46.67	33.33	0.75000	33.33	0.01322
CHEMBL4315	0.4228	66.67	46.67	66.67	0.28000	33.33	0.00354
CHEMBL4561	0.1174	0.00	46.67	33.33	1.51667	100.00	0.02341
CHEMBL4644	0.6545	33.33	93.33	100.00	0.47000	33.33	0.02091
CHEMBL4777	0.5204	33.33	66.67	66.67	0.30667	33.33	0.00419
CHEMBL4792	0.7416	33.33	80.00	100.00	0.31000	33.33	0.00348

Table C1.10. Per-protein quality at c = 0.0015773934 (index 9).

Protein	Spearman	Top-1 hit (%)	Top-5 recall (%)	Best in five (%)	Top-1 loss (pKi)	Severe miss (%)	Option regret (pKi)
CHEMBL1821	0.8112	33.33	73.33	33.33	0.01333	0.00	0.00478
CHEMBL1836	0.4101	33.33	53.33	33.33	0.38000	33.33	0.01879
CHEMBL1867	0.3344	0.00	46.67	33.33	0.21333	0.00	0.02758
CHEMBL1875	0.2644	0.00	40.00	66.67	0.22000	33.33	0.00927
CHEMBL1898	0.2495	33.33	33.33	66.67	0.33333	0.00	0.02329
CHEMBL1945	0.8493	33.33	73.33	100.00	0.68333	66.67	0.00000
CHEMBL2049	0.3678	0.00	60.00	66.67	0.36667	0.00	0.06754
CHEMBL229	0.4100	33.33	40.00	66.67	0.57000	66.67	0.00998
CHEMBL249	0.8917	33.33	80.00	100.00	0.09333	0.00	0.04355
CHEMBL3764	0.3902	0.00	46.67	33.33	0.75000	33.33	0.01170
CHEMBL4315	0.2782	66.67	40.00	66.67	0.28000	33.33	0.00760
CHEMBL4561	0.0346	0.00	40.00	0.00	1.18667	100.00	0.01976
CHEMBL4644	0.6796	66.67	86.67	100.00	0.40333	33.33	0.00958
CHEMBL4777	0.5151	33.33	66.67	66.67	0.67667	66.67	0.00477
CHEMBL4792	0.7416	33.33	80.00	100.00	0.47000	66.67	0.00389

"Option regret" is the same-history option-selection regret defined above. It is distinct from the affinity loss of the molecule with the highest predicted pKi in Q.

Per-protein quality at the next two reported settings Table C1.11 reports c = 0.0025 (index 11), which uses 39.56% fewer measurements than all-15 on average. Table C1.12 reports c $\approx$ 0.00396 (index 13), which uses 52.00% fewer measurements. Individual-protein acquisition counts and prediction errors are given in Tables C1.6–C1.8.

Table C1.11. Per-protein quality at c = 0.0025 (index 11).

Protein	Spearman	Top-1 hit (%)	Top-5 recall (%)	Best in five (%)	Top-1 loss (pKi)	Severe miss (%)	Option regret (pKi)
CHEMBL1821	0.7540	33.33	73.33	33.33	0.18000	33.33	0.00288
CHEMBL1836	0.4196	33.33	53.33	33.33	0.38000	33.33	0.01536
CHEMBL1867	0.4095	0.00	46.67	0.00	0.21333	0.00	0.03140

Protein	Spearman	Top-1 hit (%)	Top-5 recall (%)	Best in five (%)	Top-1 loss (pKi)	Severe miss (%)	Option regret (pKi)
CHEMBL1875	0.0216	0.00	26.67	33.33	0.36000	66.67	0.01522
CHEMBL1898	0.2579	100.00	40.00	100.00	0.00000	0.00	0.02718
CHEMBL1945	0.8457	33.33	73.33	100.00	0.68333	66.67	0.00000
CHEMBL2049	0.3218	0.00	53.33	66.67	0.36667	0.00	0.06629
CHEMBL229	0.3981	33.33	40.00	66.67	0.57000	66.67	0.00877
CHEMBL249	0.8917	33.33	80.00	100.00	0.09333	0.00	0.04355
CHEMBL3764	0.3627	0.00	46.67	33.33	0.75000	33.33	0.01170
CHEMBL4315	0.1252	66.67	33.33	66.67	0.28000	33.33	0.00281
CHEMBL4561	-0.0286	0.00	46.67	0.00	1.18667	100.00	0.02027
CHEMBL4644	0.6796	66.67	86.67	100.00	0.40333	33.33	0.00958
CHEMBL4777	0.4758	33.33	60.00	66.67	0.67667	66.67	0.00582
CHEMBL4792	0.7416	66.67	73.33	100.00	0.27667	33.33	0.01125

Table C1.12. Per-protein quality at c = 0.003962233 (index 13).

Protein	Spearman	Top-1 hit (%)	Top-5 recall (%)	Best in five (%)	Top-1 loss (pKi)	Severe miss (%)	Option regret (pKi)
CHEMBL1821	0.7397	33.33	73.33	66.67	0.18000	33.33	0.01017
CHEMBL1836	0.3886	0.00	53.33	33.33	0.40000	33.33	0.01192
CHEMBL1867	0.4524	0.00	46.67	0.00	0.19333	0.00	0.02182
CHEMBL1875	-0.1570	0.00	20.00	33.33	0.44667	66.67	0.01146
CHEMBL1898	0.2842	100.00	40.00	100.00	0.00000	0.00	0.02430
CHEMBL1945	0.8529	33.33	73.33	100.00	0.68333	66.67	0.00000
CHEMBL2049	0.3218	0.00	53.33	66.67	0.36667	0.00	0.06629
CHEMBL229	0.4422	0.00	53.33	33.33	1.00667	100.00	0.03281
CHEMBL249	0.8952	33.33	80.00	100.00	0.09333	0.00	0.03738
CHEMBL3764	0.3567	0.00	46.67	33.33	0.75000	33.33	0.00487
CHEMBL4315	0.2110	66.67	40.00	66.67	0.28000	33.33	0.00340
CHEMBL4561	-0.0066	0.00	33.33	0.00	1.51667	100.00	0.04886
CHEMBL4644	0.6944	66.67	80.00	100.00	0.06667	0.00	0.01299
CHEMBL4777	0.4841	33.33	66.67	66.67	0.30667	33.33	0.00431
CHEMBL4792	0.7303	33.33	80.00	100.00	0.47000	66.67	0.00588

Higher values are preferable for Spearman, top-1 hit, top-five recall and best in five. Lower values are preferable for top-1 loss, severe-miss rate and option regret.

C.2 ChEMBL: Acquisition and Stopping Comparisons

C.2.1 Which responses to acquire

We compare posterior geometry with random acquisition at each fixed count k = 0–15. Both use the same prepared predictions, prediction-option selection and residual correction, and neither applies the voluntary stopping rule. Random-acquisition MAE averages all 32 predefined orders within each task. Differences are posterior geometry minus random, paired within tasks and averaged equally across test proteins. Figure 3 shows this comparison.

Posterior geometry has lower mean MAE at every count from 1 to 14. At k = 9, the mean MAE is 0.47125 pKi for posterior geometry and 0.51402 pKi for random acquisition. Pointwise intervals are below zero at counts 3–14; the approximate simultaneous band is below zero only at 4 and 5. The band at 5 has an upper endpoint close to zero.

Table C2.1. Same-count prediction errors and paired differences (pKi).

k	Posterior geometry Mean [90% CI]	Random Mean [90% CI]	ΔMAE	Pointwise 90% CI for Δ	Approx. simultaneous 90% band for Δ
0	0.91822 [0.73500, 1.12621]	0.91822 [0.73500, 1.12621]	+2.96e-17	[-9.25e-17, 1.67e-16]	[-0.06006, 0.06006]
1	0.73036 [0.62948, 0.84205]	0.75151 [0.65478, 0.86111]	-0.02116	[-0.07276, 0.03943]	[-0.08122, 0.03890]
2	0.65762 [0.55941, 0.76522]	0.68258 [0.59240, 0.78245]	-0.02496	[-0.07279, 0.03915]	[-0.08502, 0.03509]
3	0.59649 [0.50792, 0.68994]	0.64421 [0.55431, 0.74082]	-0.04772	[-0.06551, -0.02935]	[-0.10777, 0.01234]
4	0.54712 [0.47464, 0.62185]	0.60992 [0.52381, 0.70096]	-0.06280	[-0.08656, -0.04023]	[-0.12286, -0.00274]
5	0.52414 [0.45333, 0.59848]	0.58420 [0.50121, 0.67116]	-0.06006	[-0.09293, -0.02912]	[-0.12012, -0.0000030]
6	0.50630 [0.43802, 0.57721]	0.56086 [0.48058, 0.64486]	-0.05455	[-0.07990, -0.03039]	[-0.11461, 0.00551]
7	0.49536 [0.42548, 0.56753]	0.54424 [0.46634, 0.62483]	-0.04888	[-0.07266, -0.02652]	[-0.10894, 0.01117]
8	0.48237 [0.41516, 0.55029]	0.52806 [0.45324, 0.60553]	-0.04569	[-0.06591, -0.02666]	[-0.10575, 0.01437]
9	0.47125 [0.40513, 0.53874]	0.51402 [0.44224, 0.58842]	-0.04277	[-0.05724, -0.02957]	[-0.10283, 0.01729]
10	0.47302 [0.40643, 0.54169]	0.50210 [0.43223, 0.57479]	-0.02908	[-0.04040, -0.01893]	[-0.08914, 0.03098]
11	0.47187 [0.40571, 0.54147]	0.49206 [0.42359, 0.56333]	-0.02019	[-0.03182, -0.00936]	[-0.08025, 0.03987]
12	0.46576 [0.39897, 0.53629]	0.48427 [0.41640, 0.55504]	-0.01851	[-0.02713, -0.01037]	[-0.07857, 0.04155]
13	0.46373 [0.39594, 0.53604]	0.47804 [0.41046, 0.54879]	-0.01430	[-0.02200, -0.00627]	[-0.07436, 0.04576]
14	0.46620 [0.39906, 0.53725]	0.47136 [0.40456, 0.54158]	-0.00516	[-0.00829, -0.00184]	[-0.06522, 0.05490]
15	0.46495 [0.39886, 0.53439]	0.46495 [0.39886, 0.53439]	+3.70e-18	[-5.92e-17, 7.40e-17]	[-0.06006, 0.06006]

Intervals use 20,000 paired whole-protein resamples with fitted models, settings, tasks and random orders fixed. The simultaneous family contains all 16 counts. At k = 5, its upper endpoint is -0.00000297363028905 pKi. At k = 0 and 15, the methods agree to numerical precision; their very small recorded differences are retained.

Same-count differences for individual proteins Each entry averages the three tasks of one protein after averaging the random orders within each task. Negative values favor posterior geometry. All measurement counts are shown, including the common empty and complete measurement sets.

Table C2.2. Per-protein MAE difference: posterior geometry minus random (pKi).

k	CHEMBL1821	CHEMBL1836	CHEMBL1867	CHEMBL1875	CHEMBL1898
0	0.00000	+8.88e-16	0.00000	-1.67e-16	+5.55e-17
1	-0.04992	-0.11163	-0.06385	-0.05931	+0.04931
2	-0.08016	-0.09351	-0.05725	-0.02384	-0.02848
3	-0.03508	-0.08608	-0.07420	-0.01354	-0.08142
4	-0.00683	+0.00523	-0.03930	-0.01074	-0.03308
5	-0.01540	+0.03572	-0.01605	-0.00058	-0.04240
6	-0.02538	+0.03237	-0.00575	-0.02631	-0.02153
7	-0.04196	+0.01067	-0.01245	-0.04355	-0.02342
8	-0.03938	+0.00268	-0.02717	-0.04709	-0.01212
9	-0.04710	-0.02426	-0.02247	-0.03877	-0.01048
10	-0.04283	-0.00359	-0.01609	-0.02845	-0.00837
11	-0.04277	-0.01235	-0.00015	-0.02110	-0.00415

k	CHEMBL1821	CHEMBL1836	CHEMBL1867	CHEMBL1875	CHEMBL1898
12	-0.05216	-0.01688	-0.00209	-0.01902	-0.00889
13	-0.03997	-0.00506	-0.00734	-0.01112	-0.00167
14	-0.01481	-0.00252	-0.00062	-0.00627	-0.00198
15	0.00000	-2.22e-16	-1.11e-16	0.00000	0.00000

Table C2.3. Per-protein MAE difference: posterior geometry minus random (pKi).

k	CHEMBL1945	CHEMBL2049	CHEMBL229	CHEMBL249	CHEMBL3764
0	-4.44e-16	0.00000	0.00000	-3.33e-16	0.00000
1	-0.07125	-0.15929	-0.02968	-0.05965	+0.38371
2	-0.15966	-0.12106	-0.04886	-0.08016	+0.44070
3	-0.09836	-0.09897	-0.03293	-0.04328	+0.04082
4	-0.16462	-0.15386	-0.00948	-0.11120	-0.11754
5	-0.20359	-0.18506	-0.00828	-0.09396	-0.16386
6	-0.15894	-0.12513	+0.01483	-0.11538	-0.15509
7	-0.17550	-0.08302	+0.02778	-0.11003	-0.11811
8	-0.14616	-0.09006	+0.03069	-0.07861	-0.12072
9	-0.11795	-0.08384	+0.00099	-0.06434	-0.09251
10	-0.09925	-0.05277	-0.00338	-0.04264	-0.04506
11	-0.07950	-0.05007	+0.00244	-0.01112	-0.05350
12	-0.04658	-0.04698	+0.00166	-0.01433	-0.04466
13	-0.03380	-0.03775	-0.00848	-0.01433	-0.03878
14	-0.01884	+0.00060	-0.01177	-0.00556	-0.00900
15	0.00000	0.00000	+5.55e-17	-1.67e-16	0.00000

Same-count differences and development errors

Table C2.4. Per-protein MAE difference: posterior geometry minus random (pKi).

k	CHEMBL4315	CHEMBL4561	CHEMBL4644	CHEMBL4777	CHEMBL4792
0	-1.11e-16	+2.22e-16	+4.44e-16	-2.22e-16	+1.11e-16
1	+0.08541	-0.19736	-0.09702	-0.02421	+0.08736
2	+0.02989	-0.11467	-0.02277	+0.03043	-0.04507
3	+0.01837	-0.04895	-0.10703	-0.04162	-0.01346
4	-0.00977	-0.06327	-0.11080	-0.03625	-0.08046
5	+0.01343	+0.02558	-0.06238	-0.05416	-0.12992
6	-0.02476	-0.00118	-0.07768	-0.05758	-0.07076
7	-0.01779	+0.02906	-0.07480	-0.04735	-0.05276
8	-0.00265	-0.01627	-0.06999	-0.02275	-0.04580
9	-0.01491	-0.01279	-0.04556	-0.02572	-0.04186
10	+0.00134	-0.00465	-0.02891	-0.02658	-0.03502
11	+0.00437	+0.02895	-0.03088	-0.02018	-0.01288
12	+0.00168	+0.01581	-0.01737	-0.01074	-0.01715
13	-0.01637	+0.03556	-0.02178	-0.00547	-0.00817
14	-0.00509	+0.01543	-0.01138	-0.00314	-0.00247
15	-5.55e-17	+4.44e-16	+2.22e-16	+5.55e-17	-1.67e-16

For reference, Table C2.5 gives the development prediction errors at each fixed count. These are held-protein cross-validation predictions from the 57 development proteins. The posterior-geometry curve supplies the fixed-count candidate errors used for the development choices reported in C.2.4.

Table C2.5. Development errors at each fixed acquisition count (pKi).

k	Posterior geometry	Random (32 orders)	ΔMAE
0	0.81262	0.81262	+1.85e-17
1	0.68045	0.69736	-0.01691

k	Posterior geometry	Random (32 orders)	ΔMAE
2	0.62215	0.64988	-0.02773
3	0.58668	0.62052	-0.03384
4	0.56380	0.59991	-0.03610
5	0.54886	0.58352	-0.03466
6	0.53845	0.57045	-0.03200
7	0.52924	0.55933	-0.03009
8	0.52131	0.54916	-0.02785
9	0.51537	0.54076	-0.02539
10	0.51289	0.53289	-0.02001
11	0.50968	0.52626	-0.01658
12	0.50714	0.52012	-0.01297
13	0.50601	0.51467	-0.00866
14	0.50508	0.50942	-0.00434
15	0.50498	0.50498	-1.95e-18

Development means use tasks within environment, environments within protein and equal protein weights. Table C2.5 reports point estimates, not a separate test of the later count selection.

C.2.2 Original adaptive stopping versus a fixed count

At each evaluation penalty c, the fixed-count comparator selects one k from 0–15 using development data and applies it to every test task. The adaptive procedure instead decides when to stop from its acquired responses. Both use posterior geometry and the same prediction selection and correction. Their common evaluation loss is

$$J_c = MAE + ck \quad \text{(C2.1)}$$

Here c is the cost per response in pKi units. The original adaptive procedure also uses λ = c in its stopping calculation. Table C2.6 gives the outcomes; Table C2.7 gives the paired uncertainty. For brevity, the tables denote this loss by J. Every difference is adaptive minus fixed, so negative ΔJ favors adaptive stopping.

Table C2.6. Original adaptive and development-selected fixed-count outcomes.

Index	c (pKi/response)	Fixed k	Adaptive mean k	Fixed MAE	Adaptive MAE	Δk	ΔMAE	ΔJ
0	0	15	13.53	0.46495	0.46668	-1.47	+0.00172	+0.00172
1	0.00025	14	13.04	0.46620	0.46645	-0.96	+0.00025	+0.00001
2	0.00031473135	14	13.00	0.46620	0.46640	-1.00	+0.00020	-0.00011
3	0.0003962233	14	12.89	0.46620	0.46586	-1.11	-0.00034	-0.00078
4	0.00049881558	14	12.76	0.46620	0.46620	-1.24	0.00000	-0.00062
5	0.00062797161	14	12.49	0.46620	0.46695	-1.51	+0.00075	-0.00019
6	0.00079056942	14	12.36	0.46620	0.46722	-1.64	+0.00102	-0.00028
7	0.00099526793	13	11.98	0.46373	0.46802	-1.02	+0.00428	+0.00327
8	0.0012529681	12	11.49	0.46576	0.47023	-0.51	+0.00447	+0.00383
9	0.0015773934	12	10.71	0.46576	0.46817	-1.29	+0.00241	+0.00037
10	0.0019858206	12	9.82	0.46576	0.47542	-2.18	+0.00966	+0.00534
11	0.0025	12	9.07	0.46576	0.47877	-2.93	+0.01301	+0.00568
12	0.0031473135	9	8.09	0.47125	0.48251	-0.91	+0.01126	+0.00840
13	0.003962233	9	7.20	0.47125	0.49110	-1.80	+0.01986	+0.01272
14	0.0049881558	9	6.78	0.47125	0.49865	-2.22	+0.02740	+0.01632
15	0.0062797161	8	5.47	0.48237	0.52027	-2.53	+0.03789	+0.02199
16	0.0079056942	8	4.84	0.48237	0.53231	-3.16	+0.04994	+0.02499
17	0.0099526793	6	4.47	0.50630	0.53197	-1.53	+0.02567	+0.01041
18	0.012529681	5	3.84	0.52414	0.54210	-1.16	+0.01797	+0.00349
19	0.015773934	4	3.29	0.54712	0.56777	-0.71	+0.02064	+0.00943
20	0.019858206	4	2.89	0.54712	0.59600	-1.11	+0.04888	+0.02681
21	0.025	3	2.42	0.59649	0.61336	-0.58	+0.01687	+0.00242
22	0.031473135	3	2.20	0.59649	0.63695	-0.80	+0.04046	+0.01528
23	0.03962233	2	1.87	0.65762	0.64957	-0.13	-0.00805	-0.01333

Index	c (pKi/response)	Fixed k	Adaptive mean k	Fixed MAE	Adaptive MAE	Δk	ΔMAE	ΔJ
24	0.049881558	2	1.36	0.65762	0.71544	-0.64	+0.05782	+0.02568
25	0.062797161	1	1.09	0.73036	0.73064	+0.09	+0.00029	+0.00587
26	0.079056942	1	1.04	0.73036	0.73185	+0.04	+0.00149	+0.00501
27	0.099526793	1	1.00	0.73036	0.73036	0.00	0.00000	0.00000
28	0.12529681	1	0.96	0.73036	0.73714	-0.04	+0.00679	+0.00122
29	0.15773934	0	0.69	0.91822	0.78728	+0.69	-0.13095	-0.02228
30	0.19858206	0	0.33	0.91822	0.82789	+0.33	-0.09033	-0.02414
31	0.25	0	0.00	0.91822	0.91822	0.00	0.00000	0.00000
32	0.35	0	0.00	0.91822	0.91822	0.00	0.00000	0.00000
33	0.5	0	0.00	0.91822	0.91822	0.00	0.00000	0.00000

MAE, ΔMAE and ΔJ are in pKi. Both policies are evaluated at the same cost per response, not constrained to acquire the same count.

Uncertainty for original adaptive stopping Across the 34 penalties, original adaptive stopping has lower mean J at eight, numerically tied means at four, and higher means at 22. No pointwise interval for ΔJ favors adaptive stopping. The approximate 90% simultaneous band includes zero at every penalty. These summaries describe repeated evaluations of the same proteins at different costs.

Table C2.7. Paired adaptive-minus-fixed differences and intervals.

Index	Δk	90% CI for Δk	ΔMAE	90% CI for ΔMAE	ΔJ	90% CI for ΔJ	Approx. 90% simultaneous band for ΔJ
0	-1.47	[-2.56, -0.67]	+0.00172	[-0.00275, 0.00618]	+0.00172	[-0.00275, 0.00618]	[-0.14907, 0.15252]
1	-0.96	[-2.00, -0.13]	+0.00025	[-0.00524, 0.00555]	+0.00001	[-0.00562, 0.00535]	[-0.15078, 0.15081]
2	-1.00	[-2.02, -0.20]	+0.00020	[-0.00528, 0.00550]	-0.00011	[-0.00578, 0.00523]	[-0.15091, 0.15068]
3	-1.11	[-2.13, -0.29]	-0.00034	[-0.00609, 0.00532]	-0.00078	[-0.00668, 0.00491]	[-0.15157, 0.15002]
4	-1.24	[-2.27, -0.42]	0.00000	[-0.00574, 0.00561]	-0.00062	[-0.00660, 0.00504]	[-0.15141, 0.15018]
5	-1.51	[-2.49, -0.71]	+0.00075	[-0.00510, 0.00633]	-0.00019	[-0.00637, 0.00548]	[-0.15099, 0.15060]
6	-1.64	[-2.62, -0.84]	+0.00102	[-0.00499, 0.00675]	-0.00028	[-0.00668, 0.00553]	[-0.15107, 0.15052]
7	-1.02	[-1.98, -0.22]	+0.00428	[-0.00237, 0.01043]	+0.00327	[-0.00317, 0.00928]	[-0.14753, 0.15406]
8	-0.51	[-1.38, 0.24]	+0.00447	[-0.00077, 0.00889]	+0.00383	[-0.00141, 0.00840]	[-0.14697, 0.15462]
9	-1.29	[-2.11, -0.56]	+0.00241	[-0.00246, 0.00658]	+0.00037	[-0.00481, 0.00497]	[-0.15042, 0.15117]
10	-2.18	[-3.09, -1.33]	+0.00966	[0.00173, 0.01773]	+0.00534	[-0.00230, 0.01263]	[-0.14546, 0.15613]
11	-2.93	[-3.73, -2.20]	+0.01301	[0.00339, 0.02295]	+0.00568	[-0.00335, 0.01487]	[-0.14512, 0.15647]
12	-0.91	[-1.62, -0.27]	+0.01126	[0.00506, 0.01816]	+0.00840	[0.00332, 0.01408]	[-0.14240, 0.15919]
13	-1.80	[-2.44, -1.20]	+0.01986	[0.00880, 0.03150]	+0.01272	[0.00276, 0.02337]	[-0.13807, 0.16352]
14	-2.22	[-2.82, -1.64]	+0.02740	[0.01406, 0.04315]	+0.01632	[0.00470, 0.03045]	[-0.13448, 0.16711]
15	-2.53	[-2.93, -2.13]	+0.03789	[0.02296, 0.05255]	+0.02199	[0.00669, 0.03675]	[-0.12881, 0.17278]

Index	Δk	90% CI for Δk	ΔMAE	90% CI for ΔMAE	ΔJ	90% CI for ΔJ	Approx. 90% simultaneous band for ΔJ
16	-3.16	[-3.53, -2.76]	+0.04994	[0.02955, 0.07037]	+0.02499	[0.00486, 0.04515]	[-0.12580, 0.17579]
17	-1.53	[-1.93, -1.13]	+0.02567	[0.01012, 0.04236]	+0.01041	[-0.00466, 0.02635]	[-0.14039, 0.16120]
18	-1.16	[-1.44, -0.84]	+0.01797	[-0.00346, 0.04350]	+0.00349	[-0.01708, 0.02718]	[-0.14731, 0.15428]
19	-0.71	[-0.87, -0.56]	+0.02064	[0.00433, 0.03855]	+0.00943	[-0.00627, 0.02662]	[-0.14137, 0.16022]
20	-1.11	[-1.27, -0.96]	+0.04888	[0.02712, 0.07161]	+0.02681	[0.00496, 0.04975]	[-0.12398, 0.17761]
21	-0.58	[-0.76, -0.38]	+0.01687	[-0.00617, 0.04175]	+0.00242	[-0.01809, 0.02457]	[-0.14837, 0.15322]
22	-0.80	[-0.91, -0.67]	+0.04046	[0.01757, 0.06604]	+0.01528	[-0.00741, 0.04060]	[-0.13552, 0.16607]
23	-0.13	[-0.29, 0.02]	-0.00805	[-0.03702, 0.01449]	-0.01333	[-0.04112, 0.00945]	[-0.16413, 0.13746]
24	-0.64	[-0.80, -0.47]	+0.05782	[0.02632, 0.09296]	+0.02568	[-0.00584, 0.06059]	[-0.12512, 0.17647]
25	+0.09	[0.02, 0.16]	+0.00029	[-0.00449, 0.00447]	+0.00587	[0.00003, 0.01244]	[-0.14492, 0.15666]
26	+0.04	[0.00, 0.09]	+0.00149	[0.00000, 0.00381]	+0.00501	[0.00000, 0.01084]	[-0.14579, 0.15580]
27	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.15079, 0.15079]
28	-0.04	[-0.13, 0.00]	+0.00679	[0.00000, 0.02036]	+0.00122	[0.00000, 0.00366]	[-0.14958, 0.15201]
29	+0.69	[0.51, 0.87]	-0.13095	[-0.29981, 0.00688]	-0.02228	[-0.18555, 0.11350]	[-0.17308, 0.12851]
30	+0.33	[0.16, 0.53]	-0.09033	[-0.25851, 0.02224]	-0.02414	[-0.17442, 0.08154]	[-0.17493, 0.12666]
31	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.15079, 0.15079]
32	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.15079, 0.15079]
33	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.15079, 0.15079]

Δk is the adaptive-minus-fixed difference in the number of molecules in P whose pKi values are acquired. Negative Δk means that adaptive stopping acquires fewer responses. ΔMAE and ΔJ are in pKi. Intervals for Δk, ΔMAE and ΔJ are pointwise; only the final column adjusts across the 34 ΔJ comparisons. Intervals use the original 20,000 paired whole-protein resamples. Numerical ties in the curve summary mean $|\Delta J| \leq 10^{-12}$.

C.2.3 Adaptive stopping with a development-selected penalty

This comparison also lets development data choose the adaptive procedure's stopping penalty. For each evaluation cost c, development selects λ from the same 34 adaptive settings. That λ is then shared across test tasks, while each task's measurements still determine its stopping count. The final error and acquired count are evaluated at c, not at the selected λ. The fixed-count comparator is unchanged.

Mean J is lower for adaptive stopping at nine penalties, tied at five, and higher at 20. Its separate approximate 90% simultaneous band includes zero throughout the grid. Allowing development selection of the stopping penalty therefore does not establish a general cost-inclusive advantage over a fixed count.

Table C2.8. Adaptive outcomes after development selection of λ.

Index	Evaluation c	Selected λ	Fixed k	Adaptive mean k	Fixed MAE	Adaptive MAE	ΔJ
0	0	0.00025	15	13.04	0.46495	0.46645	+0.00149
1	0.00025	0.00025	14	13.04	0.46620	0.46645	+0.00001
2	0.00031473135	0.00025	14	13.04	0.46620	0.46645	-0.00005
3	0.0003962233	0.00025	14	13.04	0.46620	0.46645	-0.00013
4	0.00049881558	0.00025	14	13.04	0.46620	0.46645	-0.00023
5	0.00062797161	0.00031473135	14	13.00	0.46620	0.46640	-0.00043
6	0.00079056942	0.00031473135	14	13.00	0.46620	0.46640	-0.00059
7	0.00099526793	0.00031473135	13	13.00	0.46373	0.46640	+0.00267
8	0.0012529681	0.00031473135	12	13.00	0.46576	0.46640	+0.00189
9	0.0015773934	0.00031473135	12	13.00	0.46576	0.46640	+0.00222
10	0.0019858206	0.0012529681	12	11.49	0.46576	0.47023	+0.00345
11	0.0025	0.0019858206	12	9.82	0.46576	0.47542	+0.00422
12	0.0031473135	0.0019858206	9	9.82	0.47125	0.47542	+0.00676
13	0.003962233	0.0031473135	9	8.09	0.47125	0.48251	+0.00765
14	0.0049881558	0.0031473135	9	8.09	0.47125	0.48251	+0.00672
15	0.0062797161	0.0031473135	8	8.09	0.48237	0.48251	+0.00070
16	0.0079056942	0.0049881558	8	6.78	0.48237	0.49865	+0.00662
17	0.0099526793	0.0062797161	6	5.47	0.50630	0.52027	+0.00865
18	0.012529681	0.0062797161	5	5.47	0.52414	0.52027	+0.00198
19	0.015773934	0.0079056942	4	4.84	0.54712	0.53231	-0.00149
20	0.019858206	0.019858206	4	2.89	0.54712	0.59600	+0.02681
21	0.025	0.019858206	3	2.89	0.59649	0.59600	-0.00327
22	0.031473135	0.019858206	3	2.89	0.59649	0.59600	-0.00399
23	0.03962233	0.031473135	2	2.20	0.65762	0.63695	-0.01275
24	0.049881558	0.049881558	2	1.36	0.65762	0.71544	+0.02568
25	0.062797161	0.049881558	1	1.36	0.73036	0.71544	+0.00742
26	0.079056942	0.062797161	1	1.09	0.73036	0.73064	+0.00732
27	0.099526793	0.15773934	1	0.69	0.73036	0.78728	+0.02596
28	0.12529681	0.19858206	1	0.33	0.73036	0.82789	+0.01401
29	0.15773934	0.25	0	0.00	0.91822	0.91822	0.00000
30	0.19858206	0.25	0	0.00	0.91822	0.91822	0.00000
31	0.25	0.25	0	0.00	0.91822	0.91822	0.00000
32	0.35	0.35	0	0.00	0.91822	0.91822	0.00000
33	0.5	0.35	0	0.00	0.91822	0.91822	0.00000

Both c and λ are in pKi per response. MAE and ΔJ are in pKi. This is selection among the original adaptive settings.

Uncertainty after development selection of λ The intervals below use the same paired test tasks as Table C2.8. No pointwise ΔJ interval favors adaptive stopping, and the simultaneous band does not establish a difference at any penalty. The band is constructed for this adaptive comparison alone, separately from Table C2.7.

Table C2.9. Paired adaptive-minus-fixed differences and intervals.

Index	Δk	90% CI for Δk	ΔMAE	90% CI for ΔMAE	ΔJ	90% CI for ΔJ	Approx. 90% simultaneous band for ΔJ
0	-1.96	[-3.00, -1.13]	+0.00149	[-0.00308, 0.00612]	+0.00149	[-0.00308, 0.00612]	[-0.06722, 0.07020]
1	-0.96	[-2.00, -0.13]	+0.00025	[-0.00524, 0.00555]	+0.00001	[-0.00562, 0.00535]	[-0.06870, 0.06872]
2	-0.96	[-2.00, -0.13]	+0.00025	[-0.00524, 0.00555]	-0.00005	[-0.00572, 0.00529]	[-0.06876, 0.06866]
3	-0.96	[-2.00, -0.13]	+0.00025	[-0.00524, 0.00555]	-0.00013	[-0.00583, 0.00523]	[-0.06884, 0.06858]

Index	Δk	90% CI for Δk	ΔMAE	90% CI for ΔMAE	ΔJ	90% CI for ΔJ	Approx. 90% simultaneous band for ΔJ
4	-0.96	[-2.00, -0.13]	+0.00025	[-0.00524, 0.00555]	-0.00023	[-0.00598, 0.00515]	[-0.06894, 0.06848]
5	-1.00	[-2.02, -0.20]	+0.00020	[-0.00528, 0.00550]	-0.00043	[-0.00625, 0.00498]	[-0.06914, 0.06828]
6	-1.00	[-2.02, -0.20]	+0.00020	[-0.00528, 0.00550]	-0.00059	[-0.00651, 0.00486]	[-0.06930, 0.06812]
7	0.00	[-1.02, 0.80]	+0.00267	[-0.00415, 0.00880]	+0.00267	[-0.00393, 0.00877]	[-0.06604, 0.07138]
8	+1.00	[-0.02, 1.80]	+0.00064	[-0.00321, 0.00439]	+0.00189	[-0.00196, 0.00584]	[-0.06682, 0.07060]
9	+1.00	[-0.02, 1.80]	+0.00064	[-0.00321, 0.00439]	+0.00222	[-0.00172, 0.00626]	[-0.06649, 0.07093]
10	-0.51	[-1.38, 0.24]	+0.00447	[-0.00077, 0.00889]	+0.00345	[-0.00189, 0.00821]	[-0.06526, 0.07216]
11	-2.18	[-3.09, -1.33]	+0.00966	[0.00173, 0.01773]	+0.00422	[-0.00338, 0.01136]	[-0.06449, 0.07293]
12	+0.82	[-0.09, 1.67]	+0.00418	[-0.00428, 0.01270]	+0.00676	[-0.00045, 0.01391]	[-0.06195, 0.07547]
13	-0.91	[-1.62, -0.27]	+0.01126	[0.00506, 0.01816]	+0.00765	[0.00278, 0.01319]	[-0.06106, 0.07636]
14	-0.91	[-1.62, -0.27]	+0.01126	[0.00506, 0.01816]	+0.00672	[0.00199, 0.01211]	[-0.06199, 0.07543]
15	+0.09	[-0.62, 0.73]	+0.00014	[-0.00808, 0.00814]	+0.00070	[-0.00696, 0.00824]	[-0.06801, 0.06941]
16	-1.22	[-1.82, -0.64]	+0.01628	[0.00207, 0.03220]	+0.00662	[-0.00549, 0.02012]	[-0.06209, 0.07533]
17	-0.53	[-0.93, -0.13]	+0.01396	[0.00222, 0.02570]	+0.00865	[-0.00248, 0.01966]	[-0.06006, 0.07736]
18	+0.47	[0.07, 0.87]	-0.00387	[-0.01748, 0.00994]	+0.00198	[-0.01112, 0.01506]	[-0.06673, 0.07069]
19	+0.84	[0.47, 1.24]	-0.01481	[-0.03345, 0.00328]	-0.00149	[-0.01603, 0.01252]	[-0.07020, 0.06722]
20	-1.11	[-1.27, -0.96]	+0.04888	[0.02712, 0.07161]	+0.02681	[0.00496, 0.04975]	[-0.04190, 0.09552]
21	-0.11	[-0.27, 0.04]	-0.00049	[-0.02219, 0.01677]	-0.00327	[-0.02192, 0.01179]	[-0.07198, 0.06544]
22	-0.11	[-0.27, 0.04]	-0.00049	[-0.02219, 0.01677]	-0.00399	[-0.02198, 0.01068]	[-0.07270, 0.06472]
23	+0.20	[0.09, 0.33]	-0.02067	[-0.04767, -0.00072]	-0.01275	[-0.03892, 0.00637]	[-0.08146, 0.05596]
24	-0.64	[-0.80, -0.47]	+0.05782	[0.02632, 0.09296]	+0.02568	[-0.00584, 0.06059]	[-0.04303, 0.09439]
25	+0.36	[0.20, 0.53]	-0.01491	[-0.03402, 0.00152]	+0.00742	[-0.00877, 0.02338]	[-0.06129, 0.07613]
26	+0.09	[0.02, 0.16]	+0.00029	[-0.00449, 0.00447]	+0.00732	[0.00112, 0.01453]	[-0.06139, 0.07602]
27	-0.31	[-0.49, -0.13]	+0.05692	[0.01361, 0.10655]	+0.02596	[-0.00697, 0.06333]	[-0.04275, 0.09467]
28	-0.67	[-0.84, -0.47]	+0.09754	[0.02296, 0.17646]	+0.01401	[-0.05275, 0.08380]	[-0.05470, 0.08272]
29	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.06871, 0.06871]
30	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.06871, 0.06871]
31	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.06871, 0.06871]
32	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.06871, 0.06871]
33	0.00	[0.00, 0.00]	0.00000	[0.00000, 0.00000]	0.00000	[0.00000, 0.00000]	[-0.06871, 0.06871]

Intervals and units are as in Table C2.7. Models, development-selected penalties, fixed counts and task assignments remain fixed during resampling. The change in controller penalty does not replace c in the evaluation loss.

C.2.4 Settings selected on development data

Table C2.10 gives every selected fixed count and adaptive penalty. Each choice minimizes development MAE plus c times the acquired count, using equal protein weights. The fixed candidates are the 16 counts in Table C2.5; the adaptive candidates are the original 34 settings. Exact ties use fewer mean acquisitions and then the smaller count or penalty index (Appendix B.3.4).

The two J columns are scores on the development data used for selection. They are not independent test results. Test proteins do not enter these choices, and no new setting is selected from the results in Tables C2.6–C2.9.

Table C2.10. Full development-selection mapping and development scores.

Cost index	Evaluation c	Fixed k	Selected λ index	Selected λ	Fixed dev. J	Adaptive dev. J	Adaptive dev. mean k
0	0	15	1	0.00025	0.50498	0.50732	13.11
1	0.00025	14	1	0.00025	0.50858	0.51060	13.11
2	0.00031473135	14	1	0.00025	0.50948	0.51145	13.11
3	0.0003962233	14	1	0.00025	0.51063	0.51251	13.11
4	0.00049881558	14	1	0.00025	0.51206	0.51386	13.11
5	0.00062797161	14	2	0.00031473135	0.51387	0.51554	12.99
6	0.00079056942	14	2	0.00031473135	0.51615	0.51766	12.99
7	0.00099526793	13	2	0.00031473135	0.51895	0.52032	12.99
8	0.0012529681	12	2	0.00031473135	0.52218	0.52366	12.99
9	0.0015773934	12	2	0.00031473135	0.52607	0.52788	12.99
10	0.0019858206	12	8	0.0012529681	0.53097	0.53256	10.78
11	0.0025	12	10	0.0019858206	0.53714	0.53762	9.40
12	0.0031473135	9	10	0.0019858206	0.54370	0.54371	9.40
13	0.003962233	9	12	0.0031473135	0.55103	0.55120	7.95
14	0.0049881558	9	12	0.0031473135	0.56027	0.55935	7.95
15	0.0062797161	8	12	0.0031473135	0.57155	0.56962	7.95
16	0.0079056942	8	14	0.0049881558	0.58456	0.58193	6.50
17	0.0099526793	6	15	0.0062797161	0.59817	0.59471	5.71
18	0.012529681	5	15	0.0062797161	0.61151	0.60943	5.71
19	0.015773934	4	16	0.0079056942	0.62690	0.62661	5.03
20	0.019858206	4	20	0.019858206	0.64324	0.64241	2.86
21	0.025	3	20	0.019858206	0.66168	0.65713	2.86
22	0.031473135	3	20	0.019858206	0.68110	0.67567	2.86
23	0.03962233	2	22	0.031473135	0.70139	0.69423	2.16
24	0.049881558	2	24	0.049881558	0.72191	0.71322	1.42
25	0.062797161	1	24	0.049881558	0.74325	0.73150	1.42
26	0.079056942	1	25	0.062797161	0.75951	0.75215	1.16
27	0.099526793	1	29	0.15773934	0.77998	0.76875	0.65
28	0.12529681	1	30	0.19858206	0.80575	0.78390	0.28
29	0.15773934	0	31	0.25	0.81262	0.78752	0.09
30	0.19858206	0	31	0.25	0.81262	0.79134	0.09
31	0.25	0	31	0.25	0.81262	0.79615	0.09
32	0.35	0	32	0.35	0.81262	0.80185	0.04
33	0.5	0	32	0.35	0.81262	0.80770	0.04

Selected λ indices refer to the original 0–33 grid in Table C1.1. The selected count or penalty is common to all test tasks at that c. The original adaptive comparison in C.2.2 uses λ = c instead of this selected λ.

C.2.5 Per-protein stopping comparisons at the reported examples

These tables show how the two adaptive comparisons differ within each test protein at the three settings labeled in Figure 2. Every entry is adaptive minus the same development-selected fixed-

count policy, averaged over that protein's three tasks. Negative Δk means fewer measurements; negative ΔMAE or ΔJ means lower error or lower cost-inclusive loss.

Table C2.11. Per-protein stopping differences at c = 0.0015773934 (index 9); fixed k = 12.

Test protein		Original adaptive: λ = c		Adaptive: development-selected λ		
Protein	Δk	ΔMAE	ΔJ	Δk	ΔMAE	ΔJ
CHEMBL1821	+1.67	+0.01197	+0.01460	+3.00	+0.01506	+0.01979
CHEMBL1836	+0.33	+0.00552	+0.00605	+3.00	+0.00156	+0.00629
CHEMBL1867	0.00	+0.00244	+0.00244	+1.67	-0.00271	-0.00009
CHEMBL1875	-2.33	-0.00759	-0.01128	+1.00	-0.00481	-0.00323
CHEMBL1898	-1.33	+0.00052	-0.00158	+1.67	+0.00302	+0.00565
CHEMBL1945	-1.33	+0.00372	+0.00162	+1.33	+0.00088	+0.00298
CHEMBL2049	-6.33	+0.00255	-0.00744	-6.33	+0.00255	-0.00744
CHEMBL229	-3.00	+0.01620	+0.01147	-0.67	+0.00376	+0.00271
CHEMBL249	-0.67	+0.00870	+0.00765	+1.33	-0.00519	-0.00309
CHEMBL3764	0.00	+0.01241	+0.01241	+2.00	+0.01868	+0.02183
CHEMBL4315	-1.67	+0.01074	+0.00811	+0.33	+0.00064	+0.00116
CHEMBL4561	-2.00	-0.03076	-0.03392	+2.67	-0.02303	-0.01882
CHEMBL4644	-2.67	+0.00114	-0.00307	+0.33	+0.00249	+0.00301
CHEMBL4777	-0.33	-0.00027	-0.00080	+2.00	-0.00487	-0.00172
CHEMBL4792	+0.33	-0.00117	-0.00064	+1.67	+0.00156	+0.00419

Table C2.12. Per-protein stopping differences at c = 0.0025 (index 11); fixed k = 12.

Test protein		Original adaptive: λ = c		Adaptive: development-selected λ		
Protein	Δk	ΔMAE	ΔJ	Δk	ΔMAE	ΔJ
CHEMBL1821	-2.33	+0.05506	+0.04922	+1.67	+0.01197	+0.01614
CHEMBL1836	-1.00	+0.01370	+0.01120	-1.00	+0.01370	+0.01120
CHEMBL1867	-2.33	-0.00034	-0.00617	-0.33	-0.00026	-0.00109
CHEMBL1875	-3.67	+0.01342	+0.00425	-3.67	+0.01342	+0.00425
CHEMBL1898	-4.00	-0.00062	-0.01062	-2.33	+0.00252	-0.00331
CHEMBL1945	-2.00	-0.00208	-0.00708	-2.00	-0.00208	-0.00708
CHEMBL2049	-8.00	+0.05664	+0.03664	-7.67	+0.05318	+0.03402
CHEMBL229	-3.67	+0.01838	+0.00921	-3.00	+0.01620	+0.00870
CHEMBL249	-0.67	+0.00870	+0.00703	-0.67	+0.00870	+0.00703
CHEMBL3764	-1.33	+0.01225	+0.00892	-0.33	+0.01235	+0.01152
CHEMBL4315	-5.00	+0.04435	+0.03185	-3.67	+0.03888	+0.02971
CHEMBL4561	-3.00	-0.03601	-0.04351	-3.00	-0.03601	-0.04351
CHEMBL4644	-3.00	+0.00051	-0.00699	-2.67	+0.00114	-0.00553
CHEMBL4777	-3.00	-0.00001	-0.00751	-3.00	-0.00001	-0.00751
CHEMBL4792	-1.00	+0.01121	+0.00871	-1.00	+0.01121	+0.00871

Δk is the adaptive-minus-fixed difference in the number of molecules in P whose pKi values are acquired. Negative Δk means that adaptive stopping acquires fewer responses. ΔMAE and ΔJ are in pKi. These descriptive protein means do not have separate confidence intervals. The corresponding whole-protein uncertainty is reported in Tables C2.7 and C2.9.

Per-protein stopping at the 52% measurement-reduction example

Table C2.13. Per-protein stopping differences at c = 0.003962233 (index 13); fixed k = 9.

Test protein		Original adaptive: λ = c		Adaptive: development-selected λ		
Protein	Δk	ΔMAE	ΔJ	Δk	ΔMAE	ΔJ
CHEMBL1821	-1.33	+0.07577	+0.07048	-0.33	+0.04076	+0.03944
CHEMBL1836	-0.67	+0.00698	+0.00434	+1.00	-0.00536	-0.00140
CHEMBL1867	-1.33	-0.00292	-0.00820	0.00	+0.00149	+0.00149
CHEMBL1875	-3.33	+0.04249	+0.02928	-2.33	+0.01399	+0.00475

Test protein	Original adaptive: $\lambda = c$			Adaptive: development-selected λ		
Protein	Δk	ΔMAE	ΔJ	Δk	ΔMAE	ΔJ
CHEMBL1898	-2.00	-0.01670	-0.02462	-1.33	+0.00026	-0.00503
CHEMBL1945	-0.33	+0.00601	+0.00469	0.00	+0.00025	+0.00025
CHEMBL2049	-5.00	+0.04827	+0.02845	-5.00	+0.04827	+0.02845
CHEMBL229	-3.33	+0.01611	+0.00290	-3.00	+0.01235	+0.00047
CHEMBL249	+0.33	+0.00497	+0.00629	+1.00	+0.00119	+0.00515
CHEMBL3764	0.00	-0.00601	-0.00601	+0.67	+0.00959	+0.01223
CHEMBL4315	-2.67	+0.02244	+0.01187	-2.67	+0.02244	+0.01187
CHEMBL4561	-2.00	+0.05405	+0.04613	-0.67	+0.02258	+0.01994
CHEMBL4644	-3.33	+0.05006	+0.03685	-0.33	+0.00019	-0.00113
CHEMBL4777	-1.33	-0.00465	-0.00993	0.00	0.00000	0.00000
CHEMBL4792	-0.67	+0.00097	-0.00168	-0.67	+0.00097	-0.00168

The 52.00% reduction labeled in Figure 2 compares original adaptive stopping with all-15. At this cost, development instead selects a fixed count of nine. Relative to that fixed-count policy, original adaptive stopping acquires 1.80 fewer responses on average, with an MAE increase of 0.01986 pKi and a cost-inclusive-loss increase of 0.01272 pKi. The mean differences therefore answer a different question from measurement savings relative to all-15.

Taken together, the same-count results show lower mean prediction error for posterior geometry than for random acquisition at counts 1–14. Neither stopping comparison establishes a general cost-inclusive advantage over a development-selected fixed count. Section 4.5 and Appendix C.6 examine mechanisms behind the different measurement-selection and stopping patterns in ChEMBL and HELP. The separate valuation-rule comparison belongs to Appendix C.5.

C.3 HELP: Full Results and Comparisons

This section gives the detailed HELP results behind Section 4.4. It first shows how measurement savings change prediction error, then compares our procedure with the HELP latency predictor proposed by Lee et al. (2021), evaluated under the measurement protocol in Appendix B.2.5. It then separates measurement choice from stopping and reports the calculation time needed by each procedure. Additional analyses assess prediction quality through architecture rankings and the size of large relative errors.

Each test task asks for latencies of 42 architectures after acquiring responses from a separate pool of 42 architectures on the same device. Our procedure uses five fixed source predictors. Settings are chosen using six development devices; results here use four test devices, with three tasks per device. We average tasks within each device and give the four devices equal weight. Appendix B.2–B.4 specifies preparation, task construction and scoring.

Throughout, k is the number of acquired responses, c is the evaluation penalty per response in log-MAE units, and log-MAE is the mean absolute difference between natural-log latencies. It is not the logarithm of an MAE. All-42 acquires the full pool and applies our final prediction rule; no acquisition applies that rule before any current-device response is obtained. HELP10 denotes the HELP baseline with ten acquired responses, including its ten reference responses.

Paired differences compare the same tasks before averaging. Unless stated otherwise, brackets are 90% pointwise confidence intervals from the existing whole-device bootstrap. Simultaneous bands are labelled separately and cover only the stated full curve. The fitted models, selected settings and tasks remain fixed, as described in Appendix B.4.5. Timing tables report means without confidence intervals.

C.3.1 MEASUREMENT SAVINGS AND PREDICTION ERROR

Table C3.1. References and the three adaptive settings labeled in Figure 4.

Procedure	Mean k	Saved vs 42 (%)	log-MAE	Relative MAE (%)
No acquisition	0.00	100.00	0.48348	63.25
All-42	42.00	0.00	0.12218	13.14
Adaptive, c ≈ 0.000914	7.08	83.13	0.15012	16.77
Adaptive, c ≈ 0.002087	5.08	87.90	0.15309	17.06
Adaptive, c ≈ 0.003153	4.58	89.09	0.15404	17.18
HELP10	10.00	76.19	0.11315	22.15

Savings use 42 measurements as the denominator for every row. The three adaptive settings illustrate the full curve; they are not a newly selected deployment policy.

Complete measurement–error curve Table C3.2 retains all 26 adaptive settings. The index is the position in the HELP cost grid of Table B.10 and is used to connect the following tables. Higher costs can end at the same outcome: indices 22–25 all acquire no responses. At zero cost the procedure acquires 21 responses on average, so its result is distinct from all-42.

Table C3.2. All adaptive costs, measurement counts and prediction errors.

Index	Cost c	Mean k [90% interval]	Saved (%)	log-MAE [90% interval]	Δlog-MAE vs all-42
0	0	21.00 [10.83, 31.48]	50.00	0.14657 [0.05693, 0.25001]	+0.02439
1	5.0897e-05	17.50 [8.00, 27.00]	58.33	0.14727 [0.05726, 0.25063]	+0.02508
2	7.68957e-05	16.00 [8.00, 23.83]	61.90	0.14741 [0.05740, 0.25067]	+0.02522
3	0.000116175	13.67 [7.83, 19.50]	67.46	0.14843 [0.05945, 0.25062]	+0.02624
4	0.000175518	12.50 [7.00, 18.00]	70.24	0.14861 [0.05983, 0.25053]	+0.02643
5	0.000265174	10.50 [6.33, 14.33]	75.00	0.14908 [0.06155, 0.24978]	+0.02690
6	0.000400627	8.75 [5.79, 11.67]	79.17	0.14760 [0.06101, 0.24839]	+0.02542
7	0.000605271	7.83 [5.25, 10.25]	81.35	0.15009 [0.06228, 0.25051]	+0.02790
8	0.000914449	7.08 [5.00, 8.92]	83.13	0.15012 [0.06375, 0.25041]	+0.02793
9	0.00138156	6.42 [4.83, 8.00]	84.72	0.15185 [0.06562, 0.25083]	+0.02967
10	0.00208727	5.08 [3.92, 6.25]	87.90	0.15309 [0.06716, 0.25129]	+0.03091
11	0.00315347	4.58 [3.58, 5.67]	89.09	0.15404 [0.06795, 0.25166]	+0.03185
12	0.00476429	3.92 [3.58, 4.33]	90.67	0.15650 [0.07254, 0.25201]	+0.03431
13	0.00719793	3.50 [3.33, 3.67]	91.67	0.16205 [0.07831, 0.25811]	+0.03986
14	0.0108747	3.08 [2.67, 3.42]	92.66	0.16090 [0.09340, 0.23938]	+0.03871
15	0.0164296	2.42 [2.33, 2.52]	94.25	0.16143 [0.10483, 0.22676]	+0.03925
16	0.024822	2.17 [2.00, 2.33]	94.84	0.18181 [0.10860, 0.25628]	+0.05962
17	0.0375013	1.58 [1.33, 1.83]	96.23	0.21881 [0.13039, 0.31723]	+0.09663

Index	Cost c	Mean k [90% interval]	Saved (%)	log-MAE [90% interval]	Δlog-MAE vs all-42
18	0.0566572	1.33 [1.17, 1.50]	96.83	0.24531 [0.13575, 0.37328]	+0.12313
19	0.0855983	1.17 [1.00, 1.33]	97.22	0.25315 [0.14858, 0.37354]	+0.13097
20	0.129323	1.08 [1.00, 1.19]	97.42	0.27346 [0.15914, 0.40400]	+0.15127
21	0.195382	0.75 [0.67, 0.85]	98.21	0.33471 [0.20280, 0.47608]	+0.21252
22	0.295185	0.00 [0.00, 0.00]	100.00	0.48348 [0.26185, 0.69696]	+0.36130
23	0.445968	0.00 [0.00, 0.00]	100.00	0.48348 [0.26185, 0.69696]	+0.36130
24	0.673772	0.00 [0.00, 0.00]	100.00	0.48348 [0.26185, 0.69696]	+0.36130
25	1.01794	0.00 [0.00, 0.00]	100.00	0.48348 [0.26185, 0.69696]	+0.36130

Positive Δlog-MAE denotes higher error than all-42. The intervals in this table describe each mean; paired-difference intervals are in Table C3.3. The full precision of each cost is given in Table B.10.

Paired changes from the two reference predictions The error reduction from no acquisition and the error increase relative to all-42 answer different questions. Table C3.3 reports both. Each difference subtracts its stated reference, and the simultaneous band covers all 26 costs for that reference separately.

Table C3.3. Adaptive minus all-42 and adaptive minus no acquisition.

Index	Δ vs all-42 [pointwise 90%]	All-42 contrast simultaneous 90%	Δ vs no acquisition [pointwise 90%]	No-acquisition contrast simultaneous 90%
0	+0.02439 [0.00120, 0.05306]	[-0.18165, 0.23043]	-0.33691 [-0.52554, -0.15204]	[-0.54515, -0.12867]
1	+0.02508 [0.00157, 0.05368]	[-0.18096, 0.23113]	-0.33622 [-0.52470, -0.15133]	[-0.54446, -0.12797]
2	+0.02522 [0.00184, 0.05370]	[-0.18082, 0.23127]	-0.33608 [-0.52438, -0.15133]	[-0.54432, -0.12783]
3	+0.02624 [0.00413, 0.05357]	[-0.17980, 0.23229]	-0.33506 [-0.52276, -0.15077]	[-0.54330, -0.12681]
4	+0.02643 [0.00439, 0.05366]	[-0.17961, 0.23247]	-0.33487 [-0.52225, -0.15087]	[-0.54312, -0.12663]
5	+0.02690 [0.00425, 0.05478]	[-0.17915, 0.23294]	-0.33440 [-0.52065, -0.15150]	[-0.54265, -0.12616]
6	+0.02542 [0.00228, 0.05347]	[-0.18063, 0.23146]	-0.33588 [-0.52252, -0.15297]	[-0.54413, -0.12764]
7	+0.02790 [0.00484, 0.05573]	[-0.17814, 0.23395]	-0.33340 [-0.51928, -0.15062]	[-0.54164, -0.12515]
8	+0.02793 [0.00628, 0.05456]	[-0.17811, 0.23398]	-0.33337 [-0.52104, -0.14898]	[-0.54161, -0.12512]
9	+0.02967 [0.00706, 0.05705]	[-0.17637, 0.23571]	-0.33163 [-0.51731, -0.14882]	[-0.53987, -0.12339]
10	+0.03091 [0.00824, 0.05835]	[-0.17514, 0.23695]	-0.33039 [-0.51537, -0.14804]	[-0.53864, -0.12215]
11	+0.03185 [0.00828, 0.05968]	[-0.17419, 0.23790]	-0.32945 [-0.51340, -0.14787]	[-0.53769, -0.12120]
12	+0.03431 [0.01098, 0.06177]	[-0.17173, 0.24036]	-0.32699 [-0.51026, -0.14574]	[-0.53523, -0.11874]
13	+0.03986 [0.01262, 0.06993]	[-0.16618, 0.24591]	-0.32144 [-0.50276, -0.14299]	[-0.52968, -0.11319]
14	+0.03871 [0.01560, 0.06417]	[-0.16733, 0.24476]	-0.32259 [-0.50218, -0.14299]	[-0.53083, -0.11434]

Index	Δ vs all-42 [pointwise 90%]	All-42 contrast simultaneous 90%	Δ vs no acquisition [pointwise 90%]	No-acquisition contrast simultaneous 90%
15	+0.03925 [0.02077, 0.05772]	[-0.16680, 0.24529]	-0.32206 [-0.50962, -0.13449]	[-0.53030, -0.11381]
16	+0.05962 [0.04076, 0.07754]	[-0.14642, 0.26567]	-0.30168 [-0.50867, -0.09407]	[-0.50992, -0.09343]
17	+0.09663 [0.06191, 0.13602]	[-0.10941, 0.30267]	-0.26467 [-0.43780, -0.09182]	[-0.47291, -0.05643]
18	+0.12313 [0.06369, 0.19137]	[-0.08291, 0.32917]	-0.23817 [-0.43512, -0.04240]	[-0.44642, -0.02993]
19	+0.13097 [0.06472, 0.20184]	[-0.07508, 0.33701]	-0.23033 [-0.41827, -0.04240]	[-0.43858, -0.02209]
20	+0.15127 [0.09838, 0.20567]	[-0.05477, 0.35732]	-0.21003 [-0.41827, -0.00178]	[-0.41827, -0.00178]
21	+0.21252 [0.12375, 0.31741]	[0.00648, 0.41857]	-0.14878 [-0.28707, -0.01078]	[-0.35702, 0.05947]
22	+0.36130 [0.15526, 0.56734]	[0.15526, 0.56734]	+0.00000 [0.00000, 0.00000]	[-0.20824, 0.20824]
23	+0.36130 [0.15526, 0.56734]	[0.15526, 0.56734]	+0.00000 [0.00000, 0.00000]	[-0.20824, 0.20824]
24	+0.36130 [0.15526, 0.56734]	[0.15526, 0.56734]	+0.00000 [0.00000, 0.00000]	[-0.20824, 0.20824]
25	+0.36130 [0.15526, 0.56734]	[0.15526, 0.56734]	+0.00000 [0.00000, 0.00000]	[-0.20824, 0.20824]

Each cell with two lines gives the mean paired difference and its pointwise interval. Simultaneous intervals are shown without repeating the mean. Zero displayed at the no-acquisition endpoint is the same output, not a missing result.

Results for each test device Device-specific results make the average measurement count interpretable. In Table C3.4, every cell reports mean acquired count / log-MAE for the device's three tasks. The identifiers are those in Table B.4: titan_rtx_1, gold_6226, pixel2 and raspi4, respectively. Costs retain the indices in Table C3.2.

Table C3.4. Complete device-level adaptive curve and references.

Cost index or reference	Titan RTX	Xeon Gold 6226	Pixel 2	Raspberry Pi 4
0	42.00 / 0.03921	20.33 / 0.06283	18.00 / 0.13243	3.67 / 0.35182
1	33.67 / 0.04055	20.33 / 0.06283	12.33 / 0.13386	3.67 / 0.35182
2	27.67 / 0.04112	20.33 / 0.06283	12.33 / 0.13386	3.67 / 0.35182
3	22.33 / 0.04408	16.67 / 0.06458	12.00 / 0.13323	3.67 / 0.35182
4	20.67 / 0.04502	15.33 / 0.06477	10.33 / 0.13284	3.67 / 0.35182
5	15.33 / 0.04814	14.00 / 0.06602	9.00 / 0.13033	3.67 / 0.35182
6	11.67 / 0.04517	11.67 / 0.06629	8.00 / 0.12711	3.67 / 0.35182
7	10.00 / 0.05041	11.00 / 0.06623	6.67 / 0.13189	3.67 / 0.35182
8	8.67 / 0.04725	9.67 / 0.06924	6.33 / 0.13216	3.67 / 0.35182
9	7.00 / 0.05388	9.00 / 0.06954	6.00 / 0.13218	3.67 / 0.35182
10	5.67 / 0.05726	7.00 / 0.07045	4.00 / 0.13283	3.67 / 0.35182
11	5.00 / 0.06071	6.33 / 0.07036	3.33 / 0.13326	3.67 / 0.35182
12	4.33 / 0.06629	4.33 / 0.07463	3.33 / 0.13326	3.67 / 0.35182
13	3.67 / 0.07288	3.67 / 0.08012	3.33 / 0.13326	3.33 / 0.36193
14	2.67 / 0.10239	3.67 / 0.08012	3.33 / 0.13326	2.67 / 0.32782
15	2.33 / 0.11331	2.33 / 0.09447	2.67 / 0.13590	2.33 / 0.30203
16	2.33 / 0.11331	2.33 / 0.09447	2.00 / 0.21552	2.00 / 0.30393
17	1.67 / 0.14956	2.00 / 0.10911	1.33 / 0.19422	1.33 / 0.42237
18	1.67 / 0.14956	1.33 / 0.11626	1.33 / 0.19422	1.00 / 0.52122
19	1.33 / 0.17680	1.00 / 0.12037	1.33 / 0.19422	1.00 / 0.52122
20	1.33 / 0.17680	1.00 / 0.12037	1.00 / 0.27544	1.00 / 0.52122
21	1.00 / 0.35250	0.67 / 0.20766	0.67 / 0.18821	0.67 / 0.59045
22	0.00 / 0.75795	0.00 / 0.37576	0.00 / 0.12359	0.00 / 0.67663

Cost index or reference	Titan RTX	Xeon Gold 6226	Pixel 2	Raspberry Pi 4
23	0.00 / 0.75795	0.00 / 0.37576	0.00 / 0.12359	0.00 / 0.67663
24	0.00 / 0.75795	0.00 / 0.37576	0.00 / 0.12359	0.00 / 0.67663
25	0.00 / 0.75795	0.00 / 0.37576	0.00 / 0.12359	0.00 / 0.67663
No acquisition	0.00 / 0.75795	0.00 / 0.37576	0.00 / 0.12359	0.00 / 0.67663
All-42	42.00 / 0.03921	42.00 / 0.06119	42.00 / 0.12765	42.00 / 0.26068

Each device has equal aggregate weight. These device means are descriptive results, not within-device confidence intervals. The three tasks are kept together in whole-device resampling.

C.3.2 Comparison with the HELP predictor

The HELP predictor uses its published network and adaptation rule with the paid-information evaluation in Appendix B.2.5. We compare the two procedures at 10, 12, 16, 20, 28, 36 and 42 acquired responses, using posterior geometry with a fixed count for our procedure. Both procedures draw from the same pool and predict the same cases; they need not choose the same measurements.

At ten measurements, HELP has lower log-MAE, whereas our procedure has lower mean relative error. Tables C3.5 and C3.6 report each error metric separately at these seven measurement counts. In Table C3.6, the simultaneous log-MAE interval is above zero at ten measurements; the relative-error pointwise intervals include zero at all seven counts.

Table C3.5. Prediction quality at the same seven measurement counts.

k	Our log-MAE	HELP log-MAE	Our relative MAE (%)	HELP relative MAE (%)	Our MAE (ms)	HELP MAE (ms)
10	0.13801	0.11315	14.29	22.15	2.375	1.597
12	0.12840	0.11404	13.40	22.14	2.241	1.563
16	0.15773	0.14883	14.89	19.31	2.859	1.606
20	0.14685	0.15146	14.64	20.42	2.619	1.648
28	0.12857	0.11629	13.63	21.44	2.173	1.581
36	0.12359	0.11413	13.27	19.50	2.049	1.537
42	0.12218	0.11584	13.14	19.27	2.028	1.544

"Our" means posterior geometry with a fixed count and the same option-selection/correction rule as the adaptive procedure. Lower is better for all columns.

Table C3.6. Our fixed-count procedure minus HELP at each count.

k	Δlog-MAE [pointwise 90%]	log-MAE simultaneous 90%	Δrelative MAE (pp) [pointwise 90%]	ΔMAE (ms) [pointwise 90%]
10	+0.02486 [0.01128, 0.03981]	[0.00151, 0.04821]	-7.86 [-19.65, 1.84]	+0.778 [0.109, 1.593]
12	+0.01436 [-0.00899, 0.03778]	[-0.00899, 0.03772]	-8.74 [-20.29, 1.05]	+0.677 [-0.138, 1.626]
16	+0.00890 [-0.00898, 0.02560]	[-0.01445, 0.03225]	-4.42 [-10.84, 1.38]	+1.253 [-0.132, 2.914]
20	-0.00461 [-0.01996, 0.00971]	[-0.02796, 0.01875]	-5.78 [-13.63, 0.96]	+0.972 [-0.098, 2.252]
28	+0.01228 [-0.00534, 0.02990]	[-0.01108, 0.03563]	-7.81 [-18.17, 0.71]	+0.592 [-0.056, 1.356]
36	+0.00946 [-0.00693, 0.02585]	[-0.01389, 0.03282]	-6.22 [-14.45, 0.72]	+0.512 [-0.087, 1.209]
42	+0.00635 [-0.00746, 0.02013]	[-0.01701, 0.02970]	-6.14 [-14.11, 0.60]	+0.484 [-0.089, 1.150]

pp denotes percentage points. Negative differences favor our procedure. The simultaneous band covers the seven log-MAE comparisons; the other metrics have pointwise intervals only.

Adaptive acquisition compared with HELP10 Table C3.7 compares every original adaptive setting with the ten-measurement HELP reference. This is a measurement–quality trade-off, not a same-count comparison. For example, at c $\approx$ 0.003153 the adaptive procedure saves 5.42 measurements relative to HELP10, but its mean log-MAE is higher by 0.04088. The pointwise interval for that error difference is above zero.

Table C3.7. Original adaptive procedure minus HELP10 over all costs.

Index	Δ measurements [pointwise 90%]	Δlog-MAE [pointwise 90%]	log-MAE simultaneous 90%	Δrelative MAE (pp) [pointwise 90%]
0	+11.00 [0.83, 21.48]	+0.03342 [-0.00073, 0.07324]	[-0.18195, 0.24879]	-5.71 [-13.52, 0.69]
1	+7.50 [-2.00, 17.00]	+0.03411 [0.00065, 0.07333]	[-0.18125, 0.24948]	-5.63 [-13.46, 0.80]
2	+6.00 [-2.00, 13.83]	+0.03426 [0.00094, 0.07333]	[-0.18111, 0.24963]	-5.62 [-13.46, 0.84]
3	+3.67 [-2.17, 9.50]	+0.03527 [0.00210, 0.07395]	[-0.18009, 0.25064]	-5.51 [-13.48, 1.13]
4	+2.50 [-3.00, 8.00]	+0.03546 [0.00238, 0.07400]	[-0.17991, 0.25083]	-5.50 [-13.49, 1.20]
5	+0.50 [-3.67, 4.33]	+0.03593 [0.00269, 0.07431]	[-0.17944, 0.25130]	-5.45 [-13.59, 1.48]
6	-1.25 [-4.21, 1.67]	+0.03445 [-0.00041, 0.07421]	[-0.18092, 0.24982]	-5.64 [-13.75, 1.25]
7	-2.17 [-4.75, 0.25]	+0.03693 [0.00435, 0.07467]	[-0.17843, 0.25230]	-5.38 [-13.55, 1.55]
8	-2.92 [-5.00, -1.08]	+0.03697 [0.00315, 0.07563]	[-0.17840, 0.25233]	-5.39 [-13.55, 1.48]
9	-3.58 [-5.17, -2.00]	+0.03870 [0.00615, 0.07599]	[-0.17667, 0.25407]	-5.20 [-13.53, 1.86]
10	-4.92 [-6.08, -3.75]	+0.03994 [0.00693, 0.07730]	[-0.17543, 0.25531]	-5.09 [-13.51, 2.03]
11	-5.42 [-6.42, -4.33]	+0.04088 [0.00709, 0.07862]	[-0.17448, 0.25625]	-4.98 [-13.46, 2.13]
12	-6.08 [-6.42, -5.67]	+0.04335 [0.00923, 0.08071]	[-0.17202, 0.25871]	-4.72 [-13.42, 2.59]
13	-6.50 [-6.67, -6.33]	+0.04890 [0.01197, 0.08887]	[-0.16647, 0.26426]	-4.26 [-13.00, 3.12]
14	-6.92 [-7.33, -6.58]	+0.04775 [0.01268, 0.08140]	[-0.16762, 0.26312]	-2.33 [-10.33, 5.14]
15	-7.58 [-7.67, -7.48]	+0.04828 [0.01970, 0.07444]	[-0.16709, 0.26365]	-1.93 [-10.77, 6.29]
16	-7.83 [-8.00, -7.67]	+0.06866 [0.05784, 0.07786]	[-0.14671, 0.28402]	+0.68 [-8.91, 8.94]
17	-8.42 [-8.67, -8.17]	+0.10566 [0.05790, 0.15543]	[-0.10971, 0.32103]	+10.39 [6.21, 14.30]
18	-8.67 [-8.83, -8.50]	+0.13216 [0.06326, 0.21103]	[-0.08321, 0.34753]	+11.98 [6.36, 17.95]
19	-8.83 [-9.00, -8.67]	+0.14000 [0.06634, 0.22125]	[-0.07537, 0.35537]	+12.90 [6.44, 18.64]
20	-8.92 [-9.00, -8.81]	+0.16031 [0.10319, 0.22219]	[-0.05506, 0.37567]	+14.42 [9.40, 19.44]
21	-9.25 [-9.33, -9.15]	+0.22156 [0.11643, 0.32922]	[0.00619, 0.43692]	+22.49 [12.16, 33.74]
22	-10.00 [-10.00, -10.00]	+0.37033 [0.15496, 0.58570]	[0.15496, 0.58570]	+41.10 [8.14, 79.30]
23	-10.00 [-10.00, -10.00]	+0.37033 [0.15496, 0.58570]	[0.15496, 0.58570]	+41.10 [8.14, 79.30]
24	-10.00 [-10.00, -10.00]	+0.37033 [0.15496, 0.58570]	[0.15496, 0.58570]	+41.10 [8.14, 79.30]

Index	Δ measurements [pointwise 90%]	Δlog-MAE [pointwise 90%]	log-MAE simultaneous 90%	Δrelative MAE (pp) [pointwise 90%]
25	-10.00 [-10.00, -10.00]	+0.37033 [0.15496, 0.58570]	[0.15496, 0.58570]	+41.10 [8.14, 79.30]

A negative measurement difference means fewer measurements than HELP10. Positive error differences mean worse predictions. The log-MAE simultaneous band covers these 26 costs as a separate family.

C.3.3 Choosing measurements at a fixed count

To isolate measurement choice, both rules use the same prepared predictors, selection rule and correction, and neither uses an adaptive stopping decision. At each acquired count, posterior geometry is compared with the mean over 32 predefined random acquisition orders. Figure C3.1(a) shows the paired error differences; panel (b) previews the stopping comparison in C.3.4.

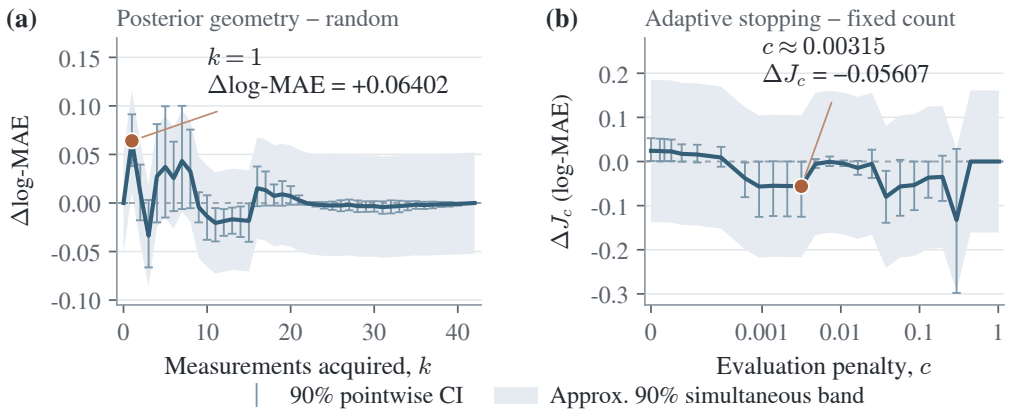

Figure C3.1: **HELP acquisition and stopping comparisons. (a)** Mean log-MAE difference between posterior geometry and random acquisition at each acquired-response count k, computed as posterior geometry minus the mean over 32 random orders within the same tasks before averaging. The horizontal axis is Measurements acquired, k. Negative values favor posterior geometry. **(b)** Mean difference in J_c = log-MAE + ck at each evaluation penalty c, computed as original adaptive stopping minus a fixed-count policy selected on development data for that penalty; both use posterior geometry. The horizontal axis is Evaluation penalty, c. Negative values favor adaptive. In both panels, the solid line shows the mean paired difference across the four test devices after averaging tasks within each device. Vertical error bars show 90% pointwise confidence intervals, and the shaded band shows the approximate 90% simultaneous band over all 43 measurement counts in **(a)** and all 26 penalties in **(b)**. Highlighted points correspond to the examples discussed in Section 4.4.

Complete fixed-count acquisition comparison Posterior geometry has higher mean log-MAE at the first measurement, with the simultaneous interval above zero there. It has lower mean error at 28 of the 41 intermediate counts, but averaging the differences equally across those counts gives +0.00263.

Table C3.8. Posterior geometry minus the mean of 32 random orders.

k	Posterior geometry log-MAE	Random mean log-MAE	Difference [pointwise 90%]	Simultaneous 90% (all 43 counts)
0	0.48348	0.48348	+0.00000 [0.00000, 0.00000]	[-0.05213, 0.05213]

k	Posterior geometry log-MAE	Random mean log-MAE	Difference [pointwise 90%]	Simultaneous 90% (all 43 counts)
1	0.32075	0.25674	+0.06402 [0.03796, 0.09132]	[0.01188, 0.11615]
2	0.22853	0.21750	+0.01103 [-0.01781, 0.03956]	[-0.04110, 0.06317]
3	0.16631	0.19998	-0.03367 [-0.06637, 0.00310]	[-0.08581, 0.01846]
4	0.20731	0.17976	+0.02755 [-0.02001, 0.08127]	[-0.02459, 0.07968]
5	0.20879	0.17164	+0.03715 [-0.01498, 0.09962]	[-0.01498, 0.08928]
6	0.18774	0.16217	+0.02557 [-0.00841, 0.06316]	[-0.02657, 0.07770]
7	0.20285	0.15956	+0.04329 [-0.00632, 0.10005]	[-0.00885, 0.09542]
8	0.18790	0.15539	+0.03251 [-0.00535, 0.07563]	[-0.01962, 0.08464]
9	0.14907	0.15335	-0.00428 [-0.02000, 0.01106]	[-0.05641, 0.04786]
10	0.13801	0.15175	-0.01374 [-0.03795, 0.00793]	[-0.06587, 0.03840]
11	0.13034	0.15096	-0.02061 [-0.03957, -0.00527]	[-0.07275, 0.03152]
12	0.12840	0.14704	-0.01863 [-0.03381, -0.00629]	[-0.07077, 0.03350]
13	0.12930	0.14610	-0.01679 [-0.03169, -0.00475]	[-0.06892, 0.03534]
14	0.12762	0.14524	-0.01762 [-0.03505, -0.00361]	[-0.06975, 0.03451]
15	0.12479	0.14324	-0.01844 [-0.03994, -0.00045]	[-0.07057, 0.03369]
16	0.15773	0.14253	+0.01520 [-0.00331, 0.03758]	[-0.03693, 0.06734]
17	0.15426	0.14100	+0.01327 [-0.00264, 0.03247]	[-0.03886, 0.06540]
18	0.14923	0.14193	+0.00730 [-0.00267, 0.01911]	[-0.04483, 0.05944]
19	0.14997	0.14105	+0.00892 [-0.00279, 0.02259]	[-0.04322, 0.06105]
20	0.14685	0.13974	+0.00711 [-0.00177, 0.01737]	[-0.04503, 0.05924]
21	0.14269	0.13978	+0.00290 [-0.00169, 0.00784]	[-0.04923, 0.05504]

Negative differences favor posterior geometry. The same 43-count band is retained at counts 0 and 42, even though the two procedures produce matching endpoint predictions up to numerical precision.

Complete fixed-count acquisition comparison (continued)

Table C3.8. Posterior geometry minus the mean of 32 random orders (continued).

k	Posterior geometry log-MAE	Random mean log-MAE	Difference [pointwise 90%]	Simultaneous 90% (all 43 counts)
22	0.13740	0.13778	-0.00038 [-0.00194, 0.00095]	[-0.05252, 0.05175]
23	0.13544	0.13696	-0.00152 [-0.00372, 0.00107]	[-0.05366, 0.05061]
24	0.13381	0.13612	-0.00232 [-0.00579, 0.00098]	[-0.05445, 0.04982]
25	0.13266	0.13524	-0.00257 [-0.00711, 0.00147]	[-0.05470, 0.04956]
26	0.13152	0.13407	-0.00254 [-0.00851, 0.00282]	[-0.05468, 0.04959]
27	0.13104	0.13285	-0.00181 [-0.00752, 0.00352]	[-0.05395, 0.05032]
28	0.12857	0.13167	-0.00310 [-0.00938, 0.00264]	[-0.05524, 0.04903]
29	0.12766	0.13098	-0.00332 [-0.00919, 0.00177]	[-0.05545, 0.04881]
30	0.12725	0.13048	-0.00323 [-0.00895, 0.00187]	[-0.05536, 0.04890]
31	0.12548	0.12972	-0.00424 [-0.01107, 0.00171]	[-0.05637, 0.04789]
32	0.12539	0.12906	-0.00367 [-0.01077, 0.00273]	[-0.05581, 0.04846]
33	0.12479	0.12797	-0.00318 [-0.00956, 0.00237]	[-0.05532, 0.04895]
34	0.12466	0.12722	-0.00256 [-0.00753, 0.00178]	[-0.05469, 0.04957]
35	0.12458	0.12635	-0.00178 [-0.00596, 0.00191]	[-0.05391, 0.05036]
36	0.12359	0.12574	-0.00215 [-0.00617, 0.00124]	[-0.05428, 0.04998]
37	0.12358	0.12532	-0.00174 [-0.00548, 0.00145]	[-0.05388, 0.05039]
38	0.12320	0.12456	-0.00137 [-0.00410, 0.00110]	[-0.05350, 0.05077]
39	0.12272	0.12391	-0.00119 [-0.00372, 0.00108]	[-0.05332, 0.05094]
40	0.12234	0.12342	-0.00108 [-0.00270, 0.00043]	[-0.05321, 0.05105]
41	0.12234	0.12283	-0.00049 [-0.00126, 0.00018]	[-0.05262, 0.05164]
42	0.12218	0.12218	+0.00000 [0.00000, 0.00000]	[-0.05213, 0.05213]

Negative differences favor posterior geometry. The same 43-count band is retained at counts 0 and 42, even though the two procedures produce matching endpoint predictions up to numerical precision.

Table C3.9. Device contributions to the fixed-count contrast.

Device	Posterior geometry at k=1	Random at k=1	Δ at k=1	Mean Δ over k=1–41
Titan RTX	0.36598	0.25847	+0.10751	-0.00184
Xeon Gold 6226	0.12037	0.09677	+0.02360	-0.00165
Pixel 2	0.27544	0.19324	+0.08220	+0.00594

Device	Posterior geometry at k=1	Random at k=1	Δ at k=1	Mean Δ over k=1–41
Raspberry Pi 4	0.52122	0.47847	+0.04275	+0.00806

Each device entry averages its three tasks. The final column also averages the 41 intermediate counts equally; it is descriptive and does not select an acquisition count.

C.3.4 Deciding when to stop

Both stopping comparisons use posterior geometry and the same final prediction rule. The fixed-count policy chooses one count on the six development devices for each evaluation cost c, then applies that count to every test task. The original adaptive policy instead uses each task's measurements to decide when to stop. We compare their final log-MAE plus c times the acquired count, denoted Jc, as in Equation (B.2).

Across the 26 costs, the original adaptive policy has lower mean Jc at 16, ties at three and higher means at seven. Pointwise intervals favor adaptive stopping at four costs and favor the fixed count at four. The simultaneous band includes zero at every cost. Figure C3.1(b) and the complete tables below show the setting-specific benefits and the unfavorable results together.

At c $\approx$ 0.003153, the development-selected fixed count is five. The adaptive policy acquires 4.58 responses on average and has lower mean log-MAE and Jc. Table C3.10 shows how this average arises: it buys more measurements on the Xeon device, fewer on Pixel 2 and Raspberry Pi 4, and the same mean count on Titan RTX. The small disadvantage on Titan RTX is retained alongside the other devices' gains.

Table C3.10. Stopping comparison at c $\approx$ 0.003153.

Device	Fixed k	Adaptive k	Fixed log-MAE	Adaptive log-MAE	Adaptive - fixed Jc
Titan RTX	5.00	5.00	0.06001	0.06071	+0.00070
Xeon Gold 6226	5.00	6.33	0.07834	0.07036	-0.00378
Pixel 2	5.00	3.33	0.12896	0.13326	-0.00097
Raspberry Pi 4	5.00	3.67	0.56784	0.35182	-0.22022
Four-device mean	5.00	4.58	0.20879	0.15404	-0.05607

Table C3.11. Paired mean differences at the same cost.

Outcome	Adaptive - fixed	90% pointwise interval
Acquired count	-0.42	[-1.42, 0.67]
log-MAE	-0.05475	[-0.12424, 0.00160]
Cost-inclusive loss Jc	-0.05607	[-0.12525, -0.00042]

The Jc interval is below zero, while the separate error and count intervals include zero. The simultaneous band across all costs also includes zero at this setting. Thus the two favorable mean components are not separate interval-supported improvements.

Development choices used in the stopping comparisons For the supplementary comparison, adaptive stopping is also allowed to choose its decision penalty on development data. This selects one of the 26 existing adaptive settings; it does not fit a new policy. The evaluation cost c remains the same for both policies. Table C3.12 gives the selected fixed count and adaptive setting at every evaluation cost.

Table C3.12. Development-selected counts and adaptive settings.

Index	Evaluation c	Fixed k	Fixed dev. mean Jc	Adaptive setting index	Selected penalty (log units)	Adaptive dev. mean k	Adaptive dev. mean Jc
0	0	41	0.06941	0	0	27.28	0.07494
1	5.0897e-05	41	0.07149	7	0.000605271	9.33	0.07590
2	7.68957e-05	41	0.07256	7	0.000605271	9.33	0.07614
3	0.000116175	41	0.07417	7	0.000605271	9.33	0.07651
4	0.000175518	28	0.07595	7	0.000605271	9.33	0.07706
5	0.000265174	28	0.07846	7	0.000605271	9.33	0.07790
6	0.000400627	10	0.08046	10	0.00208727	5.83	0.07876
7	0.000605271	8	0.08221	10	0.00208727	5.83	0.07995
8	0.000914449	5	0.08467	10	0.00208727	5.83	0.08176
9	0.00138156	5	0.08701	10	0.00208727	5.83	0.08448
10	0.00208727	5	0.09053	10	0.00208727	5.83	0.08860
11	0.00315347	5	0.09587	10	0.00208727	5.83	0.09482
12	0.00476429	3	0.10178	11	0.00315347	4.78	0.10384
13	0.00719793	3	0.10908	16	0.024822	2.17	0.10935
14	0.0108747	3	0.12011	16	0.024822	2.17	0.11732
15	0.0164296	3	0.13677	16	0.024822	2.17	0.12935
16	0.024822	3	0.16195	16	0.024822	2.17	0.14754
17	0.0375013	1	0.18814	16	0.024822	2.17	0.17501
18	0.0566572	1	0.20729	19	0.0855983	1.06	0.19892
19	0.0855983	1	0.23623	19	0.0855983	1.06	0.22947
20	0.129323	1	0.27996	19	0.0855983	1.06	0.27562
21	0.195382	1	0.34602	19	0.0855983	1.06	0.34535
22	0.295185	1	0.44582	20	0.129323	1.00	0.44582
23	0.445968	0	0.54054	22	0.295185	0.00	0.54054
24	0.673772	0	0.54054	22	0.295185	0.00	0.54054
25	1.01794	0	0.54054	22	0.295185	0.00	0.54054

The selected adaptive index refers to the original grid in Table C3.2; the adjacent penalty is expressed in reported log-MAE units. Development means average the six development devices. They are selection scores, not test results. Ties are resolved by fewer mean acquisitions, then the smaller count or setting index (B.3.4).

Original adaptive stopping: all final outcomes Table C3.13 reports the endpoints of the original adaptive and fixed-count policies at every cost. The following table reports paired differences and uncertainty. A lower Jc can result from lower error, fewer measurements, or both; it does not require the policies to acquire the same count.

Table C3.13. Original adaptive stopping versus the development-selected fixed count.

Index	Fixed k	Adaptive mean k	Fixed log-MAE	Adaptive log-MAE	Fixed Jc	Adaptive Jc	ΔJc
0	41	21.00	0.12234	0.14657	0.12234	0.14657	+0.02423
1	41	17.50	0.12234	0.14727	0.12443	0.14816	+0.02373
2	41	16.00	0.12234	0.14741	0.12549	0.14864	+0.02314
3	41	13.67	0.12234	0.14843	0.12710	0.15001	+0.02291
4	28	12.50	0.12857	0.14861	0.13348	0.15081	+0.01732
5	28	10.50	0.12857	0.14908	0.13599	0.15186	+0.01587
6	10	8.75	0.13801	0.14760	0.14202	0.15110	+0.00909
7	8	7.83	0.18790	0.15009	0.19275	0.15483	-0.03792
8	5	7.08	0.20879	0.15012	0.21336	0.15659	-0.05677
9	5	6.42	0.20879	0.15185	0.21570	0.16072	-0.05498
10	5	5.08	0.20879	0.15309	0.21922	0.16370	-0.05553
11	5	4.58	0.20879	0.15404	0.22456	0.16849	-0.05607
12	3	3.92	0.16631	0.15650	0.18060	0.17516	-0.00545
13	3	3.50	0.16631	0.16205	0.18791	0.18724	-0.00067
14	3	3.08	0.16631	0.16090	0.19894	0.19443	-0.00451

Index	Fixed k	Adaptive mean k	Fixed log-MAE	Adaptive log-MAE	Fixed Jc	Adaptive Jc	ΔJc
15	3	2.42	0.16631	0.16143	0.21560	0.20113	-0.01447
16	3	2.17	0.16631	0.18181	0.24078	0.23559	-0.00519
17	1	1.58	0.32075	0.21881	0.35825	0.27819	-0.08006
18	1	1.33	0.32075	0.24531	0.37741	0.32086	-0.05655
19	1	1.17	0.32075	0.25315	0.40635	0.35301	-0.05333
20	1	1.08	0.32075	0.27346	0.45007	0.41356	-0.03652
21	1	0.75	0.32075	0.33471	0.51613	0.48124	-0.03489
22	1	0.00	0.32075	0.48348	0.61594	0.48348	-0.13245
23	0	0.00	0.48348	0.48348	0.48348	0.48348	+0.00000
24	0	0.00	0.48348	0.48348	0.48348	0.48348	+0.00000
25	0	0.00	0.48348	0.48348	0.48348	0.48348	+0.00000

ΔJc is adaptive minus fixed; negative favors adaptive. For the original comparison, the decision penalty corresponds to c at the same index. For the supplementary comparison it is the setting selected in Table C3.12. All results use the four test devices.

Original adaptive stopping: paired intervals

Table C3.14. Original adaptive stopping: count, error and cost-inclusive differences.

Index	Δ measurements [pointwise 90%]	Δlog-MAE [pointwise 90%]	ΔJc [pointwise 90%]	Jc simultaneous 90% (all 26 costs)
0	-20.00	+0.02423	+0.02423	[-0.13672, 0.18518]
	[-30.17, -9.52]	[0.00110, 0.05283]	[0.00110, 0.05283]	
1	-23.50	+0.02493	+0.02373	[-0.13722, 0.18468]
	[-33.00, -14.00]	[0.00148, 0.05345]	[0.00082, 0.05178]	
2	-25.00	+0.02507	+0.02314	[-0.13781, 0.18409]
	[-33.00, -17.17]	[0.00179, 0.05346]	[0.00062, 0.05092]	
3	-27.33	+0.02608	+0.02291	[-0.13804, 0.18386]
	[-33.17, -21.50]	[0.00405, 0.05334]	[0.00120, 0.04973]	
4	-15.50	+0.02004	+0.01732	[-0.14363, 0.17827]
	[-21.00, -10.00]	[0.00203, 0.04220]	[-0.00031, 0.03894]	
5	-17.50	+0.02051	+0.01587	[-0.14508, 0.17682]
	[-21.67, -13.67]	[0.00202, 0.04314]	[-0.00226, 0.03793]	
6	-1.25	+0.00959	+0.00909	[-0.15186, 0.17004]
	[-4.21, 1.67]	[-0.01169, 0.03439]	[-0.01175, 0.03317]	
7	-0.17	-0.03782	-0.03792	[-0.19887, 0.12303]
	[-2.75, 2.25]	[-0.07905, -0.00292]	[-0.08075, -0.00221]	
8	+2.08	-0.05867	-0.05677	[-0.21772, 0.10418]
	[0.00, 3.92]	[-0.12609, -0.00296]	[-0.12545, -0.00021]	
9	+1.42	-0.05694	-0.05498	[-0.21593, 0.10597]
	[-0.17, 3.00]	[-0.12461, -0.00146]	[-0.12352, 0.00066]	
10	+0.08	-0.05570	-0.05553	[-0.21648, 0.10542]
	[-1.08, 1.25]	[-0.12423, 0.00055]	[-0.12436, 0.00021]	
11	-0.42	-0.05475	-0.05607	[-0.21702, 0.10488]
	[-1.42, 0.67]	[-0.12424, 0.00160]	[-0.12525, -0.00042]	
12	+0.92	-0.00981	-0.00545	[-0.16640, 0.15550]
	[0.58, 1.33]	[-0.02091, 0.00274]	[-0.01456, 0.00552]	
13	+0.50	-0.00426	-0.00067	[-0.16162, 0.16028]
	[0.33, 0.67]	[-0.01512, 0.00884]	[-0.01033, 0.01139]	
14	+0.08	-0.00541	-0.00451	[-0.16546, 0.15644]
	[-0.33, 0.42]	[-0.01344, 0.00403]	[-0.01045, 0.00235]	
15	-0.58	-0.00488	-0.01447	[-0.17542, 0.14648]
	[-0.67, -0.48]	[-0.02203, 0.01226]	[-0.03024, 0.00131]	
16	-0.83	+0.01550	-0.00519	[-0.16614, 0.15576]
	[-1.00, -0.67]	[-0.01650, 0.04749]	[-0.03719, 0.02681]	
17	+0.58	-0.10194	-0.08006	[-0.24101, 0.08089]
	[0.33, 0.83]	[-0.15951, -0.04624]	[-0.13888, -0.02124]	
18	+0.33	-0.07544	-0.05655	[-0.21750, 0.10440]
	[0.17, 0.50]	[-0.15220, -0.00308]	[-0.12294, 0.00370]	

Index	Δ measurements [pointwise 90%]	Δlog-MAE [pointwise 90%]	ΔJc [pointwise 90%]	Jc simultaneous 90% (all 26 costs)
19	+0.17	-0.06760	-0.05333	[-0.21428, 0.10762]
	[0.00, 0.33]	[-0.13687, 0.00000]	[-0.11012, 0.00000]	
20	+0.08	-0.04729	-0.03652	[-0.19747, 0.12443]
	[0.00, 0.19]	[-0.10641, 0.00000]	[-0.08216, 0.00000]	
21	-0.25	+0.01396	-0.03489	[-0.19584, 0.12606]
	[-0.33, -0.15]	[-0.04811, 0.07375]	[-0.08937, 0.01314]	
22	-1.00	+0.16273	-0.13245	[-0.29340, 0.02850]
	[-1.00, -1.00]	[-0.00264, 0.32368]	[-0.29782, 0.02850]	
23	+0.00	+0.00000	+0.00000	[-0.16095, 0.16095]
	[0.00, 0.00]	[0.00000, 0.00000]	[0.00000, 0.00000]	
24	+0.00	+0.00000	+0.00000	[-0.16095, 0.16095]
	[0.00, 0.00]	[0.00000, 0.00000]	[0.00000, 0.00000]	
25	+0.00	+0.00000	+0.00000	[-0.16095, 0.16095]
	[0.00, 0.00]	[0.00000, 0.00000]	[0.00000, 0.00000]	

The mean and pointwise interval occupy separate lines in each difference column. The last column is the simultaneous interval for Jc, not for its error and count components. The two adaptive comparisons use separate simultaneous families.

Development-selected adaptive setting: all final outcomes Giving the adaptive policy a development-based choice of its decision penalty does not establish a grid-wide advantage. This comparison has lower mean Jc than the fixed policy at 13 costs, ties at three and higher means at ten. Its pointwise Jc interval is below zero only at index 17 and above zero at indices 0–3; every simultaneous interval includes zero.

Table C3.15. Development-selected adaptive setting versus the development-selected fixed count.

Index	Fixed k	Adaptive mean k	Fixed log-MAE	Adaptive log-MAE	Fixed Jc	Adaptive Jc	ΔJc
0	41	21.00	0.12234	0.14657	0.12234	0.14657	+0.02423
1	41	7.83	0.12234	0.15009	0.12443	0.15048	+0.02606
2	41	7.83	0.12234	0.15009	0.12549	0.15069	+0.02519
3	41	7.83	0.12234	0.15009	0.12710	0.15100	+0.02389
4	28	7.83	0.12857	0.15009	0.13348	0.15146	+0.01798
5	28	7.83	0.12857	0.15009	0.13599	0.15216	+0.01617
6	10	5.08	0.13801	0.15309	0.14202	0.15513	+0.01311
7	8	5.08	0.18790	0.15309	0.19275	0.15617	-0.03658
8	5	5.08	0.20879	0.15309	0.21336	0.15774	-0.05562
9	5	5.08	0.20879	0.15309	0.21570	0.16011	-0.05558
10	5	5.08	0.20879	0.15309	0.21922	0.16370	-0.05553
11	5	5.08	0.20879	0.15309	0.22456	0.16912	-0.05544
12	3	4.58	0.16631	0.15404	0.18060	0.17587	-0.00473
13	3	2.17	0.16631	0.18181	0.18791	0.19740	+0.00950
14	3	2.17	0.16631	0.18181	0.19894	0.20537	+0.00643
15	3	2.17	0.16631	0.18181	0.21560	0.21741	+0.00180
16	3	2.17	0.16631	0.18181	0.24078	0.23559	-0.00519
17	1	2.17	0.32075	0.18181	0.35825	0.26306	-0.09519
18	1	1.17	0.32075	0.25315	0.37741	0.31925	-0.05816
19	1	1.17	0.32075	0.25315	0.40635	0.35301	-0.05333
20	1	1.17	0.32075	0.25315	0.45007	0.40403	-0.04605
21	1	1.17	0.32075	0.25315	0.51613	0.48110	-0.03504
22	1	1.08	0.32075	0.27346	0.61594	0.59324	-0.02270
23	0	0.00	0.48348	0.48348	0.48348	0.48348	+0.00000
24	0	0.00	0.48348	0.48348	0.48348	0.48348	+0.00000
25	0	0.00	0.48348	0.48348	0.48348	0.48348	+0.00000

ΔJc is adaptive minus fixed; negative favors adaptive. For the original comparison, the decision penalty corresponds to c at the same index. For the supplementary comparison it is the setting selected in Table C3.12. All results use the four test devices.

Development-selected adaptive setting: paired intervals

Table C3.16. Development-selected adaptive setting: count, error and cost-inclusive differences.

Index	Δ measurements [pointwise 90%]	Δlog-MAE [pointwise 90%]	ΔJc [pointwise 90%]	Jc simultaneous 90% (all 26 costs)
0	-20.00	+0.02423	+0.02423	[-0.07180, 0.12027]
	[-30.17, -9.52]	[0.00110, 0.05283]	[0.00110, 0.05283]	
1	-33.17	+0.02774	+0.02606	[-0.06998, 0.12209]
	[-35.75, -30.75]	[0.00469, 0.05557]	[0.00311, 0.05380]	
2	-33.17	+0.02774	+0.02519	[-0.07084, 0.12123]
	[-35.75, -30.75]	[0.00469, 0.05557]	[0.00230, 0.05290]	
3	-33.17	+0.02774	+0.02389	[-0.07214, 0.11993]
	[-35.75, -30.75]	[0.00469, 0.05557]	[0.00108, 0.05153]	
4	-20.17	+0.02152	+0.01798	[-0.07806, 0.11401]
	[-22.75, -17.75]	[0.00263, 0.04408]	[-0.00060, 0.04026]	
5	-20.17	+0.02152	+0.01617	[-0.07986, 0.11220]
	[-22.75, -17.75]	[0.00263, 0.04408]	[-0.00223, 0.03831]	
6	-4.92	+0.01508	+0.01311	[-0.08293, 0.10914]
	[-6.08, -3.75]	[-0.00522, 0.03813]	[-0.00702, 0.03591]	
7	-2.92	-0.03481	-0.03658	[-0.13261, 0.05945]
	[-4.08, -1.75]	[-0.07827, 0.00245]	[-0.08074, 0.00079]	
8	+0.08	-0.05570	-0.05562	[-0.15166, 0.04041]
	[-1.08, 1.25]	[-0.12423, 0.00055]	[-0.12428, 0.00040]	
9	+0.08	-0.05570	-0.05558	[-0.15162, 0.04045]
	[-1.08, 1.25]	[-0.12423, 0.00055]	[-0.12431, 0.00032]	
10	+0.08	-0.05570	-0.05553	[-0.15156, 0.04051]
	[-1.08, 1.25]	[-0.12423, 0.00055]	[-0.12436, 0.00021]	
11	+0.08	-0.05570	-0.05544	[-0.15147, 0.04060]
	[-1.08, 1.25]	[-0.12423, 0.00055]	[-0.12442, 0.00003]	
12	+1.58	-0.01228	-0.00473	[-0.10077, 0.09130]
	[0.58, 2.67]	[-0.02550, 0.00240]	[-0.01438, 0.00611]	
13	-0.83	+0.01550	+0.00950	[-0.08654, 0.10553]
	[-1.00, -0.67]	[-0.01650, 0.04749]	[-0.02250, 0.04149]	
14	-0.83	+0.01550	+0.00643	[-0.08960, 0.10247]
	[-1.00, -0.67]	[-0.01650, 0.04749]	[-0.02556, 0.03843]	
15	-0.83	+0.01550	+0.00180	[-0.09423, 0.09784]
	[-1.00, -0.67]	[-0.01650, 0.04749]	[-0.03019, 0.03380]	
16	-0.83	+0.01550	-0.00519	[-0.10122, 0.09085]
	[-1.00, -0.67]	[-0.01650, 0.04749]	[-0.03719, 0.02681]	
17	+1.17	-0.13894	-0.09519	[-0.19123, 0.00084]
	[1.00, 1.33]	[-0.22613, -0.05142]	[-0.18551, -0.01079]	
18	+0.17	-0.06760	-0.05816	[-0.15419, 0.03788]
	[0.00, 0.33]	[-0.13687, 0.00000]	[-0.11917, 0.00000]	
19	+0.17	-0.06760	-0.05333	[-0.14937, 0.04270]
	[0.00, 0.33]	[-0.13687, 0.00000]	[-0.11012, 0.00000]	
20	+0.17	-0.06760	-0.04605	[-0.14208, 0.04999]
	[0.00, 0.33]	[-0.13687, 0.00000]	[-0.09646, 0.00000]	
21	+0.17	-0.06760	-0.03504	[-0.13107, 0.06100]
	[0.00, 0.33]	[-0.13687, 0.00000]	[-0.07581, 0.00000]	
22	+0.08	-0.04729	-0.02270	[-0.11873, 0.07334]
	[0.00, 0.19]	[-0.10641, 0.00000]	[-0.05106, 0.00000]	
23	+0.00	+0.00000	+0.00000	[-0.09603, 0.09603]
	[0.00, 0.00]	[0.00000, 0.00000]	[0.00000, 0.00000]	
24	+0.00	+0.00000	+0.00000	[-0.09603, 0.09603]
	[0.00, 0.00]	[0.00000, 0.00000]	[0.00000, 0.00000]	
25	+0.00	+0.00000	+0.00000	[-0.09603, 0.09603]
	[0.00, 0.00]	[0.00000, 0.00000]	[0.00000, 0.00000]	

The mean and pointwise interval occupy separate lines in each difference column. The last column is the simultaneous interval for Jc, not for its error and count components. The two adaptive comparisons use separate simultaneous families.

Device-level stopping results over the full cost grid Table C3.17 retains the per-device Jc differences for both adaptive comparisons. Each cell gives the original adaptive result on the first line and the development-selected adaptive result on the second. Both subtract the same development-selected fixed-count policy at that cost. This exposes differences between devices without treating their three tasks as independent devices.

Table C3.17. Per-device adaptive-minus-fixed Jc: original / selected penalty.

Index	Titan RTX	Xeon Gold 6226	Pixel 2	Raspberry Pi 4
0	-0.00003	+0.00154	+0.00452	+0.09091
	-0.00003	+0.00154	+0.00452	+0.09091
1	+0.00093	+0.00048	+0.00449	+0.08901
	+0.00959	+0.00341	+0.00222	+0.08901
2	+0.00085	-0.00005	+0.00374	+0.08804
	+0.00878	+0.00263	+0.00133	+0.08804
3	+0.00266	+0.00045	+0.00195	+0.08658
	+0.00756	+0.00145	-0.00002	+0.08658
4	+0.00169	-0.00115	+0.00052	+0.06823
	+0.00521	-0.00045	-0.00108	+0.06823
5	+0.00274	-0.00138	-0.00393	+0.06605
	+0.00360	-0.00197	-0.00299	+0.06605
6	-0.00617	-0.00311	-0.01734	+0.06296
	+0.00351	-0.00081	-0.01324	+0.06296
7	+0.00372	-0.00474	-0.02003	-0.13062
	+0.00795	-0.00294	-0.02071	-0.13062
8	-0.00940	-0.00484	+0.00441	-0.21724
	-0.00214	-0.00606	+0.00295	-0.21724
9	-0.00337	-0.00328	+0.00460	-0.21786
	-0.00183	-0.00513	+0.00248	-0.21786
10	-0.00136	-0.00372	+0.00177	-0.21880
	-0.00136	-0.00372	+0.00177	-0.21880
11	+0.00070	-0.00378	-0.00097	-0.22022
	-0.00065	-0.00158	+0.00071	-0.22022
12	-0.01536	-0.01429	-0.01100	+0.01886
	-0.01776	-0.00903	-0.01100	+0.01886
13	-0.01032	-0.01035	-0.01019	+0.02819
	+0.02051	-0.00559	+0.06248	-0.03941
14	+0.01077	-0.00790	-0.00896	-0.01194
	+0.01806	-0.00804	+0.05880	-0.04308
15	+0.01436	-0.01175	-0.01542	-0.04506
	+0.01436	-0.01175	+0.05324	-0.04864
16	+0.00876	-0.01734	+0.04485	-0.05703
	+0.00876	-0.01734	+0.04485	-0.05703
17	-0.19142	+0.02625	-0.06872	-0.08635
	-0.20267	+0.02411	-0.02242	-0.17978
18	-0.17865	+0.01478	-0.06234	+0.00000
	-0.17029	+0.00000	-0.06234	+0.00000
19	-0.16064	+0.00000	-0.05269	+0.00000
	-0.16064	+0.00000	-0.05269	+0.00000
20	-0.14607	+0.00000	+0.00000	+0.00000
	-0.14607	+0.00000	-0.03812	+0.00000
21	-0.01348	+0.02217	-0.15235	+0.00411
	-0.12405	+0.00000	-0.01610	+0.00000
22	+0.09679	-0.03979	-0.44703	-0.13977
	-0.09078	+0.00000	+0.00000	+0.00000
23	+0.00000	+0.00000	+0.00000	+0.00000
	+0.00000	+0.00000	+0.00000	+0.00000
24	+0.00000	+0.00000	+0.00000	+0.00000
	+0.00000	+0.00000	+0.00000	+0.00000
25	+0.00000	+0.00000	+0.00000	+0.00000
	+0.00000	+0.00000	+0.00000	+0.00000

First line: original adaptive penalty. Second line: development-selected adaptive penalty. Negative values favor the respective adaptive policy. Zero is an observed tie, not a missing value.

C.3.5 Ranking, candidate selection and tail errors

Average error and the quality of selecting fast architectures need not change together. We therefore report the additional metrics defined in Appendix B.4.3. Spearman correlation measures ranking agreement, with higher values better. The fastest-selection penalty is the true latency of the predicted-fastest architecture divided by the true fastest latency, minus one. Its millisecond and log-ratio forms describe the same selected architecture in different units.

The adaptive settings labeled in Figure 4 have similar log-MAE but share a much larger mean fastest-selection penalty than all-42. At c $\approx$ 0.003153, it is 51.19%, compared with 1.18% for all-42. This result is retained rather than inferred from average error. Tables C3.18–C3.21 show the full original adaptive grid, all seven HELP counts and paired uncertainty for the main comparisons.

Table C3.18. Full adaptive-grid error, ranking and fastest-architecture selection.

Index / reference	Relative MAE (%)	MAE (ms)	Spearman ρ	Fastest penalty (%)	Extra latency (ms)	Log latency ratio
0	16.44	2.638	0.9358	51.19	1.094	0.2099
1	16.52	2.645	0.9364	51.19	1.094	0.2099
2	16.54	2.646	0.9368	51.19	1.094	0.2099
3	16.64	2.653	0.9367	51.12	1.092	0.2093
4	16.65	2.653	0.9366	51.12	1.092	0.2093
5	16.70	2.650	0.9360	51.12	1.092	0.2093
6	16.51	2.634	0.9366	51.12	1.092	0.2093
7	16.78	2.661	0.9363	51.19	1.094	0.2099
8	16.77	2.665	0.9356	51.19	1.094	0.2099
9	16.95	2.679	0.9354	51.19	1.094	0.2099
10	17.06	2.691	0.9340	51.19	1.094	0.2099
11	17.18	2.697	0.9343	51.19	1.094	0.2099
12	17.43	2.719	0.9340	51.19	1.094	0.2099
13	17.89	2.812	0.9326	51.19	1.094	0.2099
14	19.82	2.627	0.9243	51.19	1.094	0.2099
15	20.22	2.497	0.9220	51.19	1.094	0.2099
16	22.83	2.890	0.9191	51.19	1.094	0.2099
17	32.54	3.372	0.9066	147.58	1.511	0.3925
18	34.14	3.989	0.9055	147.58	1.511	0.3925
19	35.06	4.055	0.9062	147.58	1.511	0.3925
20	36.57	4.316	0.9019	147.58	1.511	0.3925
21	44.64	5.209	0.8897	147.58	1.511	0.3925
22	63.25	6.913	0.8705	147.25	1.498	0.3892
23	63.25	6.913	0.8705	147.25	1.498	0.3892
24	63.25	6.913	0.8705	147.25	1.498	0.3892
25	63.25	6.913	0.8705	147.25	1.498	0.3892
None	63.25	6.913	0.8705	147.25	1.498	0.3892
All-42	13.14	2.028	0.9328	1.18	0.078	0.0114
HELP10	22.15	1.597	0.9370	21.25	0.611	0.1536

Higher Spearman is better; lower is better for the other columns. The indices and reference definitions match Tables C3.1–C3.2. Entries are means; representative paired uncertainty is in Table C3.21.

Shortlists and large prediction errors For a five-architecture shortlist, the mean penalty compares the mean true latency of the predicted five-fastest set with that of the true five-fastest set. The best penalty asks whether the shortlist includes an architecture near the actual fastest. Tail summaries separately show large relative errors: the mean device 95th percentile averages four within-device percentiles, whereas the worst error is the maximum over all 504 scored predictions.

Table C3.19. Full adaptive-grid shortlist and tail-error results.

Index / reference	Mean-five penalty (%)	Best-in-five penalty (%)	Mean device p95 (%)	Worst relative error (%)	Errors >100%	Log-floored outputs
0	10.48	15.37	34.45	1162.57	4	0
1	10.92	15.37	34.20	1162.57	4	0
2	10.92	15.37	33.98	1162.57	4	0
3	10.92	15.37	34.21	1162.57	4	0
4	11.15	15.37	33.61	1162.57	4	0
5	11.42	15.37	33.68	1162.57	4	0
6	11.15	15.37	33.54	1162.57	4	0
7	11.07	15.37	33.64	1162.57	4	0
8	11.07	15.37	33.63	1162.57	4	0
9	11.15	15.37	33.78	1162.57	4	0
10	11.15	15.37	34.25	1162.57	4	0
11	11.15	15.37	34.65	1162.57	4	0
12	11.15	15.37	34.65	1162.57	4	0
13	11.15	15.37	34.68	1162.57	4	0
14	10.06	3.80	59.62	1162.57	9	0
15	12.57	3.80	60.13	1162.57	11	0
16	11.52	3.80	67.61	1162.57	10	0
17	26.46	3.80	83.96	3239.94	14	0
18	29.31	3.80	84.26	3239.94	13	0
19	29.31	3.80	86.02	3239.94	13	0
20	28.90	3.80	86.32	3239.94	13	0
21	30.69	3.80	102.14	3239.94	39	0
22	24.87	3.80	92.56	2436.47	90	0
23	24.87	3.80	92.56	2436.47	90	0
24	24.87	3.80	92.56	2436.47	90	0
25	24.87	3.80	92.56	2436.47	90	0
None	24.87	3.80	92.56	2436.47	90	0
All-42	12.19	0.00	37.28	327.88	5	0
HELP10	13.16	15.37	58.69	4018.69	9	0

Error counts use 504 prediction cases. A relative error above 100% is counted strictly. The log floor is used only for log scoring; raw linear predictions remain the basis for the other HELP metrics. No adaptive or HELP10 output in this table requires the floor.

Additional quality at each of the seven common counts Table C3.20 retains both procedures at every common acquisition count. In particular, HELP has one log-floored output at counts 16 and 20. Those outputs still contribute their original linear predictions to relative-error and ranking metrics. The different behavior of log-MAE, relative error and worst error should therefore be read directly from their respective columns, not as equivalent scores.

Table C3.20. Ranking, shortlist and tail metrics at the same counts.

Procedure	k	Spearman ρ	Fastest penalty (%)	Mean-five penalty (%)	Best-five penalty (%)	Device p95 (%)	Worst error (%)	Log floor
Our fixed count	10	0.9363	6.37	8.83	0.00	34.68	544.30	0
HELP predictor	10	0.9370	21.25	13.16	15.37	58.69	4018.69	0
Our fixed count	12	0.9396	4.92	7.93	0.00	32.19	540.57	0
HELP predictor	12	0.9368	22.64	14.03	15.37	48.46	4358.32	0
Our fixed count	16	0.9276	1.12	10.01	0.00	33.41	237.45	0
HELP predictor	16	0.9363	22.70	12.78	15.37	53.61	2794.02	1

Procedure	k	Spearman ρ	Fastest penalty (%)	Mean-five penalty (%)	Best-five penalty (%)	Device p95 (%)	Worst error (%)	Log floor
Our fixed count	20	0.9305	1.12	12.81	0.00	34.09	279.24	0
HELP predictor	20	0.9366	18.54	12.78	15.37	55.81	3115.74	1
Our fixed count	28	0.9341	1.18	12.84	0.00	36.52	315.54	0
HELP predictor	28	0.9343	18.47	13.01	15.37	62.14	3790.21	0
Our fixed count	36	0.9334	1.18	12.63	0.00	37.66	330.18	0
HELP predictor	36	0.9357	22.64	12.28	15.37	39.28	3375.82	0
Our fixed count	42	0.9328	1.18	12.19	0.00	37.28	327.88	0
HELP predictor	42	0.9358	22.64	12.28	15.37	36.92	3374.67	0

Our fixed-count procedure uses posterior geometry. Quality comparisons use the original prediction records, including the original HELP scoring conventions. The millisecond and relative-MAE columns are in Table C3.5.

At ten measurements, the fixed-count procedure has a smaller mean fastest-selection penalty and smaller relative-error tails than HELP10, while HELP10 has the smaller mean log-MAE. The pointwise intervals in the next table show the uncertainty of these paired differences.

Paired uncertainty for the additional quality metrics Table C3.21 uses the same two examples already discussed in the main text: fixed-count acquisition at ten measurements and original adaptive acquisition at c $\approx$ 0.003153. The latter is compared with both all-42 and HELP10. These paired intervals keep the four-device grouping and the original metric definitions.

Table C3.21. Representative paired quality differences and pointwise 90% intervals.

Metric	Fixed 10 minus HELP10	Adaptive minus all-42	Adaptive minus HELP10
log-MAE	+0.02486 [0.01128, 0.03981]	+0.03185 [0.00828, 0.05968]	+0.04088 [0.00709, 0.07862]
Relative MAE (pp)	-7.86 [-19.65, 1.84]	+4.04 [0.79, 7.92]	-4.98 [-13.46, 2.13]
MAE (ms)	+0.778 [0.109, 1.593]	+0.670 [0.111, 1.360]	+1.100 [0.068, 2.353]
Spearman ρ	-0.0006 [-0.0075, 0.0063]	+0.0015 [-0.0058, 0.0089]	-0.0026 [-0.0143, 0.0104]
Fastest-selection penalty (pp)	-14.88 [-31.04, 0.47]	+50.00 [0.00, 111.47]	+29.94 [-8.20, 75.35]
Selected extra latency (ms)	-0.434 [-0.837, 0.016]	+1.016 [0.000, 2.224]	+0.483 [-0.496, 1.574]
Selected log latency ratio	-0.0981 [-0.2022, 0.0046]	+0.1985 [0.0000, 0.4371]	+0.0563 [-0.0663, 0.1908]
Mean-five shortlist penalty (pp)	-4.33 [-11.50, 1.81]	-1.04 [-3.64, 1.35]	-2.01 [-4.53, 0.01]
Best-in-five penalty (pp)	-15.37 [-34.59, 0.00]	+15.37 [0.00, 34.59]	+0.00 [0.00, 0.00]
Mean device relative p95 (pp)	-24.01 [-55.61, 1.79]	-2.63 [-9.03, 2.89]	-24.04 [-54.99, 1.27]
Mean device maximum error (pp)	-865.48 [-1954.14, 6.60]	+209.12 [0.55, 469.80]	-713.25 [-1606.46, 1.53]

Positive Spearman differences favor our procedure; negative differences favor it for the other metrics. pp means percentage points. The last row compares means of device maxima, not the global worst error reported in Tables C3.19–C3.20. Simultaneous log-MAE intervals for these families are in Tables C3.3, C3.6 and C3.7; no simultaneous claim is made for the other metrics.

C.3.6 Computation time and acquisition-time scenarios

The timing comparison measures the computation needed to obtain each procedure's final predictions on the same CPU host, apart from the time needed to collect the responses. Appendix B.5 defines the timer, the two passes of three repeats, and the task/device averaging. All methods use their existing trained models and selected settings. Each timed call executes its own measurement choices, adaptation or stopping calculations.

Procedure time begins with visible task inputs prepared. Preparation-inclusive time also includes our task-specific initial-error calculation, lookup of source predictions, kernel construction and input checks. HELP's corresponding input processing is already in its procedure time. Both exclude offline training, model/file loading and physical measurement waits. We use preparation-inclusive time for the main comparisons and retain both scopes below.

Table C3.22. Representative same-host computation times.

Procedure	Mean k	Procedure (ms)	Including task preparation (ms)	log-MAE
HELP10	10.00	34.285	34.285	0.11315
Our fixed count 10	10.00	11.815	13.901	0.13801
Adaptive, c $\approx$ 0.000914	7.08	89.495	91.581	0.15012
Adaptive, c $\approx$ 0.002087	5.08	69.013	71.098	0.15309
Adaptive, c $\approx$ 0.003153	4.58	63.694	65.779	0.15404
Direct all-42	42.00	1.519	3.604	0.12218
No acquisition	0.00	0.540	2.626	0.48348

These are means of six timed repetitions per task, then three tasks per device and four devices equally. The measured time is elapsed CPU-host computation, not process CPU time or actual device profiling. No timing confidence intervals were estimated.

Table C3.23. All seven same-count computation comparisons.

k	Our procedure (ms)	Our preparation-inclusive time (ms)	HELP (ms)	HELP / our inclusive time
10	11.815	13.901	34.285	2.47
12	13.840	15.925	35.176	2.21
16	18.049	20.135	36.717	1.82
20	21.877	23.962	38.436	1.60
28	28.867	30.953	41.711	1.35
36	35.246	37.331	45.041	1.21
42	39.726	41.811	47.394	1.13

Our fixed-count calculation is shorter at all seven mean comparisons. At 42 measurements, the direct all-42 procedure in Table C3.22 skips the ordering calculation and is therefore a separate, faster reference. Equal measurement count does not imply equal prediction quality (Table C3.5).

Complete adaptive timing curve The adaptive calculation adds a stopping-value assessment to measurement choice and correction. The three settings labeled in Figure 4 take longer to compute than HELP10, while acquiring fewer responses. Table C3.24 retains every cost rather than choosing a new setting from its measured runtime.

For the illustrative totals, every measurement is assumed to take 100 ms, collected sequentially without overlap with computation. Total time is preparation-inclusive computation plus 0.1 seconds times the acquired count, as in Equation (B.3). The HELP10 and direct all-42 totals under

this assumption are 1.034285 and 4.203604 seconds. These are scenarios, not measured profiling durations.

Table C3.24. All adaptive times and 100 ms serial-measurement scenarios.

Index	Mean k	Procedure (ms)	Including task preparation (ms)	Total at 100 ms (s)	Total-time reduction vs HELP10 (%)	Total-time reduction vs all-42 (%)
0	21.00	206.846	208.932	2.3089	-123.24	45.07
1	17.50	182.247	184.333	1.9343	-87.02	53.98
2	16.00	170.109	172.195	1.7722	-71.34	57.84
3	13.67	149.773	151.859	1.5185	-46.82	63.88
4	12.50	139.243	141.329	1.3913	-34.52	66.90
5	10.50	121.360	123.446	1.1734	-13.45	72.08
6	8.75	104.624	106.710	0.9817	5.08	76.65
7	7.83	96.071	98.157	0.8815	14.77	79.03
8	7.08	89.495	91.581	0.7999	22.66	80.97
9	6.42	82.677	84.763	0.7264	29.77	82.72
10	5.08	69.013	71.098	0.5794	43.98	86.22
11	4.58	63.694	65.779	0.5241	49.33	87.53
12	3.92	56.826	58.911	0.4506	56.44	89.28
13	3.50	52.208	54.293	0.4043	60.91	90.38
14	3.08	48.126	50.212	0.3585	65.33	91.47
15	2.42	40.780	42.866	0.2845	72.49	93.23
16	2.17	37.809	39.894	0.2566	75.19	93.90
17	1.58	31.816	33.901	0.1922	81.41	95.43
18	1.33	29.134	31.220	0.1646	84.09	96.09
19	1.17	27.075	29.161	0.1458	85.90	96.53
20	1.08	26.071	28.157	0.1365	86.80	96.75
21	0.75	21.970	24.055	0.0991	90.42	97.64
22	0.00	12.587	14.672	0.0147	98.58	99.65
23	0.00	12.552	14.638	0.0146	98.58	99.65
24	0.00	12.522	14.608	0.0146	98.59	99.65
25	0.00	12.612	14.698	0.0147	98.58	99.65

Total-time reduction (%) is 100 × (1 - adaptive mean total time / reference mean total time). Negative percentages indicate longer total time than the stated reference under the same 100 ms-per-measurement assumption. Mean task preparation contributes 2.086 ms to each adaptive setting. Prediction error remains the value for the same index in Table C3.2; the assumed measurement duration does not change that error or the chosen measurements.

How long must a measurement take to offset extra computation? When the adaptive procedure acquires fewer responses than HELP10, the difference in mean computation time can be divided by the mean number of responses saved. This gives the measurement duration at which the two mean totals agree. A positive value is a break-even threshold; a negative algebraic value means that the adaptive computation is already shorter even when collection takes zero time. The physical threshold is then zero.

Using preparation-inclusive time, the three settings labeled in Figure 4 break even with HELP10 at 19.64, 7.49 and 5.81 ms per measurement. Their total-time reductions in the 100 ms scenario are 22.66%, 43.98% and 49.33%, respectively. Each comparison retains the higher adaptive log-MAE reported in Table C3.7.

Table C3.25. HELP10 comparison: computation overhead and algebraic break-even times.

Index	Responses saved (10 - mean k)	Δprocedure time (ms)	Δinclusive time (ms)	Break-even: procedure (ms)	Break-even: inclusive (ms)
0	-11.00	+172.561	+174.647	—	—
1	-7.50	+147.961	+150.047	—	—
2	-6.00	+135.824	+137.909	—	—

Index	Responses saved (10 - mean k)	Δprocedure time (ms)	Δinclusive time (ms)	Break-even: procedure (ms)	Break-even: inclusive (ms)
3	-3.67	+115.488	+117.574	—	—
4	-2.50	+104.957	+107.043	—	—
5	-0.50	+87.075	+89.160	—	—
6	1.25	+70.339	+72.424	56.271	57.939
7	2.17	+61.786	+63.872	28.517	29.479
8	2.92	+55.210	+57.296	18.929	19.644
9	3.58	+48.392	+50.478	13.505	14.087
10	4.92	+34.727	+36.813	7.063	7.487
11	5.42	+29.408	+31.494	5.429	5.814
12	6.08	+22.540	+24.626	3.705	4.048
13	6.50	+17.922	+20.008	2.757	3.078
14	6.92	+13.841	+15.927	2.001	2.303
15	7.58	+6.495	+8.581	0.856	1.132
16	7.83	+3.523	+5.609	0.450	0.716
17	8.42	-2.470	-0.384	-0.293	-0.046
18	8.67	-5.151	-3.066	-0.594	-0.354
19	8.83	-7.210	-5.124	-0.816	-0.580
20	8.92	-8.214	-6.129	-0.921	-0.687
21	9.25	-12.316	-10.230	-1.331	-1.106
22	10.00	-21.699	-19.613	-2.170	-1.961
23	10.00	-21.733	-19.648	-2.173	-1.965
24	10.00	-21.763	-19.677	-2.176	-1.968
25	10.00	-21.673	-19.588	-2.167	-1.959

All Δ times are adaptive minus HELP10. A dash means that the adaptive policy does not acquire fewer responses, so this fewer-measurements threshold is not defined. Negative break-even values describe an advantage already present at zero collection time, not a negative physical duration. Thresholds are ratios of aggregate differences, not averages of taskwise ratios.

Device-level computation and conditional totals Table C3.26 shows each test device at the main-text settings and references. Computation is performed on the common CPU host; the device name identifies whose architecture latencies are being predicted. It does not name a processor used to execute the prediction algorithm.

Table C3.26. Per-device timing results at the same existing settings.

Test device	Procedure / cost c	Mean k	Procedure (ms)	Including preparation (ms)	log-MAE	Total at 100 ms (s)
Titan RTX	HELP10	10.00	34.355	34.355	0.03394	1.0344
Titan RTX	Fixed 10	10.00	11.850	14.122	0.05201	1.0141
Titan RTX	Adaptive 0.000914	8.67	108.280	110.552	0.04725	0.9772
Titan RTX	Adaptive 0.002087	5.67	78.265	80.537	0.05726	0.6472
Titan RTX	Adaptive 0.003153	5.00	71.076	73.348	0.06071	0.5733
Titan RTX	All-42	42.00	1.508	3.780	0.03921	4.2038
Xeon Gold 6226	HELP10	10.00	34.234	34.234	0.05026	1.0342
Xeon Gold 6226	Fixed 10	10.00	11.847	13.861	0.07006	1.0139
Xeon Gold 6226	Adaptive 0.000914	9.67	118.484	120.498	0.06924	1.0872
Xeon Gold 6226	Adaptive 0.002087	7.00	90.725	92.739	0.07045	0.7927
Xeon Gold 6226	Adaptive 0.003153	6.33	84.243	86.257	0.07036	0.7196
Xeon Gold 6226	All-42	42.00	1.518	3.531	0.06119	4.2035
Pixel 2	HELP10	10.00	34.292	34.292	0.13917	1.0343
Pixel 2	Fixed 10	10.00	11.784	13.667	0.14366	1.0137
Pixel 2	Adaptive 0.000914	6.33	83.942	85.824	0.13216	0.7192
Pixel 2	Adaptive 0.002087	4.00	59.814	61.697	0.13283	0.4617
Pixel 2	Adaptive 0.003153	3.33	52.204	54.087	0.13326	0.3874
Pixel 2	All-42	42.00	1.518	3.401	0.12765	4.2034
Raspberry Pi 4	HELP10	10.00	34.261	34.261	0.22924	1.0343
Raspberry Pi 4	Fixed 10	10.00	11.780	13.955	0.28632	1.0140
Raspberry Pi 4	Adaptive 0.000914	3.67	47.276	49.451	0.35182	0.4161

Test device	Procedure / cost c	Mean k	Procedure (ms)	Including preparation (ms)	log-MAE	Total at 100 ms (s)
Raspberry Pi 4	Adaptive 0.002087	3.67	47.247	49.421	0.35182	0.4161
Raspberry Pi 4	Adaptive 0.003153	3.67	47.251	49.425	0.35182	0.4161
Raspberry Pi 4	All-42	42.00	1.530	3.705	0.26068	4.2037

Each row averages three tasks after averaging six timing repetitions within task. The conditional total uses the same 100 ms serial-measurement assumption. No time was supplied for the development-selected fixed count of five, so the stopping comparison at that count is not converted into a timing claim.

Together, these comparisons distinguish three quantities: the prediction error obtained, the number of responses acquired, and the computation required to choose and use those responses. Fewer measurements can offset more computation when measurements take long enough to obtain, but the associated prediction-quality differences remain part of the comparison.

C.4 Value Estimates and Realized Gains

Checking the value calculation. This appendix gives the detailed value diagnostics used in Section 4.5. We compare the gain estimated before an acquisition with the realized gain from the same continuation. We retain the fitted models, development-selected settings and original measurement paths. This diagnostic evaluates both accepted and rejected continuations; it does not use its outcomes to change the policy.

Matching the continuation. At a history h, the continuation in Section 3.6 acquires the proposal from posterior geometry and then uses its observed response to decide whether to acquire one further response. We apply that same one-further-acquisition decision and fix the resulting history before scoring its predictions on Q. If only one candidate remains, the continuation ends after that acquisition. We do not choose the endpoint with the lower realized error, or restart two-step planning and follow the policy to its eventual stop.

The estimated net gain is $\widehat{G}_2(h;\lambda) \;=\; \widehat{R}_F(h) \;-\; \widehat{A}_2(h;\lambda)$.

For the same continuation, we calculate

$$G_{2,realized}(h;\lambda) \;=\; L_Q(F(h)) \;-\; L_Q(F(h_{end})) \;-\; \lambda N_{add}.$$

Here N_{add} is the one or two additional responses acquired, and both losses are evaluated on the same prediction set. A positive net gain means that the reduction in prediction error exceeds the acquisition penalty. ChEMBL gains use pKi MAE units. For HELP, losses, penalties and gains are converted together to natural-log latency units; the reported cost is c, as defined in Section 4.1 and Appendix B.3.3. The diagnostic tables use c for both applications; the original ChEMBL policy has c = λ.

Purchase and stopping states. We distinguish histories at which the original policy acquires a response from its first stopping history with candidates still available. Pool exhaustion is excluded because no further continuation is possible. Within each group, we average eligible histories within each task, then tasks within each protein or device, and finally give the included proteins or devices equal weight. The purchase and stopping groups differ in depth and composition; their means are not a randomized comparison of buying versus stopping.

Gains from accepted continuations. Table C4.1 uses the three ChEMBL settings labeled in Figure 2 and the two higher-cost HELP settings labeled in Figure 4. At their purchase states, mean realized net gains are positive in both applications and exceed the corresponding estimates. Across the full grids, mean realized gain is positive at 30 of the 31 ChEMBL penalties with a nonempty purchase group and at all 22 such HELP penalties. Empty purchase groups are not assigned zero gain.

Table C4.1. Estimated and realized net gains for the same continuation.

Application and cost c	Purchase states		Stopping states	
	Estimated	Realized	Estimated	Realized
ChEMBL: c ≈ 0.00158	+0.037795	+0.063306	-0.001177	+0.000600
ChEMBL: c ≈ 0.00250	+0.042516	+0.070404	-0.001527	+0.006226
ChEMBL: c ≈ 0.00396	+0.050160	+0.081421	-0.001822	+0.010519
HELP: c ≈ 0.00209	+0.087372	+0.112772	-0.020323	-0.066283
HELP: c ≈ 0.00315	+0.093478	+0.119838	-0.021518	-0.067521

Positive values favor the specified continuation over stopping at its starting history. ChEMBL values are in pKi MAE units and HELP values in log-MAE units, so their magnitudes should not be compared across applications. Each ChEMBL purchase group contains 45 tasks from 15 proteins; the stopping groups contain 44, 45 and 45 tasks, respectively, still covering all 15 proteins. Each HELP group contains 12 tasks from four devices. Entries are descriptive means, not confidence limits.

Gains remaining after stopping. For ChEMBL, the lowest-penalty example leaves little mean net gain after stopping, whereas positive gains remain at the other two settings. Across the full ChEMBL grid, rejected continuations have positive mean realized gain at 24 of 34 penalties. This is compatible with stopping before some useful acquisitions along the evaluated continuation. In HELP, both displayed stopping groups instead have negative mean realized gains, and this direction holds at 25 of 26 penalties. In those group means, declining the proposed continuation is beneficial. The agreement is in average direction, however, not in the numerical size of the estimated gain.

The optional second acquisition. Keeping the first proposed measurement fixed, we also compare allowing the modeled optional second acquisition with stopping after the first. At the purchase states in Table C4.1, its mean realized additional net gain is positive: 0.02139–0.02720 in ChEMBL and 0.05034–0.05183 in HELP. This benefit is not uniform. It is negative at the two less expensive ChEMBL stopping settings and at both HELP stopping settings shown. These are local continuation comparisons, not a comparison of complete one-step and two-step policies.

Interpretation and scope. These are retrospective diagnostics on the existing test tasks. A realized gain uses one response sequence and a finite prediction set; it is not the unknown conditional expected gain. Price settings and overlapping continuations are not independent replications, and their gains cannot be added to recover total policy performance. Table C4.1 gives the representative continuation comparisons; Appendix C.6.5 reports the related score-ordering analyses. Section 4.5 uses the relevant contrasts to interpret the measurement-selection and stopping results.

Local gains and final comparisons. The local gains above are not additive contributions to the final adaptive–fixed difference. For the HELP example c ≈ 0.00315, accounting along the saved path gives a final benefit of 0.056066 log-MAE units for adaptive stopping: 0.007508 from correction with the current option, 0.047244 from option reselection, and 0.001314 from acquisition cost. The sign here is fixed minus adaptive. This sequential accounting reproduces the observed endpoint difference; it is not the performance effect of removing reselection or another component. Appendix C.6.7 and Table C6.11 give this endpoint accounting. Table C3.17 reports the per-device full-grid differences between final policies.

C.5 Acquisition Valuation and Prediction Updates

This section expands the valuation comparison in Section 4.3 and examines the prediction changes made after one more response. The first analysis changes how a measurement is valued while keeping its eventual use fixed. The second separates correction of the previously selected prediction option from the additional change after option reselection.

C.5.1 Two valuation rules, one final prediction procedure

We compare two ways to estimate the value of further measurements. **F-based valuation (A)** evaluates the expected absolute error of the selected-and-corrected predictions $F(h)$. **Posterior-median valuation (U)** instead evaluates the expected absolute error of median predictions from the same

outcome model. These medians minimize expected absolute loss under that model. The acquisition policies using A and U both return predictions from F; median predictions are used only in U's acquisition-value calculation. The ChEMBL source predictions, outcome model and posterior-geometry proposal are shared.

We use A and U as compact labels in the tables. At history h, the F-based and posterior-median risks are

$$\begin{aligned} \widehat{R}_A(h) &= \mathbb{E}_{\widehat{P}_h}\left[L_{\mathcal{Q}}\left(F(h); Y_{\mathcal{Q}}\right)\right], \\ \widehat{R}_U(h) &= \min_{f \in \mathbb{R}^{|\mathcal{Q}|}} \mathbb{E}_{\widehat{P}_h}\left[L_{\mathcal{Q}}\left(f; Y_{\mathcal{Q}}\right)\right]. \end{aligned} \tag{C5.1}$$

The second risk is attained by a median of each full Gaussian-mixture marginal on the prediction set. It is not a variance score or an average of the component medians. We find the point where each marginal mixture cumulative distribution equals one-half, using a safeguarded Newton iteration bracketed by the component means. Expected absolute loss is then evaluated as in Appendix A.4.

A and U each use their own risk at the current history and at every hypothetical history in the continuation calculation. The two valuation calculations retain the three quadrature nodes per component, the response-dependent second posterior-geometry proposal and the option to stop after the first response. Only one response is acquired before planning again, as in Section 3.6. With one candidate left, the continuation has one acquisition; an exhausted pool ends the episode.

The acquisition policies using F-based valuation (A) and posterior-median valuation (U) both return $F(h)$. At the same task and measurement count, their histories, outcome distributions, proposals and final predictions agree. They can differ in when they stop and how measurements are allocated across tasks. This comparison therefore changes acquisition valuation, not the deployed predictor.

The study reuses the 57 development proteins and 513 tasks, and the 15 test proteins and 45 tasks in Appendix B. Each task has 15 acquisition candidates and 15 separate prediction cases. The final prediction settings remain $\alpha = 0.1$ and $s = 0.25$. No test responses enter preparation or setting selection; the existing test benchmark is reused rather than replaced by a new population.

C.5.2 Gain calibration and development selection

For F-based valuation (A) and posterior-median valuation (U), we evaluate two versions: one uses the modeled gains directly, labelled **raw**, and the other calibrates those gains using development data. This calibration is separate from the residual correction used to predict responses: it changes the acquisition score, not the source forests, option-selection rule or corrected output F.

For a proposed one-response transition, let g_i be the modeled risk reduction for the valuation rule, u_i the realized reduction in the MAE of F, and k_i the current measurement count. For A and U, we fit separate coefficients using the same three-coefficient weighted regression:

$$\min_{\beta \geq 0,\, \eta_0,\, \eta_1} \ \sum_i w_i \left[u_i - (\beta g_i + \eta_0 + \eta_1 k_i)\right]^2. \tag{C5.2}$$

Weights give equal importance to proteins, environments, tasks and potential measurement depths. Rank-deficient fits use the deterministic minimum-norm solution. We fit separate calibration coefficients for A and U, using the same target: the realized one-response reduction in the MAE of F on development tasks. For U, the modeled gain is based on posterior-median predictions, but the calibration target is the observed improvement of *F*, not the observed improvement of the median predictions. This calibration changes U's acquisition-gain estimates, not the final prediction rule.

To use these coefficients consistently at every depth, we replace the corresponding risk by the score

$$S_v(h) = \beta_v \widehat{R}_v(h) - \eta_{0,v} k - \tfrac{1}{2}\eta_{1,v} k(k-1), \qquad v \in \{A, U\}. \tag{C5.3}$$

The modeled one-response reduction of this score is $\beta_v g_v(h) + \eta_{0,v} + \eta_{1,v} k$. We use the score throughout the same first-plus-optional-second calculation and still charge the controller penalty λ for each acquisition. Scores are not clipped to zero. Raw valuation corresponds to coefficients $(1, 0, 0)$.

Calibration follows the existing six-outer/five-inner protein splits. For an outer-held group, its calibrator is fitted only on inner-held paths within the outer training group, using the corresponding inner outcome models and that outer fold's selected prediction settings. It is evaluated with the outer

model on the held-out proteins. Thus, calibration does not use a prediction from a model that saw the protein currently being validated.

The final calibration uses all 57 development proteins' cross-fitted outcome-model paths with the final selected prediction settings: 7,695 potential one-response transitions. These final coefficients are used only when the calibrated variant is selected for testing.

Table C5.1. Final development-fitted gain-calibration coefficients.

Valuation rule	β	η_0	η_1
A: F-based	0.8125074809	0.006174011762	-0.0003813058157
U: posterior-median	0.7858674681	0.008842748704	-0.0006833354762

These coefficients belong to acquisition-gain calibration. They are not the source-model biases or the residual-correction coefficient α in Section 3.

C.5.3 Selecting settings and evaluating outcomes

The controller penalty λ determines the purchase decision, whereas the evaluation cost c determines the reported error-plus-acquisition loss $J_c = L_{\mathcal{Q}}\left(F\left(h_{\text{stop}}\right); y_{\mathcal{Q}}\right) + ck$, where k is the final number acquired. We start with the 34 penalties in Appendix B.3.3 and check their mean acquisition counts on development data. If two neighboring penalties differ by more than one acquired response on average for any A/U and raw/calibrated combination, we add their geometric midpoint. This single refinement adds six penalties, giving a common set of 40 controller settings for A and U. Evaluation still uses the original 34 costs.

At each evaluation cost, development results select either a raw setting from the 40 candidates, or a variant and setting from the 80 raw/calibrated candidates, separately for A and U. Selection minimizes mean J_c; exact ties prefer fewer mean acquisitions, then raw valuation, then smaller λ and its index. The selected setting is fixed across test tasks at that cost, but the acquired count remains task-dependent. The comparison is not restricted to $\lambda = c$.

Differences are calculated within the same test tasks, averaged over the three tasks within each protein, and then over proteins equally. For grid averages, all 34 evaluation costs receive equal weight. They are repeated evaluations of the same proteins, not independent trials.

Uncertainty uses 10,000 paired whole-protein bootstrap resamples. Fits, calibrations and development-selected settings are held fixed. All intervals reported here use a 90% confidence level. Pointwise limits are the 5th and 95th percentiles. For each contrast and outcome, the simultaneous band uses the 90th percentile of the largest absolute centered bootstrap deviation across the 34 costs after division by each point's bootstrap standard error; its radius at each cost is that critical value times the same standard error. Costs with bootstrap standard error at most 10^{-14} contribute zero to the standardized maximum; the band at each cost remains the mean plus or minus the critical value times its standard error. It does not jointly cover every comparison and outcome.

The raw grid-average J_c difference is +0.00251, with the 90% interval [-0.00136, 0.00670] reported in Section 4.3. This is a single aggregate, not a curve band. Table C5.2 reports the 90% intervals for all three grid-average comparisons. None establishes an overall cost-inclusive advantage for F-based valuation; intervals containing zero do not establish equivalence.

Table C5.2. Differences averaged equally over the 34 evaluation costs.

Comparison: A minus U	**ΔMAE (pKi) [90% interval]**	**Δmeasurements [90% interval]**	**ΔJc (pKi) [90% interval]**
Raw A - raw U	+0.005770 [0.001848, 0.010187]	-0.281046 [-0.688889, 0.016993]	+0.002508 [-0.001363, 0.006698]
Calibration allowed for both rules	+0.004938 [0.001107, 0.009278]	+0.086928 [-0.153627, 0.270588]	+0.001690 [-0.002140, 0.005887]
Raw A - calibration allowed for U	+0.005700 [0.001582, 0.010136]	-0.286928 [-0.699379, 0.013072]	+0.001954 [-0.002360, 0.006357]

"Calibration allowed" means that development data choose between raw and calibrated valuation together with λ; it does not require the calibrated variant. Negative differences favor A for each displayed outcome. The third row is a supplementary asymmetric comparison, not the raw comparison in Section 4.3.

C.5.4 Complete controller-setting curves

Tables C5.3–C5.4 show every controller setting before development-based selection. Measurement reduction is $100\left(1-\overline{k}/15\right)$. All-15 and no acquisition are separate references, not controller settings.

Table C5.3. Uncalibrated valuation: all 40 controller settings.

Controller index	λ (pKi/response)	A mean k	A MAE (pKi)	U mean k	U MAE (pKi)
0	0	13.53	0.46668	15.00	0.46495
1	0.00025	13.04	0.46645	14.89	0.46505
2	0.0003147313529	13.00	0.46640	14.82	0.46490
3	0.0003962232981	12.89	0.46586	14.71	0.46511
4	0.0004988155787	12.76	0.46620	14.53	0.46560
5	0.0006279716079	12.49	0.46695	14.20	0.46586
6	0.000790569415	12.36	0.46722	13.78	0.46752
7	0.0009952679264	11.98	0.46802	13.24	0.47025
8	0.001252968084	11.49	0.47023	12.62	0.46968
9	0.001577393361	10.71	0.46817	11.49	0.46878
10	0.001985820587	9.82	0.47542	10.71	0.46877
11	0.0025	9.07	0.47877	9.69	0.47697
12	0.003147313529	8.09	0.48251	8.82	0.48007
13*	0.003531343862	7.51	0.48933	8.18	0.48684
14	0.003962232981	7.20	0.49110	7.80	0.48574
15*	0.004445698525	7.00	0.49420	7.49	0.48300
16	0.004988155787	6.78	0.49865	6.91	0.48750
17*	0.005596802846	6.16	0.50324	6.47	0.49361
18	0.006279716079	5.47	0.52027	6.04	0.50744
19*	0.007045957328	5.24	0.52706	5.60	0.51454
20	0.00790569415	4.84	0.53231	5.02	0.52710
21*	0.008870334731	4.60	0.53373	4.76	0.52663
22	0.009952679264	4.47	0.53197	4.56	0.52934
23*	0.0111670898	4.20	0.53433	4.29	0.53339
24	0.01252968084	3.84	0.54210	3.96	0.54110
25	0.01577393361	3.29	0.56777	3.36	0.56753
26	0.01985820587	2.89	0.59600	2.96	0.58618
27	0.025	2.42	0.61336	2.53	0.60665
28	0.03147313529	2.20	0.63695	2.20	0.63946
29	0.03962232981	1.87	0.64957	1.93	0.64240
30	0.04988155787	1.36	0.71544	1.44	0.70329
31	0.06279716079	1.09	0.73064	1.09	0.72522
32	0.0790569415	1.04	0.73185	1.04	0.73185
33	0.09952679264	1.00	0.73036	1.00	0.73036
34	0.1252968084	0.96	0.73714	0.98	0.73626
35	0.1577393361	0.69	0.78728	0.73	0.78609
36	0.1985820587	0.33	0.82789	0.33	0.83625
37	0.25	0.00	0.91822	0.00	0.91822
38	0.35	0.00	0.91822	0.00	0.91822
39	0.5	0.00	0.91822	0.00	0.91822
All-15	—	15.00	0.46495	15.00	0.46495
None	—	0.00	0.91822	0.00	0.91822

A: F-based valuation; U: posterior-median valuation. Means give the 15 test proteins equal weight. * marks the six settings added using development counts. Controller indices run from 0 to 39 and differ from evaluation-cost indices 0–33. Values are not reoptimized on test results. These tables

report means; Tables C5.6 and C5.8 give paired cost-inclusive intervals for the development-selected comparisons.

Calibrated controller-setting curves This table uses the final calibration coefficients in Table C5.1 at all 40 controller settings, whether or not a setting is selected during development. It is not the calibration-allowed comparison, which may select a raw setting instead.

Table C5.4. Calibrated valuation: all 40 controller settings.

Controller index	λ (pKi/response)	A mean k	A MAE (pKi)	U mean k	U MAE (pKi)
0	0	14.56	0.46253	14.16	0.46692
1	0.00025	14.42	0.46321	14.02	0.46707
2	0.0003147313529	14.42	0.46321	14.00	0.46775
3	0.0003962232981	14.42	0.46321	13.91	0.46711
4	0.0004988155787	14.31	0.46374	13.78	0.46674
5	0.0006279716079	14.24	0.46376	13.64	0.46660
6	0.000790569415	14.22	0.46373	13.49	0.46766
7	0.0009952679264	13.91	0.46449	13.20	0.46737
8	0.001252968084	13.71	0.46641	13.04	0.46706
9	0.001577393361	13.29	0.46938	12.62	0.46768
10	0.001985820587	12.89	0.46905	12.22	0.46636
11	0.0025	12.20	0.47139	11.53	0.46889
12	0.003147313529	11.27	0.46563	11.00	0.47041
13*	0.003531343862	10.82	0.46835	10.60	0.47035
14	0.003962232981	10.11	0.47155	10.24	0.47378
15*	0.004445698525	9.56	0.47056	9.69	0.47498
16	0.004988155787	9.18	0.47348	9.31	0.47050
17*	0.005596802846	8.31	0.48068	8.87	0.47611
18	0.006279716079	7.82	0.48325	8.33	0.48153
19*	0.007045957328	7.00	0.49364	7.73	0.48512
20	0.00790569415	6.53	0.50176	7.27	0.49038
21*	0.008870334731	5.89	0.51106	6.67	0.49249
22	0.009952679264	5.24	0.52230	6.07	0.50007
23*	0.0111670898	4.69	0.53034	5.36	0.52060
24	0.01252968084	4.53	0.53214	4.84	0.52502
25	0.01577393361	3.69	0.54445	4.09	0.54038
26	0.01985820587	3.02	0.58591	3.36	0.56753
27	0.025	2.49	0.61366	2.76	0.59124
28	0.03147313529	2.20	0.63695	2.22	0.63309
29	0.03962232981	1.78	0.67044	1.91	0.65682
30	0.04988155787	1.29	0.71942	1.31	0.71182
31	0.06279716079	1.07	0.73289	1.07	0.72960
32	0.0790569415	1.02	0.73151	1.02	0.73151
33	0.09952679264	0.98	0.73626	1.00	0.73036
34	0.1252968084	0.87	0.73522	0.87	0.73919
35	0.1577393361	0.40	0.84401	0.38	0.84490
36	0.1985820587	0.00	0.91822	0.00	0.91822
37	0.25	0.00	0.91822	0.00	0.91822
38	0.35	0.00	0.91822	0.00	0.91822
39	0.5	0.00	0.91822	0.00	0.91822
All-15	—	15.00	0.46495	15.00	0.46495
None	—	0.00	0.91822	0.00	0.91822

A: F-based valuation; U: posterior-median valuation. Means give the 15 test proteins equal weight. * marks the six settings added using development counts. Controller indices run from 0 to 39 and differ from evaluation-cost indices 0–33. Values are not reoptimized on test results. These tables report means; Tables C5.6 and C5.8 give paired cost-inclusive intervals for the development-selected comparisons.

C.5.5 DEVELOPMENT-SELECTED RAW VALUATION

The following comparison is the one summarized in Section 4.3. Both rules use raw gains, but each receives its own development-selected controller penalty for each evaluation cost. A has lower mean J_c at 13 costs, U at 17, and the means are numerically tied at four. The 90% simultaneous band includes zero at every cost. Table C5.5 shows the selections and test outcomes; Table C5.6 shows their paired cost-inclusive differences.

Table C5.5. Development-selected settings and test outcomes: raw gains.

Cost index	c (pKi/response)	A λ index	U λ index	A mean k	U mean k	A MAE (pKi)	U MAE (pKi)
0	0	1	3	13.04	14.71	0.46645	0.46511
1	0.00025	1	3	13.04	14.71	0.46645	0.46511
2	0.000314731353	1	3	13.04	14.71	0.46645	0.46511
3	0.000396223298	1	6	13.04	13.78	0.46645	0.46752
4	0.000498815579	1	6	13.04	13.78	0.46645	0.46752
5	0.000627971608	2	7	13.00	13.24	0.46640	0.47025
6	0.000790569415	2	7	13.00	13.24	0.46640	0.47025
7	0.000995267926	2	8	13.00	12.62	0.46640	0.46968
8	0.00125296808	2	8	13.00	12.62	0.46640	0.46968
9	0.00157739336	2	8	13.00	12.62	0.46640	0.46968
10	0.00198582059	8	10	11.49	10.71	0.47023	0.46877
11	0.0025	10	10	9.82	10.71	0.47542	0.46877
12	0.00314731353	10	11	9.82	9.69	0.47542	0.47697
13	0.00396223298	12	11	8.09	9.69	0.48251	0.47697
14	0.00498815579	13	11	7.51	9.69	0.48933	0.47697
15	0.00627971608	13	16	7.51	6.91	0.48933	0.48750
16	0.00790569415	13	16	7.51	6.91	0.48933	0.48750
17	0.00995267926	18	18	5.47	6.04	0.52027	0.50744
18	0.0125296808	18	22	5.47	4.56	0.52027	0.52934
19	0.0157739336	21	22	4.60	4.56	0.53373	0.52934
20	0.0198582059	26	22	2.89	4.56	0.59600	0.52934
21	0.025	26	27	2.89	2.53	0.59600	0.60665
22	0.0314731353	26	27	2.89	2.53	0.59600	0.60665
23	0.0396223298	28	28	2.20	2.20	0.63695	0.63946
24	0.0498815579	30	30	1.36	1.44	0.71544	0.70329
25	0.0627971608	30	31	1.36	1.09	0.71544	0.72522
26	0.0790569415	31	31	1.09	1.09	0.73064	0.72522
27	0.0995267926	35	35	0.69	0.73	0.78728	0.78609
28	0.125296808	36	35	0.33	0.73	0.82789	0.78609
29	0.157739336	37	36	0.00	0.33	0.91822	0.83625
30	0.198582059	37	38	0.00	0.00	0.91822	0.91822
31	0.25	37	38	0.00	0.00	0.91822	0.91822
32	0.35	38	38	0.00	0.00	0.91822	0.91822
33	0.5	38	38	0.00	0.00	0.91822	0.91822

The cost index identifies c in Appendix B.3.3. Each λ index identifies a value in Tables C5.3–C5.4, not a cost index. All settings were selected on development data. A and U still return the same prediction rule F, but their stopping counts may differ. All selected variants are raw.

Cost-inclusive differences: raw gains We score both policies at the same evaluation cost: $\Delta J_c = (\mathrm{MAE}_A - \mathrm{MAE}_U) + c\,(k_A - k_U)$. Negative values favor A. Pointwise intervals concern one cost; the simultaneous band covers the fixed 34-cost curve for this comparison and this outcome only.

Table C5.6. All evaluation costs: A-minus-U loss differences, raw gains.

Cost index	c (pKi/response)	ΔJc (pKi)	90% pointwise interval	90% simultaneous band
0	0	+0.001336	[-0.002988, 0.005877]	[-0.006149, 0.008821]

Cost index	c (pKi/response)	ΔJc (pKi)	90% pointwise interval	90% simultaneous band
1	0.00025	+0.000919	[-0.003548, 0.005527]	[-0.006739, 0.008577]
2	0.000314731353	+0.000811	[-0.003685, 0.005436]	[-0.006894, 0.008517]
3	0.000396223298	-0.001359	[-0.007697, 0.004326]	[-0.011562, 0.008843]
4	0.000498815579	-0.001435	[-0.007838, 0.004257]	[-0.011724, 0.008855]
5	0.000627971608	-0.004003	[-0.012238, 0.003471]	[-0.017412, 0.009406]
6	0.000790569415	-0.004043	[-0.012362, 0.003488]	[-0.017591, 0.009505]
7	0.000995267926	-0.002904	[-0.011333, 0.004561]	[-0.016450, 0.010641]
8	0.00125296808	-0.002807	[-0.011405, 0.004744]	[-0.016569, 0.010955]
9	0.00157739336	-0.002685	[-0.011499, 0.004969]	[-0.016730, 0.011361]
10	0.00198582059	+0.003004	[-0.001418, 0.007311]	[-0.004415, 0.010422]
11	0.0025	+0.004429	[0.000044, 0.008894]	[-0.003083, 0.011940]
12	0.00314731353	-0.001126	[-0.006096, 0.004404]	[-0.010037, 0.007785]
13	0.00396223298	-0.000798	[-0.008204, 0.007267]	[-0.013888, 0.012292]
14	0.00498815579	+0.001496	[-0.008650, 0.012743]	[-0.016606, 0.019598]
15	0.00627971608	+0.005598	[-0.003826, 0.015422]	[-0.010651, 0.021846]
16	0.00790569415	+0.006573	[-0.002799, 0.016339]	[-0.009594, 0.022740]
17	0.00995267926	+0.007071	[-0.004132, 0.017815]	[-0.011427, 0.025568]
18	0.0125296808	+0.002339	[-0.012003, 0.017029]	[-0.022524, 0.027202]
19	0.0157739336	+0.005091	[-0.001838, 0.014228]	[-0.008750, 0.018932]
20	0.0198582059	+0.033561	[0.003605, 0.065701]	[-0.018591, 0.085714]
21	0.025	-0.001758	[-0.017860, 0.012234]	[-0.027332, 0.023815]
22	0.0314731353	+0.000543	[-0.015133, 0.014503]	[-0.024615, 0.025701]
23	0.0396223298	-0.002512	[-0.011073, 0.005972]	[-0.017197, 0.012172]
24	0.0498815579	+0.007720	[-0.000731, 0.016992]	[-0.008030, 0.023470]
25	0.0627971608	+0.006966	[-0.005944, 0.019940]	[-0.014873, 0.028805]
26	0.0790569415	+0.005419	[0.000000, 0.011011]	[-0.004522, 0.015359]
27	0.0995267926	-0.003239	[-0.010723, 0.001005]	[-0.013653, 0.007174]
28	0.125296808	-0.008317	[-0.065409, 0.049675]	[-0.105796, 0.089162]
29	0.157739336	+0.029398	[-0.086354, 0.184893]	[-0.203370, 0.262166]
30	0.198582059	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]
31	0.25	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]
32	0.35	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]
33	0.5	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]

The intervals use 10,000 paired whole-protein resamples with fitted models and development selections held fixed. The simultaneous construction follows the standardized-deviation procedure in C.5.3. Printed zero means agreement to six decimal places; numerical ties in the count summaries use an absolute difference of at most 10^{-12}.

C.5.6 Allowing development to choose gain calibration

Here development may select either raw or calibrated gains for each rule at each evaluation cost. A has lower mean J_c at 15 costs, U at 15, and four means are numerically tied. One cost favors A under the 90% simultaneous band. This is a separate comparison from the raw results: both the valuation variant and its controller penalty may change.

Table C5.7. Development-selected settings and test outcomes: calibration allowed.

Cost index	c (pKi/response)	A variant / λ index	U variant / λ index	A mean k	U mean k	A MAE (pKi)	U MAE (pKi)
0	0	cal 0	raw 3	14.56	14.71	0.46253	0.46511
1	0.00025	cal 0	raw 3	14.56	14.71	0.46253	0.46511
2	0.000314731353	cal 0	raw 3	14.56	14.71	0.46253	0.46511
3	0.000396223298	cal 0	raw 6	14.56	13.78	0.46253	0.46752
4	0.000498815579	cal 0	raw 6	14.56	13.78	0.46253	0.46752
5	0.000627971608	cal 0	raw 7	14.56	13.24	0.46253	0.47025
6	0.000790569415	cal 0	raw 7	14.56	13.24	0.46253	0.47025
7	0.000995267926	cal 0	raw 8	14.56	12.62	0.46253	0.46968
8	0.00125296808	cal 10	raw 8	12.89	12.62	0.46905	0.46968

Cost index	c (pKi/response)	A variant / λ index	U variant / λ index	A mean k	U mean k	A MAE (pKi)	U MAE (pKi)
9	0.00157739336	cal 10	raw 8	12.89	12.62	0.46905	0.46968
10	0.00198582059	cal 14	raw 10	10.11	10.71	0.47155	0.46877
11	0.0025	cal 14	raw 10	10.11	10.71	0.47155	0.46877
12	0.00314731353	cal 14	raw 11	10.11	9.69	0.47155	0.47697
13	0.00396223298	cal 15	raw 11	9.56	9.69	0.47056	0.47697
14	0.00498815579	raw 13	cal 18	7.51	8.33	0.48933	0.48153
15	0.00627971608	raw 13	cal 18	7.51	8.33	0.48933	0.48153
16	0.00790569415	raw 13	raw 16	7.51	6.91	0.48933	0.48750
17	0.00995267926	raw 18	raw 18	5.47	6.04	0.52027	0.50744
18	0.0125296808	raw 18	raw 22	5.47	4.56	0.52027	0.52934
19	0.0157739336	raw 21	raw 22	4.60	4.56	0.53373	0.52934
20	0.0198582059	raw 26	raw 22	2.89	4.56	0.59600	0.52934
21	0.025	raw 26	raw 27	2.89	2.53	0.59600	0.60665
22	0.0314731353	raw 26	raw 27	2.89	2.53	0.59600	0.60665
23	0.0396223298	raw 28	raw 28	2.20	2.20	0.63695	0.63946
24	0.0498815579	raw 30	cal 30	1.36	1.31	0.71544	0.71182
25	0.0627971608	raw 30	cal 30	1.36	1.31	0.71544	0.71182
26	0.0790569415	cal 31	raw 31	1.07	1.09	0.73289	0.72522
27	0.0995267926	raw 35	raw 35	0.69	0.73	0.78728	0.78609
28	0.125296808	cal 35	raw 35	0.40	0.73	0.84401	0.78609
29	0.157739336	raw 37	cal 35	0.00	0.38	0.91822	0.84490
30	0.198582059	raw 37	raw 38	0.00	0.00	0.91822	0.91822
31	0.25	raw 37	raw 38	0.00	0.00	0.91822	0.91822
32	0.35	raw 38	raw 38	0.00	0.00	0.91822	0.91822
33	0.5	raw 38	raw 38	0.00	0.00	0.91822	0.91822

The cost index identifies c in Appendix B.3.3. Each λ index identifies a value in Tables C5.3–C5.4, not a cost index. All settings were selected on development data. A and U still return the same prediction rule F, but their stopping counts may differ. raw: uncalibrated gains; cal: the calibrated score in Equation (C5.3).

At cost index 13 ($c \approx 0.003962$), development selects calibrated A and raw U. They acquire 9.56 and 9.69 responses with MAEs of 0.47056 and 0.47697 pKi. The A-minus-U MAE difference is -0.006411, with a 90% simultaneous interval of [-0.011013, -0.001808]. The cost-inclusive interval also favors A (Table C5.8), whereas the measurement-count interval includes zero. This is lower error at similar mean acquisition counts at one cost, not a general advantage of raw A or an isolated effect of calibration alone.

Cost-inclusive differences: calibration allowed We score both policies at the same evaluation cost: $\Delta J_c = (\mathrm{MAE}_A - \mathrm{MAE}_U) + c\,(k_A - k_U)$. Negative values favor A. Pointwise intervals concern one cost; the simultaneous band covers the fixed 34-cost curve for this comparison and this outcome only.

Table C5.8. All evaluation costs: A-minus-U loss differences, calibration allowed.

Cost index	c (pKi/response)	ΔJc (pKi)	90% pointwise interval	90% simultaneous band
0	0	-0.002581	[-0.005317, -0.000352]	[-0.006813, 0.001651]
1	0.00025	-0.002620	[-0.005486, -0.000316]	[-0.007025, 0.001785]
2	0.000314731353	-0.002630	[-0.005527, -0.000305]	[-0.007080, 0.001820]
3	0.000396223298	-0.004678	[-0.009687, -0.000719]	[-0.012298, 0.002943]
4	0.000498815579	-0.004598	[-0.009657, -0.000608]	[-0.012287, 0.003091]
5	0.000627971608	-0.006896	[-0.014062, -0.000404]	[-0.018561, 0.004769]
6	0.000790569415	-0.006683	[-0.013907, -0.000166]	[-0.018426, 0.005061]
7	0.000995267926	-0.005226	[-0.012601, 0.001573]	[-0.017381, 0.006930]
8	0.00125296808	-0.000299	[-0.004811, 0.004549]	[-0.008246, 0.007648]
9	0.00157739336	-0.000212	[-0.004725, 0.004633]	[-0.008167, 0.007742]
10	0.00198582059	+0.001583	[-0.002369, 0.005294]	[-0.004958, 0.008125]
11	0.0025	+0.001275	[-0.002809, 0.005051]	[-0.005432, 0.007981]

Cost index	c (pKi/response)	ΔJc (pKi)	90% pointwise interval	90% simultaneous band
12	0.00314731353	-0.004093	[-0.006973, -0.001202]	[-0.008988, 0.000802]
13	0.00396223298	-0.006939	[-0.010201, -0.003598]	[-0.012565, -0.001313]
14	0.00498815579	+0.003695	[-0.005778, 0.013066]	[-0.012347, 0.019738]
15	0.00627971608	+0.002634	[-0.006753, 0.011922]	[-0.013331, 0.018598]
16	0.00790569415	+0.006573	[-0.002799, 0.016339]	[-0.009606, 0.022752]
17	0.00995267926	+0.007071	[-0.004132, 0.017815]	[-0.011440, 0.025582]
18	0.0125296808	+0.002339	[-0.012003, 0.017029]	[-0.022542, 0.027221]
19	0.0157739336	+0.005091	[-0.001838, 0.014228]	[-0.008761, 0.018942]
20	0.0198582059	+0.033561	[0.003605, 0.065701]	[-0.018630, 0.085752]
21	0.025	-0.001758	[-0.017860, 0.012234]	[-0.027351, 0.023834]
22	0.0314731353	+0.000543	[-0.015133, 0.014503]	[-0.024634, 0.025720]
23	0.0396223298	-0.002512	[-0.011073, 0.005972]	[-0.017208, 0.012183]
24	0.0498815579	+0.005839	[-0.001277, 0.014460]	[-0.007548, 0.019226]
25	0.0627971608	+0.006413	[-0.001087, 0.015451]	[-0.007616, 0.020442]
26	0.0790569415	+0.005906	[0.000487, 0.011812]	[-0.003986, 0.015799]
27	0.0995267926	-0.003239	[-0.010723, 0.001005]	[-0.013661, 0.007182]
28	0.125296808	+0.016155	[-0.031183, 0.066905]	[-0.067676, 0.099985]
29	0.157739336	+0.013735	[-0.113278, 0.170667]	[-0.231176, 0.258645]
30	0.198582059	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]
31	0.25	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]
32	0.35	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]
33	0.5	+0.000000	[0.000000, 0.000000]	[0.000000, 0.000000]

The intervals use 10,000 paired whole-protein resamples with fitted models and development selections held fixed. The simultaneous construction follows the standardized-deviation procedure in C.5.3. Printed zero means agreement to six decimal places; numerical ties in the count summaries use an absolute difference of at most 10^{-12}.

C.5.7 Correction and option reselection on the same histories

The second analysis asks what changes in the predictions after a response is obtained. Let $a = \widehat{a}(h)$ be the option selected before the response, and let h^+ be the history after revealing the next posterior-geometry proposal. We first keep option a and update only its residual correction. We then reselect the option using the enlarged history.

$$\begin{aligned} g_{\text{corr}}(h) &= L_{\mathcal{Q}}\left(F(h); y_{\mathcal{Q}}\right) - L_{\mathcal{Q}}\left(f_a\left(h^+\right); y_{\mathcal{Q}}\right), \\ g_{\text{reselect}}(h) &= L_{\mathcal{Q}}\left(f_a\left(h^+\right); y_{\mathcal{Q}}\right) - L_{\mathcal{Q}}\left(F\left(h^+\right); y_{\mathcal{Q}}\right). \end{aligned} \tag{C5.4}$$

These two changes add exactly to the total one-response reduction in prediction error:

$$g_{\text{total}}(h) = g_{\text{corr}}(h) + g_{\text{reselect}}(h) = L_{\mathcal{Q}}\left(F(h); y_{\mathcal{Q}}\right) - L_{\mathcal{Q}}\left(F\left(h^+\right); y_{\mathcal{Q}}\right). \tag{C5.5}$$

Positive values indicate lower MAE and negative values indicate higher MAE. This calculation uses the same actual history for both updates. It excludes the acquisition charge and does not include an optional second response. The withheld prediction responses are used only for this retrospective evaluation.

The analysis covers every potential one-response transition on the 45 test tasks' complete posterior-geometry paths: counts $k = 0, \ldots, 14$, giving 675 transitions. It includes states beyond a particular adaptive policy's stopping point. We average depths within each task, tasks within environment, environments within protein, and proteins equally. It is therefore not an average restricted to responses actually purchased at a selected cost.

Table C5.9. Mean one-response improvement and frequency of option changes.

Quantity	Mean	90% interval	Unit
Total improvement	+0.030218	[0.019059, 0.043042]	pKi
Previous-option correction	+0.029887	[0.018921, 0.042398]	pKi
Additional option reselection	+0.000331	[-0.001000, 0.001631]	pKi

Quantity	Mean	90% interval	Unit
Option changes	20.00	[15.56, 24.59]	% of transitions

Intervals are paired whole-protein percentile intervals for the fixed paths. Option changes count a different selected option whether its additional effect is beneficial or harmful. This frequency is not an additive part of the MAE decomposition.

Most of the mean improvement in this ordered decomposition comes from updating the correction of the previously selected option. The mean additional effect of reselection is small, with an interval spanning zero, even though the selected option changes in 20% of transitions. The depth-specific results include both positive and negative reselection effects (Table C5.10).

The decomposition does not measure the value of the initial option choice, and it does not evaluate a policy that disables reselection throughout. The order of the accounting matters. In particular, it cannot be read as a causal contribution percentage for the complete method, or substituted for the HELP endpoint accounting in Appendix C.6.7. Appendix C.4 addresses the separate question of expected and realized gains from the original first-plus-optional-second continuation.

Depth-specific and protein-specific prediction updates Table C5.10 keeps each measurement depth separate. Table C5.11 averages all 15 depths and the three tasks within each test protein. These descriptive tables report means for all proteins and all potential depths.

Table C5.10. One-response changes by the number of measurements already acquired.

Measurements acquired, k	Total gain (pKi)	Correction gain (pKi)	Reselection gain (pKi)	Option changes (%)
0	+0.187868	+0.195746	-0.007878	64.44
1	+0.072735	+0.063206	+0.009530	33.33
2	+0.061130	+0.065374	-0.004244	33.33
3	+0.049368	+0.043258	+0.006110	31.11
4	+0.022986	+0.020198	+0.002788	22.22
5	+0.017832	+0.018727	-0.000896	24.44
6	+0.010945	+0.013206	-0.002260	11.11
7	+0.012987	+0.010382	+0.002605	17.78
8	+0.011125	+0.009317	+0.001807	13.33
9	-0.001771	-0.001636	-0.000135	2.22
10	+0.001151	+0.003942	-0.002790	11.11
11	+0.006104	+0.004920	+0.001184	11.11
12	+0.002028	+0.001802	+0.000226	8.89
13	-0.002464	-0.001352	-0.001112	8.89
14	+0.001243	+0.001216	+0.000028	6.67

Each row averages 45 transitions, with equal protein weight. k is the count before the next response is obtained, not the number bought by an adaptive policy. Negative gains are retained.

Table C5.11. One-response changes by test protein, averaged over all depths.

Test protein	Total gain (pKi)	Correction gain (pKi)	Reselection gain (pKi)	Option changes (%)
CHEMBL1821	+0.017256	+0.016535	+0.000720	22.22
CHEMBL1836	+0.115067	+0.113132	+0.001935	4.44
CHEMBL1867	+0.045847	+0.051722	-0.005874	17.78
CHEMBL1875	+0.013355	+0.012980	+0.000375	20.00
CHEMBL1898	-0.001692	+0.000139	-0.001831	28.89
CHEMBL1945	+0.063764	+0.057688	+0.006077	13.33
CHEMBL2049	+0.029868	+0.034863	-0.004995	37.78
CHEMBL229	+0.000173	-0.001037	+0.001211	31.11
CHEMBL249	+0.025208	+0.022470	+0.002737	22.22

Test protein	Total gain (pKi)	Correction gain (pKi)	Reselection gain (pKi)	Option changes (%)
CHEMBL3764	+0.017117	+0.018302	-0.001186	15.56
CHEMBL4315	+0.012945	+0.009968	+0.002978	17.78
CHEMBL4561	+0.012177	+0.013957	-0.001779	42.22
CHEMBL4644	+0.032244	+0.033939	-0.001695	4.44
CHEMBL4777	+0.044203	+0.040339	+0.003864	6.67
CHEMBL4792	+0.025735	+0.023309	+0.002426	15.56

Each protein contributes 45 potential transitions: three tasks and 15 depths. Averaging these rows gives Table C5.9. Acquisition costs and the local optional second response are absent from both tables. Small discrepancies between printed sums result only from rounding.

C.6 Detailed Analyses of Acquisition and Stopping

This appendix gives the definitions, controlled comparisons, and detailed results behind Section 4.5. We examine where an acquired response can change other predictions, whether those changes correct the remaining errors, and whether the original stopping score recognizes the value of further measurements. The prediction rules and acquisition policies are those evaluated in Sections 4.2–4.4.

C.6.1 Evaluation scope and correction coefficients

ChEMBL uses the 15 test proteins and 45 tasks from Sections 4.2–4.3. Each task has 15 molecules available for measurement and a separate set of 15 molecules whose responses are to be predicted.

HELP uses five individual source predictors, six development devices, and four test devices. Each test device contributes three tasks, each with 42 architectures available for measurement and a separate set of 42 architectures whose latencies are to be predicted. In both applications, P denotes the acquisition pool and Q denotes the prediction set. The three tasks associated with each protein or device are referred to below as its three streams. We keep the fitted models, development-selected settings, measurement histories, and original decisions fixed throughout these retrospective diagnostics.

Let k denote the number of responses already acquired. After at least one acquisition, candidates remain available at counts k = 1–14 in ChEMBL and k = 1–41 in HELP. We compare both applications over the common range k = 1–14, and additionally report HELP results over its full range k = 1–41. The initial state, k = 0, is reported separately.

The complete posterior-geometry path records the sequence of histories obtained by continuing acquisition until the pool is exhausted. A history contains the identities and responses of the candidates acquired so far. For a given cost, the adaptive policy may stop before this path ends. We distinguish all analyzed histories from those the policy actually reaches, including its stopping point. We also report separately the histories at which the policy stops while unobserved candidates remain.

ChEMBL evaluates all 15 prediction cases in each task, whereas HELP evaluates all 42. To compare diagnostics using the same number of prediction cases, we also use a fixed subset of 15 HELP architectures. This subset is selected by a deterministic ordering of architecture identifiers, without using latency responses. It changes only which predictions are summarized or scored; all 42 acquisition candidates, original histories, and decisions remain unchanged.

For comparisons using separate prediction cases, we divide the relevant prediction set into two disjoint subsets, A and B, as evenly as possible. These are the two halves of Q used below. We can calculate a correction descriptor on A and evaluate the corresponding outcome on B, then reverse the roles, while retaining the original decisions made for the complete prediction set.

Let a be the prediction option selected at history h and hold it fixed for the next correction. Write e for the signed Q error $\eta_a(h)$ in Appendix A.2.4, and let z_j be candidate j's response minus its current corrected prediction. With $A_j = A_{h,j}$, the exact update is

$$\Delta_{a,j}(h) = A_j z_j, \qquad g_j = \tfrac{1}{m} \textstyle\sum_{i=1}^{m} \left(|e_i| - |e_i - A_{ij} z_j| \right). \quad \text{(C6.1)}$$

Here m is the number of cases in the evaluation view. A depends on the correction kernel, regularization, and acquired identities. It is not the posterior-geometry score or the outcome-model covariance. Its columns can be calculated before the remaining responses are revealed. Computing the realized gains also uses hidden candidate responses and Q responses, only after the original decisions have been fixed. Reselection of the prediction option is excluded from this single-correction calculation and retained separately in final-policy comparisons.

For a nonzero column, define $p_{ij} = A_{ij}^2 / \parallel A_j \parallel_2^2$. Its normalized breadth is $1/\left(m \sum_i p_{ij}^2\right)$; it is larger when squared influence is spread over more Q cases. Signed overlap is the mean cosine between distinct nonzero columns. We measure variation in correction strength by the coefficient of variation: the population standard deviation of the coefficient-vector norms divided by their mean. Negative coefficient energy is the mean fraction of each column's squared norm carried by negative entries. Zero columns are omitted where normalization is undefined, and overlap is missing when fewer than two nonzero columns remain.

Table C6.1. Correction-map descriptors at common measurement counts.

Descriptor	ChEMBL All 15 prediction cases	HELP All 42 prediction cases	HELP 15-case prediction subset
Normalized breadth	0.316356 [+0.311172, +0.322716]	0.445490 [+0.435785, +0.449671]	0.534239 [+0.524779, +0.537928]
Signed direction overlap	0.474119 [+0.457370, +0.482444]	0.743176 [+0.727911, +0.749268]	0.773281 [+0.759493, +0.778526]
Variation in correction strength	0.238886 [+0.224036, +0.245763]	0.125639 [+0.123464, +0.127368]	0.205691 [+0.202316, +0.213180]
Negative coefficient energy	0.095797 [+0.082723, +0.101206]	0.000895 [+0.000824, +0.000997]	0.001922 [+0.001656, +0.002421]

We first average each descriptor over the included measurement counts within a task, then average tasks within each protein or device, and give proteins or devices equal weight. For each deletion analysis, we repeat the aggregate after omitting one entire protein or device at a time. Brackets show the smallest and largest resulting means; they are not confidence intervals. ChEMBL overlap uses the 13 defined counts k = 1–13; its other descriptors include k = 14. The HELP subset contains the fixed 15 prediction architectures described above.

The conditional correction effects are more differentiated in ChEMBL and broader and more overlapping in HELP. This contrast is not present in the same form at the initial state: HELP already has breadth 0.961645 and signed overlap 0.993227 at k = 0, and its initial map is identical across devices within each stream. Matching the number of prediction cases or the measurement count does not match the remaining pool size, representation, kernel, or measurement history.

C.6.2 Relation to acquisition comparisons

We ask whether greater variation in correction strength accompanies a larger acquisition advantage. We calculate the variation on one half of the prediction set and score the advantage on the other half, then reverse the roles. The advantage is random-order MAE minus posterior-geometry MAE at the same measurement count, divided by the initial MAE for that task and prediction subset. The random comparator averages all 32 retained orders. Both methods use the same prediction-option selection and correction rule; scoring separate prediction subsets does not change the original decisions made for the complete prediction set.

Across-protein comparisons without MAE matching first average the three tasks within each protein at each exact count, calculate a Spearman correlation across proteins, and average the defined correlations across counts. The other comparisons in Table C6.2 use gain-difference-weighted ordering. This asks whether the task or protein with greater variation in correction strength also has the larger

acquisition advantage on the opposite prediction subset, giving more weight to pairs with a larger advantage difference.

For each pair, multiply the sign of its correction-strength-variation difference by its acquisition-advantage difference. Sum these signed contributions and divide by the sum of the absolute advantage differences. Agreement contributes positively, disagreement negatively, and an exact descriptor tie contributes zero to the numerator. A zero denominator remains undefined. This is the same pairwise form used in Equation (C6.7), with correction-strength variation replacing the stopping score and acquisition advantage replacing realized continuation gain. It is neither a correlation coefficient nor a classification accuracy.

Within a protein or device, compare its three tasks at the same measurement count, pool the pairwise numerators and denominators across included counts, form one ratio per protein or device, and then average the defined ratios equally. For the MAE-matched across-protein comparison, first average the three tasks within each protein at each count, retain protein pairs meeting the MAE-matching rule, compute one ordering ratio per count, and average the defined ratios across counts. MAE matching requires both initial and current MAE on the outcome subset to differ by no more than a factor of 1.25. It restricts the retrospective comparison pairs; it does not change the acquisition policy.

Table C6.2. Variation in correction strength and the acquisition advantage on the opposite prediction subset.

Comparison, k = 1–14	**Structure on A; outcome on B**	**Structure on B; outcome on A**
ChEMBL: across proteins (Spearman)	+0.239286	+0.209341
	[+0.160440, +0.357566]	[+0.143533, +0.323753]
ChEMBL: within protein (ordering)	+0.095633	+0.154601
	[+0.051630, +0.142840]	[+0.116503, +0.205419]
ChEMBL: within protein, MAE-matched	+0.181861	+0.383432
(ordering)	[+0.113683, +0.250716]	[+0.336003, +0.463490]
ChEMBL: across proteins, MAE-matched	+0.161458	+0.228951
(ordering)	[-0.095788, +0.313508]	[+0.044638, +0.397179]
HELP: all 42 prediction cases	-0.365942	-0.127915
Within device (ordering)	[-0.468211, -0.306428]	[-0.208499, -0.060886]
HELP: 15-case prediction subset	-0.300589	-0.117502
Within device (ordering)	[-0.448046, -0.218835]	[-0.232895, +0.009069]

Each entry gives the recorded statistic and its whole-unit deletion range. ChEMBL's Spearman averages have 13 defined depths. Unrestricted within-protein comparisons use 15 proteins and 630 same-count stream pairs; HELP uses four devices and 168 pairs. MAE-matched pairs have both initial and current MAE within a factor of 1.25.

For ChEMBL, the within-protein MAE-matched contrasts include 13 proteins and 140 stream pairs for A $\rightarrow$ B, and 14 proteins and 184 pairs for B $\rightarrow$ A. The across-protein matched comparison is less stable: 86 pairs remain for A $\rightarrow$ B and 122 for B $\rightarrow$ A, and the A $\rightarrow$ B deletion range includes zero. For HELP, the within-device associations are negative when calculated using all 42 prediction cases. With the 15-case prediction subset, the B $\rightarrow$ A deletion range includes zero. Thus the ChEMBL association is not a common or reversed selection rule for both applications.

The broad HELP maps also do not imply uniformly similar realized gains. Here a candidate gain is the decrease in MAE after adding that candidate's response and applying the complete selection-and-correction rule F, including any source-predictor reselection. This comparison ends after one measurement and includes neither measurement cost nor an optional second measurement. At k=0, the standard deviation of these candidate gains divided by initial MAE is 0.327605 in HELP and 0.232311 in ChEMBL, whereas division by mean absolute candidate gain gives 0.703088 and 0.914355. These normalizations answer different questions. The adverse first-measurement comparison in Section 4.4 remains part of the acquisition evidence.

C.6.3 WHICH RESIDUAL IS PAIRED WITH WHICH CORRECTION?

At each fixed history, retain A, the current Q error e, and the multiset of residuals of all remaining candidates. Let n_h be their number. Form the candidate-by-residual gain table

$$g_{j\updownarrow} = \frac{1}{m}\sum_{i=1}^{m}\left(|e_i| - \left|e_i - A_{ij}z_{\updownarrow}\right|\right). \text{ (C6.2)}$$

The diagonal uses each candidate's own residual. A permutation π assigns the same residual values one-to-one to the fixed columns. The original mean and the exact reassignment mean are

$$\mu_{orig} = \frac{1}{n_h}\sum_j g_{jj}, \qquad \mathbb{E}_\pi\mu_\pi = \frac{1}{n_h^2}\sum_{j,\updownarrow} g_{j\updownarrow}. \text{ (C6.3)}$$

No new responses are sampled and no model or policy is fitted. This is a local comparison within the correction calculation, not a physically reconstructed benchmark. Gains are normalized by the same task/panel's initial MAE before averaging; variances are normalized by its square. It is not a ratio of application-level averages.

The absolute-error identity separates useful signed movement from overshoot. Set $s_i = 1$ when $e_i \geq 0$ and $s_i = -1$ otherwise, and define

$$w_j = \frac{1}{m}\sum_i s_i A_{ij}, \qquad O_{j\updownarrow} = \frac{1}{m}\sum_i \max\{s_i A_{ij} z_{\updownarrow} - |e_i|, 0\}. \text{ (C6.4)}$$

$$g_{j\updownarrow} = w_j z_{\updownarrow} - 2O_{j\updownarrow}. \text{ (C6.5)}$$

Consequently, the original-pairing mean advantage is the population covariance of w_j with z_j, minus twice the difference between diagonal and grand-mean overshoot. The first term measures whether a residual is attached to a direction that helps the signed errors still present. The second retains the cost of moving past the correct value.

Table C6.3. Original residual pairing, reassignment, and the exact mean-gain decomposition.

Quantity	ChEMBL All 15 prediction cases	HELP All 42 prediction cases	HELP 15-case prediction subset
Mean gain: original pairing	+0.011508	+0.030536	+0.031114
Mean gain: reassignment mean	+0.001858	+0.027815	+0.028116
Original minus reassignment	+0.009650	+0.002721	+0.002998
Signed-alignment contribution	+0.008285	+0.002231	+0.002644
Reduced-overshoot contribution	+0.001365	+0.000490	+0.000354
Candidate variance: original	+0.002400	+0.004350	+0.005274
Candidate variance: reassignment	+0.003567	+0.006816	+0.008697
Candidate variance difference	-0.001167	-0.002466	-0.003423

Current prediction option held fixed; k=1–14. Gain rows are divided by initial MAE and variance rows by initial MAE squared, within task/panel before aggregation. Candidate variance means the population variance of candidate gains within a history.

The original pairing improves mean correction gain in both applications. Signed alignment is the larger contribution in these common-count averages. In raw loss units, the mean advantage is 0.007507 pKi in ChEMBL and 0.001767 log-MAE in HELP; these are not directly comparable loss scales. Positive mean pairing advantages persist in prediction subsets A and B (Table C6.4). HELP's reduced-overshoot term is positive for both the complete 42-case set and the complete 15-case subset, but its device-deletion range includes zero on half B of the 42-case set; positivity of that component is not uniform across the evaluated subsets.

Table C6.4. Residual-pairing sensitivity to the Q evaluation view.

Application / evaluation view	Mean-gain advantage	Whole-unit deletion range	Variance change
ChEMBL: 15 prediction cases Complete set	+0.009650	[+0.008483, +0.010300]	-0.001167

Application / evaluation view	Mean-gain advantage	Whole-unit deletion range	Variance change
ChEMBL: 15 prediction cases Subset A	+0.010816	[+0.009516, +0.011526]	-0.001100
ChEMBL: 15 prediction cases Subset B	+0.008417	[+0.007197, +0.009047]	-0.001661
HELP: 42 prediction cases Complete set	+0.002721	[+0.000917, +0.003525]	-0.002466
HELP: 42 prediction cases Subset A	+0.002559	[+0.000564, +0.003561]	-0.004006
HELP: 42 prediction cases Subset B	+0.002946	[+0.001299, +0.003540]	-0.001480
HELP: 15-case prediction subset Complete subset	+0.002998	[+0.001078, +0.003776]	-0.003423
HELP: 15-case prediction subset Half A	+0.002343	[+0.000143, +0.003692]	-0.004035
HELP: 15-case prediction subset Half B	+0.003760	[+0.002168, +0.004518]	-0.002753

Current option; common k=1–14. "Mean-gain advantage" is original minus exact reassignment mean, normalized by the view-specific initial MAE. "Variance change" is the corresponding candidate-variance difference, normalized by initial MAE squared. Displayed intervals are deletion sensitivities of the mean-gain advantage. All these mean-advantage ranges are positive; the recorded variance-change deletion ranges are negative.

C.6.4 Sensitivity of the mean to residual reassignment

Candidate-gain variance in Table C6.3 is different from the variability of the candidate mean across residual assignments. To quantify the latter, double-center the gain table: subtract its row and column means and add its grand mean, giving $\widetilde{g}_{j\updownarrow}$. For $n_h > 1$, the exact variance is

$$V_h = Var_\pi\left(\mu_\pi\right) = \frac{\sum_{j,\updownarrow} \widetilde{g}_{j\updownarrow}^2}{n_h^2(n_h-1)}. \quad \text{(C6.6)}$$

If every coefficient column is identical in both direction and magnitude, reassignment only reorders the candidate gains: their mean and distribution do not change. High cosine overlap alone does not imply this property when column magnitudes differ. Identical residuals also make the reassignment sensitivity vanish. Thus this is a property of A, e, and z together, not a response-free measure of information or uncertainty. For a single remaining candidate there is only one assignment and its variance is zero.

Table C6.5. Initial-state, fixed-initial-option, and longer-history controls.

Application / source / depths	Mean-gain advantage	Whole-unit deletion range	Var. of reassignment mean
ChEMBL / current / k=0	+0.020805	[+0.016790, +0.022403]	0.000157444
ChEMBL / current / k=1–14	+0.009650	[+0.008483, +0.010300]	9.31659e-05
ChEMBL / initial option / k=1–14	+0.009873	[+0.008406, +0.010554]	9.52864e-05
HELP: all 42 prediction cases Current option; k=0	+0.009146	[+0.001235, +0.012884]	2.23494e-05
HELP: all 42 prediction cases Current option; k=1–14	+0.002721	[+0.000917, +0.003525]	4.63798e-06
HELP: all 42 prediction cases Initial option; k=1–14	+0.002837	[+0.000527, +0.003862]	3.89712e-06
HELP: all 42 prediction cases Current option; k=1–41	+0.001181	[+0.000471, +0.001472]	2.54469e-06

All rows use the complete prediction set: 15 cases in ChEMBL and 42 in HELP. Mean advantages and deletion ranges use initial-MAE normalization; reassignment-mean variances use its square. "Current" holds the option selected at that history fixed for one correction. "Initial option" holds the option selected at k=0 throughout the diagnostic, without changing the original histories.

The current HELP reassignment-mean variance is smaller in the longer-history average, but this does not establish a monotone decrease along every path. More generally, variation across permutations is distinct from the estimated value used by STOP: the latter must predict the remaining gain before the hidden residuals and Q errors used here are available. The positive mean pairing advantages in the initial-option controls show that a change of prediction option is not required for this local effect.

C.6.5 What the observed state contributes to STOP

We now use the original cost-inclusive score and the same first-plus-optional-second continuation as in Section 3.6 and Appendix C.4. Let $M_{u,k,s}(c)$ be the estimated net gain and $G_{u,k,s}(c)$ its realized net gain for unit u, measurement count k, and stream s at cost c. Candidate selection and the conditional second-acquisition decision are fixed before Q responses score the outcome. This compares different histories, not candidate rankings within a history.

At each device, count, and cost, compare the three streams. For a declared scope $\mathcal{S}_u$ of valid matched pairs, define

$$S_u = \frac{\sum_{(k,c,s,t)\in\mathcal{S}_u} sgn_\tau(M_{u,k,s}-M_{u,k,t})(G_{u,k,s}-G_{u,k,t})}{\sum_{(k,c,s,t)\in\mathcal{S}_u}|G_{u,k,s}-G_{u,k,t}|}. \quad \text{(C6.7)}$$

The reported value is the equal-device mean of these unit ratios, not an average of cellwise ratios. Modeled-score differences of at most 10^{-12} in native units count as diagnostic ties. A zero denominator remains missing. Positive values mean that larger scores tend to accompany larger realized gains, with larger gain differences receiving more weight; the statistic is neither Spearman correlation nor decision accuracy.

The frozen-initial comparator replaces the current score by the same stream and cost's score at k=0 while scoring the current realized gains. It preserves an initial ordering; it is not a new policy with measurements removed. HELP costs and gains are expressed in natural-log loss units, with $c = \lambda_{internal} \times 0.6103144578607641$. ChEMBL's original policy uses $c = \lambda$. These unit conversions do not select new operating points.

Table C6.6. Current and frozen-initial STOP-score ordering in HELP.

HELP: all 42 prediction cases **Histories / score**	**Ordering**	**90% pointwise interval**	**Device-deletion range**
k=1–41: current	+0.604802	[+0.410438, +0.790788]	[+0.501838, +0.681227]
k=1–41: frozen initial	-0.038745	[-0.142194, +0.086722]	[-0.080902, +0.018090]
k=1–41: current - initial	+0.643547	[+0.454664, +0.836098]	[+0.513552, +0.762130]
k=1–14: current	+0.581419	[+0.368837, +0.782496]	[+0.477006, +0.693503]
k=1–14: frozen initial	-0.029798	[-0.143596, +0.097571]	[-0.074948, +0.034632]
k=1–14: current - initial	+0.611217	[+0.384316, +0.830193]	[+0.484525, +0.764859]
Histories reached by the adaptive policy: current score	+0.578727	[+0.365203, +0.792250]	[+0.470547, +0.700663]
k=1–10: current	+0.594746	[+0.391729, +0.802461]	[+0.493691, +0.704520]
k=11–31: current	+0.458833	[+0.232428, +0.717867]	[+0.290073, +0.557298]
k=32–41: current	+0.514136	[+0.150330, +0.882719]	[+0.388566, +0.716480]
Histories where the adaptive policy stops: current score	+0.253189	[-0.037700, +0.544078]	[+0.076838, +0.408670]

All 42 HELP prediction cases; four test devices. The 90% pointwise percentile intervals are obtained from all 256 ordered bootstrap resamples of the four devices, keeping all tasks from each resampled device together. They condition on the prepared fits, development selections, and current test devices; they are not simultaneous bands. Device-deletion ranges are a separate sensitivity summary. All included histories have at least one unobserved candidate. For the reached-history and stopping-history rows, both streams of a matched pair must satisfy the stated restriction.

The updated-minus-initial difference is positive for each device (Table C6.7). The observed-depth average therefore is not driven by a favorable sign on only one device. The depth-range rows in Table C6.6 do not constitute a test that ordering remains unchanged with depth. In particular, ordering at histories where the adaptive policy stops has a pointwise interval crossing zero, despite the favorable all-observed average.

Table C6.7. Per-device current versus frozen-initial ordering, k=1–41.

Test device	Current score	Frozen initial	Difference
gold_6226	+0.375525	+0.087725	+0.287800
pixel2	+0.422075	-0.209253	+0.631328
raspi4	+0.913692	-0.119841	+1.033533
titan_rtx_1	+0.707915	+0.086387	+0.621528

All 42 HELP prediction cases and all 26 original costs. Each row is the unit ratio in Equation (C6.7); the application mean weights these four rows equally. Differences of ordering statistics can exceed one even though each individual ordering statistic lies between -1 and +1.

As a separate check, use the realized gain on one Q half as the ordering score for gains on the other half, retaining the original full-Q policy and the same conditional continuation. The sign is taken from the unrounded gain difference, without the modeled-score tie tolerance. The two directions use different gain-difference denominators.

Table C6.8. Reciprocal Q-half correspondence of realized continuation gains in HELP.

HELP evaluation view	A → B	B → A
All 42 prediction cases	0.734236	0.758393
	[+0.664013, +0.922279]	[+0.688077, +0.944893]
15-case prediction subset	0.682727	0.663277
	[+0.613093, +0.830644]	[+0.570761, +0.839087]

k=1–41. A → B means that gain on A supplies the ordering and gain on B supplies the signed and absolute differences; B → A reverses those roles. Entries include whole-device deletion ranges. These are retrospective outcome comparisons: the acquisition policy does not see Q responses. Comparing full Q with itself is undefined as a coherence check, rather than favorable evidence.

This correspondence and the current-score comparison address different questions. The first checks whether the differences in realized gains extend across distinct scored cases. The second checks whether the original value calculation uses the observed state to distinguish those differences. Neither replaces the actual cost-boundary diagnostics in Appendix C.4, which compare each estimated and realized gain with zero.

C.6.6 Useful ChEMBL corrections after adaptive stopping

For a fixed task, let h_k denote the retained history after k measurements, and write $F_k = F(h_k)$. Thus $F_0 = F(h_0)$ is the output before any measurement. Let L_k be the observed MAE of F_k on the prediction subset being evaluated. The initial signed error used below is calculated retrospectively from withheld responses; it is distinct from the prepared absolute-error estimate used to initialize source-predictor selection.

The ChEMBL control changes the assignment of initial Q errors, not the acquired responses or the predictions. For one fixed task and evaluation panel, let $e_0 = y_Q - F_0(Q)$. Reassign these error values among Q cases and score $y_Q^\pi = F_0(Q) + \pi(e_0)$, using one common permutation across all depths of that task. Initial MAE and the signed error distribution are preserved. All P responses, selected options, scores, actions, optional-second decisions, endpoint counts, and costs remain fixed.

Writing $d_{k,i} = F_k(i) - F_0(i)$, the exact mean loss over every reassignment is

$$\overline{L}_k = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |e_{0,j} - d_{k,i}|, \qquad E_k = \overline{L}_k - L_k. \text{ (C6.8)}$$

A uniform correction across Q survives this control. What changes is how the fixed case-specific update is paired with the actual errors. The control is separate for each Q panel and task; it is not one globally relabeled response database across overlapping streams. No policy is rerun on the reassigned responses.

For the STOP-score comparison, keep the original score differences and the original absolute gain-difference denominator in both columns of Table C6.9. Only the gain contrast in the numerator

changes. This makes their difference a linear accounting of the original error assignment, not the expected rank correlation obtained by ranking permutation-mean gains. Let $B = J_{\text{fixed}} - J_{adaptive}$ denote endpoint benefit. At the original stopping counts,

$$B_{orig} - \mathbb{E}_\pi B_\pi = E_{n_{adaptive}} - E_{n_{\text{fixed}}}. \quad \text{(C6.9)}$$

Table C6.9. ChEMBL initial-error reassignment: score ordering and final-policy benefit.

ChEMBL quantity / panel	Original	Reassignment	Difference	Deletion range of difference
Ordering: all 15 prediction cases	+0.007028	+0.107425	-0.100397	[-0.129811, -0.059283]
Ordering: prediction subset A	+0.109481	+0.116177	-0.006695	[-0.033984, +0.029781]
Ordering: prediction subset B	-0.131489	+0.027285	-0.158774	[-0.185738, -0.112199]
Endpoint benefit, all costs	-0.004368	+0.007697	-0.012065	[-0.012987, -0.010770]

Ordering rows use common k=1–14 and all original costs. They are dimensionless. The endpoint row is an equal-grid summary over all 34 costs in pKi MAE units, retaining the original adaptive control and development-selected fixed count. Positive endpoint benefit favors adaptive stopping. Brackets are whole-protein deletion ranges of original minus reassignment, not confidence intervals.

On full Q, the original case assignment contributes negatively to the score-order functional and to the relative benefit of adaptive stopping over fixed count under the fixed scoring control. The A-half ordering contribution is less stable and its deletion range includes zero. At the endpoint, the assignment contribution favors fixed count for 13 of the 15 proteins. Across costs its point estimate is negative at 26, positive at four, and tied at four; at 24 costs it remains negative under every whole-protein deletion. These costs are repeated evaluations, not independent replications.

Table C6.10. Where the ChEMBL endpoint assignment contribution arises.

ChEMBL endpoint contribution	Mean (pKi)	Whole-protein deletion range
Adaptive stops before fixed count: total	-0.012701	[-0.013516, -0.011619]
Same-option correction part	-0.011940	[-0.012761, -0.010647]
Option-reselection part	-0.000761	[-0.000971, -0.000553]
Original local continuation part	-0.009596	[-0.010336, -0.007870]
Remainder to fixed endpoint	-0.003105	[-0.003748, -0.002569]
Adaptive acquires beyond fixed count: total	+0.000636	[+0.000166, +0.000930]
Equal acquisition counts	+0.000000	[+0.000000, +0.000000]

Each row contributes to the same mean over the full evaluation population. Cases outside the row's group contribute zero; the row is not a conditional mean within that group. All values are original minus exact reassignment, in pKi units. The correction and reselection rows partition the contribution from cases in which adaptive stopping ends before the fixed-count policy. Independently, the local-continuation and remainder rows partition that same contribution, so those two decompositions must not be added together.

When adaptive stopping ends before the fixed-count policy, its contribution to the overall difference is -0.012701; the same-option correction part is -0.011940. Within that contribution, -0.009596 is already associated with the original local continuation. The remainder also includes reconciliation when a conditional second measurement passes the comparator endpoint, so it is not entirely a more distant horizon. This connects the positive gains remaining after ChEMBL stops in Appendix C.4 with corrections that the later fixed-count predictions actually obtain, without summing overlapping local gains into a policy effect.

C.6.7 FINAL-POLICY ACCOUNTING AND LIMITS OF STRUCTURAL SUMMARIES

For HELP, we decompose the observed difference between the adaptive and fixed-count policies into contributions from prediction correction, option reselection, and acquisition cost. We sum prediction-loss changes along the observed path between their stopping points. At each acquisition, we first update the option selected before the response is revealed and then apply any option reselection. Adding the difference in acquisition costs gives the actual difference between the two final-policy outcomes.

Table C6.11. HELP endpoint accounting at c ≈ 0.00315.

HELP endpoint, c ≈ 0.00315	**Value**
Fixed acquisition count	5.000000
Adaptive mean acquisition count	4.583333
Fixed log-MAE	0.208789
Adaptive log-MAE	0.154036
Same-option correction contribution	+0.007508
Option-reselection contribution	+0.047244
Acquisition-cost contribution	+0.001314
Total: J_fixed - J_adaptive	+0.056066

Original-control adaptive versus the development-selected fixed count, evaluated on all 42 HELP prediction cases. Contributions and losses are in log-MAE units; positive contributions favor adaptive stopping. The 90% pointwise interval for the total benefit is [0.000417, 0.125250]. Counts, losses, and the additive total agree with Appendix C.4. This is path accounting, not an ablation that removes option reselection.

The reselection term is the largest at this setting. The benefit therefore cannot be attributed to the correction coefficients alone. The local optional-two gains, the same-option residual-pairing comparison, and this endpoint accounting use different endpoints and must remain separate.

We also retain the less conclusive structural comparisons. At each measurement count, the HELP analysis relates signed overlap to the original value-ordering statistic across the four devices and averages the defined Spearman correlations. A second check averages a descriptor over two streams and relates it to the third stream's final benefit, rotating the held stream. This is a descriptive association, not a fitted selection rule.

Table C6.12. Signed overlap does not by itself supply an advance endpoint selector.

Association with signed overlap	**Value**	**Aggregation / scope**
All 42 prediction cases	+0.364012	Mean exact-depth Spearman; 38 depths
Descriptor on B; ordering on A	+0.353770	Mean exact-depth Spearman; 38 depths
Descriptor on A; ordering on B	+0.336842	Mean exact-depth Spearman; 38 depths
Held-stream endpoint benefit	-0.200000	Mean of three held-stream rotations
Held-stream benefit / initial MAE	-0.266667	Mean of three held-stream rotations
Initial overlap → held-stream benefit	Undefined	No between-device variation in initial overlap

HELP with five source predictors, evaluated on all 42 prediction cases. The first three rows average defined exact-depth Spearman correlations over k=1–41. The two endpoint rows use observed-depth descriptors and the preserved all-grid endpoint summaries, then average the three held-stream rotations. Undefined initial association reflects identical initial overlap across devices; it is not set to zero.

The positive relation to score ordering does not extend to a positive, scale-stable held-stream endpoint relation in these results. The analyses therefore separate useful correction effects, their recognition by the value calculation, and the final cost-inclusive outcome.

D COMPUTATIONAL ENVIRONMENT

The experiments used 64-bit Ubuntu 24.04 long-term support (LTS). ChEMBL and our HELP procedure use CPU computation. The HELP baseline was trained on a single NVIDIA L4 graphics processing unit (GPU; 24 GB), using the Compute Unified Device Architecture (CUDA) 12.8 build of PyTorch 2.9.1 with the CUDA Deep Neural Network library (cuDNN) 9.10.2 and NVIDIA driver 595.71.05. The reported prediction comparisons and same-host timings use CPU execution; CUDA was not used in these timed calls. Table D.1 gives the numerical software versions.

Table D.1. Numerical software used for the reported computations.

Component	**Our procedure (ChEMBL and HELP)**	**HELP predictor (timed process)**
Python	3.12.3	3.11.16
NumPy	2.2.6	1.26.4
pandas	2.3.2	2.3.3
SciPy	1.15.3	1.16.3
scikit-learn	1.7.1	1.7.2
joblib	1.6.0 (HELP)	1.5.2
RDKit	2025.03.5 (ChEMBL)	Not used in the timed call
PyTorch	Not used in the timed call	2.9.1+cu128; CPU execution

The same-host comparison used an Intel Xeon 6975P-C with 32 virtual CPUs. Each method ran with one worker and one thread per numerical library; PyTorch intra-op and inter-op thread counts were both one. The two methods used the same CPU host and thread limits while retaining their respective software environments. Timing scope and aggregation are specified in Appendix B.5.