WHAT PREDICTIVE-IMPROVEMENT REWARDS MAKE AGENTS COLLECT

Anonymous authors
Paper under double-blind review

ABSTRACT

Predictive-improvement rewards value an observation by how much it helps predict later measurements. An agent that chooses those measurements can spend its budget making earlier discoveries earn credit. We study what this incentive makes it collect. With enough independent noiseless bits and an even budget, every policy maximizing the complete observation-deletion reward queries each chosen bit twice and acquires half the attainable information. An exact frontier characterizes the tradeoff, and a matching formulation shows how it depends on scoring times. The preference can also redirect learned behavior: switching an effective Gaussian-field policy to deletion reward increases credit, reduces coverage and leaves 45% more reconstruction error than continuing task training. Fitting masked predictions to policy records adds another dependence. Retained actions reveal deleted outcomes, changing the reward even with exact predictions; refitting can cycle or sustain a self-consistent optimum with lower information. Finally, interventions on temperature records show how unavailable inputs and changing evaluation targets alter the task value of selected measurements. These results connect the choices used to assign predictive credit to the information a finite acquisition budget buys.

1 INTRODUCTION

Consider an agent that can make four measurements. Querying four independent unknown bits reveals four bits of its environment. Querying two bits twice reveals two. A reward based on improvement in future prediction can prefer the second allocation: the repeated measurement makes the predictive value of the first measurement observable. With an exact predictor, the four distinct measurements earn zero observation-deletion reward, whereas the two repeated pairs earn two bits of reward. The same budget supports two different objectives: acquiring new information and acquiring information that explains subsequent measurements.

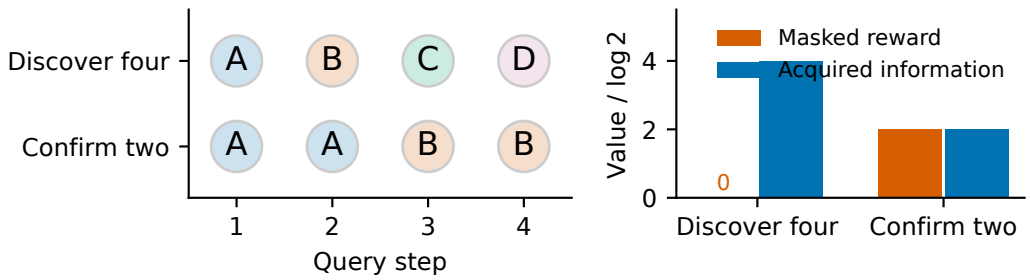


Figure 1: Four measurements can discover four bits or confirm two. The complete observation-deletion reward credits the first measurement when the same bit is observed again. Bars compare reward and acquired information, normalized by log 2. Proposition 1 optimizes over all adaptive policies.

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.

Prediction-based intrinsic rewards use a world model to value observations. NDIGO measures their effect on prediction at a fixed future horizon (Azar et al., 2019). In-context curiosity sums improvements over later predictions and relates the score to information gain through a finite-horizon residual in Bayesian experimental design (Elmoznino et al., 2026). To understand exploration under a limited budget, we also need to know which policy maximizes that score. A small residual under one policy can still change the ordering of policies. The example shows what is at stake: queries spent on confirmation leave other unknowns unresolved.

We characterize this competition between discovery and confirmation for observation-deletion rewards. With enough independent noiseless bits and an even budget, every optimal adaptive policy pairs its queries and acquires half the attainable information. An exact frontier shows how much information can be recovered by accepting less reward. The cost persists under any strictly positive time weights; a matching formulation also determines the optimum when only selected prediction intervals earn credit. Noise and shared structure change this allocation because confirmation can then reveal additional information.

We test whether increasing the reward moves a learned policy in the same direction by changing the objective of a policy that already reconstructs a Gaussian field effectively. Continuing under deletion reward increases predictive credit, reduces coverage and leaves 45% more reconstruction error than continuing task training. A separate experiment learns and refits the predictor: likelihood improves, yet the acquisition gap remains.

Fitting predictions directly to policy records introduces another way for the policy to affect its credit. An action chosen in response to an observation can reveal that observation even after it is deleted from the predictor's input. We show how this evidence changes reward preferences. Exact refitting can cycle, and a self-consistent reward optimum can collect less information than another policy. In Mastermind, optimizing the fitted source over a continuous policy family favors nearly uniform queries, with 60.5% identification accuracy versus 81.5% for posterior sampling.

The task determines what these acquisition choices are worth. On temperature records, unavailable reports change both the inputs a requested set provides and the targets used to evaluate it. Matched comparisons and interventions on the selector's availability assumptions explain a reversal in reconstruction ranking. These interventions complete the link from how predictive credit is assigned to what the selected measurements provide.

2 Reward and acquisition

An episode draws a latent environment $\theta \sim p(\theta)$. At step k, a nonanticipating policy selects A_k from $h_{k-1} = (A_1, Y_1, \ldots, A_{k-1}, Y_{k-1})$ and observes $Y_k \sim p_\theta(\cdot \mid A_k)$. Given θ and the query, the observation law is independent of the preceding history. The episode has T queries, and acquired information is $U(\pi) = I_\pi(\theta; h_T)$.

Let $H_k = (h_{k-1}, A_k)$. The record S_{jk} deletes Y_j from H_k and retains every action and other outcome, together with j and k. For full and masked predictors F and M, the complete future-sum reward is

$$G(F, M) = \sum_{1 \le j < k \le T} \log \frac{F(Y_k \mid H_k)}{M(Y_k \mid S_{jk})}. \tag{1}$$

Each observation earns credit through every later prediction. Section 3 also studies fixed nonnegative weights on these terms. All information and log rewards use natural logarithms.

The exact full predictor $f(y \mid H_k)$ uses the prior and observed likelihoods. Policy probabilities cancel from its posterior because the full history is given. Deleting an outcome requires specifying how to interpret the actions that remain.

The **interventional masked predictor** m_{do} uses retained actions to specify what was measured, with posterior

$$p_{\mathrm{do}}(\theta \mid S_{jk}) \;\propto\; p(\theta) \prod_{t<k,\; t\neq j} p_\theta(Y_t \mid A_t). \tag{2}$$

This rule conditions on the remaining observations at their query locations. Write $R(\pi, m) = \mathbb{E}_\pi G(f, m)$. Sections 3–4 use m_{do} or a fitted joint model with this rule. Section 5 also uses what the source policy's actions reveal about the deleted outcome.

A fixed design clarifies what the complete score measures. Let p_a be the joint predictive law for predetermined queries $a = (a_1, \ldots, a_T)$. The chain rule gives $G = \sum_j \log p_a(Y_j \mid Y_{-j}) - \log p_a(Y_{1:T})$, hence

$$\mathbb{E}_{p_a} G = H(Y_{1:T}) - \sum_j H(Y_j \mid Y_{-j}) = I(\theta; Y_{1:T}) - \sum_j I(\theta; Y_j \mid Y_{-j}). \tag{3}$$

The first expression is the established dual total correlation (Han, 1978; Austin, 2020); the second uses conditional independence given θ. It rewards dependence among the collected observations and omits information that disappears when an individual observation is removed. With adaptive queries, the realized actions and outcomes are dependent. We evaluate the actual pathwise score in (1), including its observation-dependent terms. Appendix I derives the identity and its Gaussian computation.

3 How scoring determines acquisition

3.1 Discovery and confirmation compete for queries

Let $\theta_1, \ldots, \theta_d$ be independent fair bits, each revealed without noise when queried. Let N_i count queries to coordinate i.

Proposition 1 (optimal allocation). If $d \geq T$ and T is even, every adaptive policy maximizing $R(\pi, m_{\mathrm{do}})$ queries $T/2$ distinct coordinates exactly twice, almost surely. Its reward and information are both $(T/2)\log 2$, while the information optimum is $T \log 2$.

Deleting a coordinate's first observation changes the prediction at its second query. At every later query, another copy remains in the record. Thus, on every path,

$$G(f, m_{\mathrm{do}}) = \log 2 \sum_{i=1}^{d} \mathbf{1}\{N_i \geq 2\}. \tag{4}$$

Each rewarded coordinate costs at least two queries. Attaining the bound forces pairs, including on adaptive paths. This also determines the best information achievable while maintaining a reward threshold. For finite d, put $c^* = \min(d, \lfloor T/2 \rfloor)$. Then

$$\max_{\pi:\; R(\pi, m_{\mathrm{do}}) \geq r} U(\pi) = \min(d \log 2, T \log 2 - r), \qquad 0 \leq r \leq c^* \log 2. \tag{5}$$

Near-optimal reward therefore permits only a small gain in information. Once $T \geq 2d$, every reward optimizer learns all d bits. Appendix B proves both statements and handles odd budgets.

3.2 Time weights and prediction intervals

The need for confirmation may depend on when an observation earns credit. To expose that dependence, weight term (j, k) in (1) by a fixed $w_{jk} \geq 0$, giving G_w.

Proposition 2 (weighted allocation). In the same noiseless model with $d \geq T$, let $t_1(i), t_2(i)$ be the first two query times of coordinate i. On every path,

$$G_w = \log 2 \sum_{i:N_i \geq 2} w_{t_1(i),t_2(i)}, \qquad \max_{\pi} \mathbb{E}_\pi G_w = \log 2 \max_{\mathcal{M}} \sum_{(j,k)\in\mathcal{M}} w_{jk}, \tag{6}$$

where $\mathcal{M}$ ranges over matchings of the T query times.

Credited pairs cannot share a query time. Conversely, any matching is realized by assigning one coordinate to each edge and a fresh coordinate to each unmatched time. A predetermined schedule therefore attains the adaptive optimum. If T is even and every weight is strictly positive, a maximum matching is perfect. Every reward optimizer still learns exactly $T/2$ bits. Positive discounting and normalization by the number of future scores preserve this conclusion.

For a fixed interval H, take $w_{jk} = \mathbf{1}\{k = j + H\}$. The time graph consists of chains, with lengths n_r for residues modulo H. Put $m_H = \sum_r \lfloor n_r/2 \rfloor$. The optimal reward is $m_H \log 2$, and the greatest information among reward optimizers is $(T - m_H) \log 2$. NDIGO uses prefixes before and after an observation rather than deleting it from the full intervening history (Azar et al., 2019). With future actions supplied as interventional query inputs, its fixed-interval oracle score has the same matching optimum in this bit model. Appendix I proves this statement with the appropriate conditioning sets and gives a Brier-score counterpart.

An independently chosen scoring measurement changes the allocation more directly. A uniform terminal deletion probe credits singly observed bits and favors T distinct acquisitions when $d \geq T$. With eight acquisitions, policy learning reaches approximately four bits under the original reward and eight under the integrated terminal-probe reward across three seeds (Appendix B). The extra measurement required by a physical probe is charged explicitly in the GP comparisons of Appendix J.

3.3 Noise and shared structure change the allocation

In the noiseless model, a second report confirms a bit without revealing anything new. With reports independently flipped at probability q, repetition can also resolve uncertainty about that bit. If I_n denotes the information from n reports of one fair bit, its expected reward under a fixed allocation is

$$r_n = nI_{n-1} - (n-1)I_n. \tag{7}$$

Adaptive policies can additionally use agreements and disagreements to decide where to measure next. We optimize them by a Bellman recursion over per-coordinate report counts (Appendix B). At $q = .1$ and $T = 20$, the fixed-count optimum acquires 3.195 nats; the adaptive optimum acquires 5.065 of the attainable 7.361 nats. Feedback raises the fraction from 43.4% to 68.8%.

To vary shared structure while keeping every marginal bit fair, use

$$p_\rho(\theta) = (1-\rho)\,\mathrm{Unif}(\{0,1\}^4) + \frac{\rho}{2}(\delta_{0000} + \delta_{1111}). \tag{8}$$

We optimize reward and information separately over full-history policies at $T = 4$. Without noise, the reward-optimal information fraction rises from .500 at $\rho = 0$ to .788 at .75 and 1 at 1. With $q = .1$, it rises from .699 to .831, then falls to .785 at $\rho = .9$ (Figure 2). Shared structure lets different coordinates confirm common information, but its interaction with noise need not improve the fraction monotonically.

To connect these allocations to a prediction task, we optimize measurement designs for a Gaussian field on $[0, 1]$, with covariance $K(x, x') = \exp[-(x - x')^2/(2\ell^2)]$. A design makes eight measurements chosen from eight candidate locations, allowing repeats and Gaussian noise of variance σ^2. We compare deletion reward, information and reconstruction on 201 prediction locations (Appendix B.8). Across nine length/noise settings, reward-optimal designs acquire 57.5%–100% of optimal information (Figure 2c–d). At $\ell = .2$, $\sigma^2 = .01$, four repeated pairs acquire 76.1% and leave 14.25 times the minimum MSE. At $\ell = 1$, $\sigma^2 = 1$, reward and information prefer four measurements at

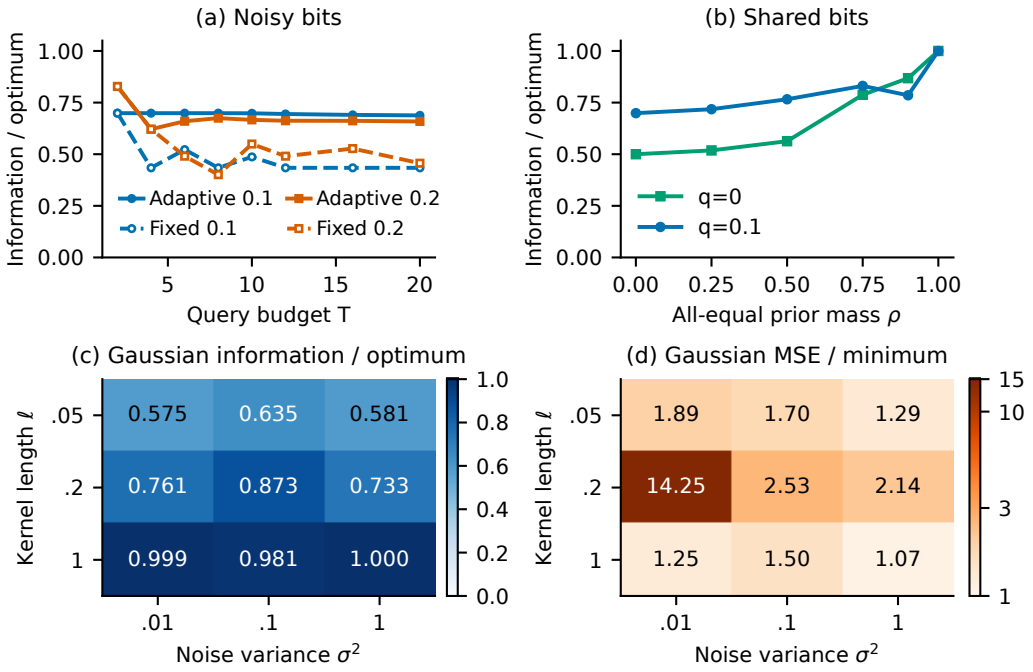


Figure 2: Noise and shared structure change the cost of predictive credit. (a) Adaptive and fixed allocations at flip rates .1 and .2. (b) Four queries under (8). Both show the greatest information among reward optima, divided by the information optimum. (c–d) Exact optimal fixed designs in a Gaussian field: information fraction and reconstruction MSE ratio for all nine settings. MSE colors use a logarithmic scale. Appendix B gives proofs, tie ranges and numerical checks.

each endpoint; reconstruction prefers neighboring interior locations. Agreement on information can therefore coexist with different task preferences.

4 What policies learn to collect

4.1 Changing the reward of an effective policy

We test whether increasing predictive credit redirects a policy that already collects useful measurements. Starting from a policy trained to reconstruct a Gaussian field, we continue training under either deletion reward or the original task score. The field follows the acquisition setting of (Elmoznino et al., 2026), with a rational-quadratic kernel and checkerboard observation noise. Our implementation draws 100 latent grid values and interpolates to 64 query locations, using the matching predictor. The task score is the expected marginal log score over the latent grid; evaluation measures reconstruction MSE. Appendix J gives the implementation and its differences from (Elmoznino et al., 2026).

We train three task policies for 26 queries, using 2,000 updates of 128 new fields per seed. Each checkpoint supplies identical initial parameters to the two continuations. Both receive 2,000 actor-only REINFORCE updates with a leave-one-out baseline and a fixed reward scale. Fresh Adam state, no critic loss and no entropy bonus keep the comparison focused on the actor's response to its reward.

On 4,096 fresh shared fields per seed, switching raises mean deletion reward from 20.00 to 24.97, reduces distinct sites from 18.17 to 12.18, and raises MSE from .1959 to .2692. Continuing task training lowers MSE to .1856, so the switched policies leave 45.1% more error. All three seeds agree in direction.

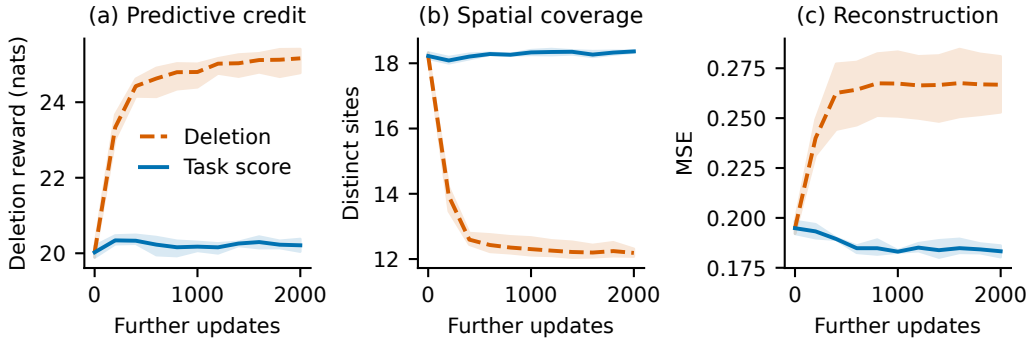


Figure 3: Changing the reward changes acquisition from the same starting policy. Each task-trained policy continues under deletion or task reward using actor-only updates. Lines average three seeds on 512 fixed diagnostic fields per seed; shading spans the seeds. Final estimates in the text use 4,096 separate fields per seed.

The original actor/critic protocol also increases credit and reconstruction error after the switch. Separate training from the original initialization yields less coverage and larger error under deletion at budgets 13 and 26. Appendix J reports both protocols, the original comparisons with random and greedy information acquisition, learning curves and paired uncertainty.

4.2 Learning and refitting the predictor

We next examine whether the acquisition gap survives learning the predictor and improving it on policy data. A free 64×64 latent covariance is fitted to random acquisition records, with known measurement noise. Policies train for 2,000 updates on learned deletion or information rewards. We then refit the covariance on equal amounts of initial and policy-generated records. Each policy continues for 2,000 updates in two branches, one using the refitted predictor and one keeping it frozen. This separates a change in the predictor from additional policy training (Appendix J).

Refitting improves likelihood in all six comparisons on identical fresh trajectories from the original policies, but the acquisition gap remains. After refitting and continued training, full-grid MSE is .2871 under deletion versus .1916 under information reward; keeping the predictor frozen gives .2871 versus .1917. The exact model evaluates the acquired measurements here. Reconstruction using the learned model at the 64 candidate sites also favors information training (.2939 versus .2132). A better fit to policy data therefore coexists with a persistent difference in what the policies collect.

5 Learning from the collecting policy

5.1 Retained actions change masked predictions

So far, actions specify query locations, including when the observation model is learned and refitted. Directly fitting masked predictions to policy records also lets actions serve as evidence. If a policy tends to repeat a query after a positive report, a retained repeated action makes that report more likely even after it is deleted. The masked predictor can recover information through the policy's response.

Let μ be the policy that generates the fitting records. The population target for masked log-loss fitting is the **observational masked conditional**

$$m_\mu(y \mid S_{jk}) = p_\mu(Y_k = y \mid S_{jk}). \tag{9}$$

The source policy now affects how much the deleted observation improves prediction. Appendix A computes m_μ by integrating over the deleted outcome and latent environment. Evaluation uses records supported by the source.

Improving a policy against a fixed predictor changes π while holding m_μ fixed. Refitting on each improved policy's records instead gives $\pi_{t+1} \in \arg\max_\pi R(\pi, m_{\pi_t})$. We also optimize the value of a policy under its own fitted predictor, $V(\pi) = R(\pi, m_\pi)$. Proper log scoring separates this value from the interventional reward:

$$V(\pi) = R(\pi, m_{\mathrm{do}}) - \sum_{j<k} \mathbb{E}_\pi \mathrm{KL}\big(m_\pi(\cdot \mid S_{jk}) \| m_{\mathrm{do}}(\cdot \mid S_{jk})\big). \tag{10}$$

A predetermined paired schedule makes the penalty zero. Thus the budget cost in Proposition 1 survives self-fitting: its optimum, information loss and near-optimal bound all hold. The same argument extends Proposition 2 to weighted self-fitting (Appendix I).

5.2 A STRICT CYCLE AND A SELF-CONSISTENT OPTIMUM

The next example separates stability from information quality. Exact refitting can cycle, while a self-consistent optimum can favor less informative queries. Consider a uniform secret among three candidates and two equality queries, each report flipped with probability .1. The first query is uniform. The second repeats with probability u after a negative report and v after a positive report, otherwise switching uniformly. Allow every $(u, v) \in [.05, .95]^2$. For a fixed source predictor, reward is affine in (u, v), so four corners suffice to optimize the entire rectangle. Write S for .05 and R for .95, with the negative-report response first.

Proposition 3 (refitting response). Exact observational refitting and reward maximization give strict responses SS→RR, SR→RS, RS→SR and RR→RR. RR also maximizes V over the rectangle, while SR maximizes information.

The smallest corner margins range from .057 to .112 nats, certified using rational probabilities and interval bounds on logarithms. RR ignores the report in choosing its second action, has $m_{\mathrm{RR}} = m_{\mathrm{do}}$, and attains the upper bound in (10). Its information is .4774 nats, compared with .6390 for SR. A stable response and a strict cycle therefore coexist in the same class, and the stable reward optimum collects less information (Appendix C).

5.3 SOURCE OPTIMIZATION AND IDENTIFICATION IN MASTERMIND

We next examine what source optimization does to success at a larger identification task. In eight-step Mastermind with six colors and seven positions, feedback has 35 possible pairs of exact-position and total-color matches; with probability .2, it is replaced by a uniformly drawn pair. Policy π_ϵ queries a secret drawn from its posterior with probability $1-\epsilon$ and queries uniformly with probability ϵ. We optimize over $\epsilon \in [0, 1]$ and mixtures that select one policy per episode.

We compute masked predictions by summing over hidden reports and secrets. For episode mixtures, the component weights update after conditioning on the retained record. The self-fitted value is concave in mixture weights, and record densities are polynomials in ϵ. These properties let us optimize over the continuous family and its mixtures (Appendix D).

Optimizing the source on the full trajectory sample or either half favors near-uniform queries. We freeze the first-half source, $\epsilon_s \approx .9998$, and its masked predictor. Under this common predictor, the source policy earns 2.00 more nats of reward than posterior sampling (SE .08) on 2,048 new paired trajectories per target, while acquiring .61 fewer nats of information (SE .02). Terminal identification accuracy is 60.5% versus 81.5%: a 21.0-point gap (SE .77 points). Fresh response estimates slightly favor uniform queries over the fitted source, supporting the near-uniform region. Appendix D gives the sampled-objective certificates and independent response estimates.

Rescoring the same uniform-query and posterior-sampling trajectories isolates the source effect. A uniform-source predictor favors posterior sampling; the near-uniform source reverses that ranking. Although this source rarely uses the posterior rule, posterior-sampling records contain actions that are unlikely under uniform querying. Conditioning on those actions can make a posterior-rule choice

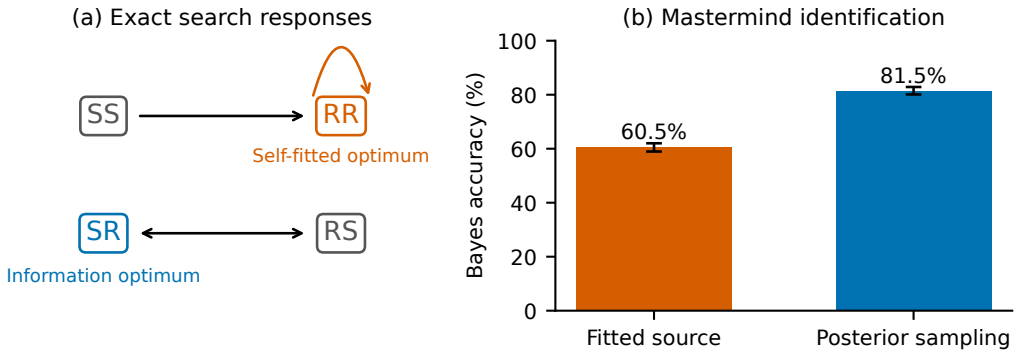


Figure 4: Source fitting changes acquisition and identification. (a) Exact search responses over the complete symmetric rectangle: arrows show strict reward-maximizing responses; RR maximizes the self-fitted value while SR acquires more information. (b) Mastermind identification under the frozen near-uniform source and posterior sampling, using 2,048 new paired trajectories. Error bars are 95% normal intervals conditional on the fitted source.

likely, allowing the masked predictor to recover information about the deleted report (Appendix D.10). Thus a small change in how fitting records are generated can substantially change credit on another policy's records. Appendix F examines how prediction errors alter these preferences.

6 Task consequences in sensor records

The GP and Mastermind comparisons evaluate acquisition against a fixed latent field or secret. In sensor records, missing reports change both what a requested set provides and what can be scored. We use Intel Lab temperature reports (Bodik et al., 2004) to study report reconstruction through two comparisons: first changing the evaluated records, then changing the selection criterion with the model and records fixed (Figure 5; Appendix K).

For Intel, a rolling seven-day window selects 16 candidate sensors and fits their joint Gaussian distribution. We enumerate requests for four or eight five-minute temperature reports, comparing deletion reward with model-predicted reconstruction MSE. The task selector initially assumes every requested report arrives and evaluates all 16 sensors. The complete-record analysis retains 13 test days; fitting incomplete records by Gaussian EM allows evaluation on 21 days. Failed requests consume budget, predictions use only received reports, and both methods are scored on the same available targets. The mean ranking reverses, although deletion has larger MSE on 14/21 days at either budget.

Matched evaluations locate the reversal when additional dates enter (Figure 5a). Replacing the fit and selected sets on the original complete records preserves the task selector's mean advantage. Adding incomplete intervals on those same dates also preserves it: deletion-minus-task MSE is .619 and 1.000 at budgets four and eight. Including eight more dates changes these gaps to -1.429 and -11.684. Crossing old and new sets with both reconstruction models preserves positive gaps on the common dates. March 25 illustrates how availability changes the comparison: only 1.30 targets remain per interval on average, and the eight-request deletion set receives all of them, giving zero reconstruction error.

The second comparison changes which requests the task objective favors. Holding the model and all 21 evaluation days fixed, we average that objective over observed availability patterns (Figure 5b). Training-period patterns still leave deletion with lower mean MSE. Test-period patterns select sets with MSE 8.26 and 2.42 at budgets four and eight, below deletion's 24.91 and 5.81. This diagnostic uses future availability but no test temperatures. Correcting only the assumed received inputs or only the evaluated targets also restores the task selector's mean advantage. The ranking can therefore change through the selection criterion alone, with the same temperature model and test records.

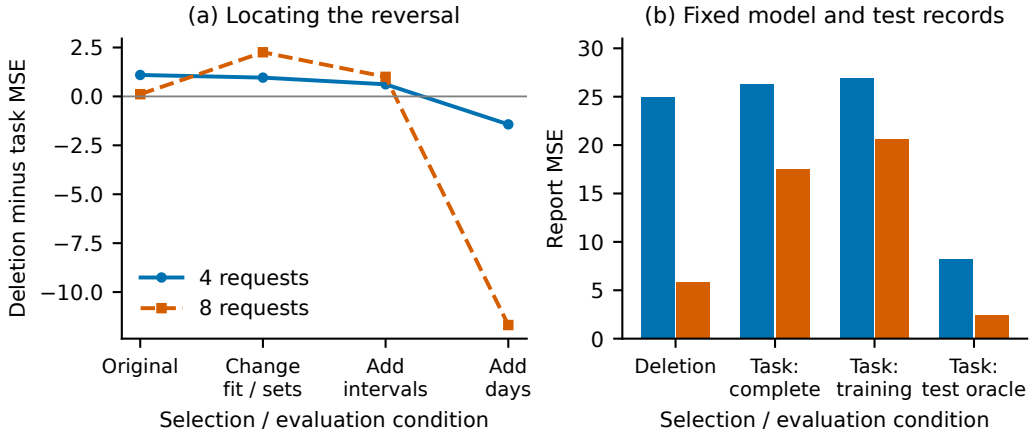


Figure 5: Availability changes the Intel task comparison. (a) Matched evaluations locate the mean reversal when additional dates enter. (b) With the model and 21 evaluation days fixed, task selectors use complete-observation assumptions, training availability or test availability. The test-availability oracle uses no temperature values. MSE is in squared degrees Celsius, averaged equally over days.

Model error also matters. At four requests, even conditioning on actual received inputs and averaging over actual targets, the model predicts that the original task set is better; realized mean error favors deletion. Withholding each target's own report from both decoders reduces the eight-request advantage but retains both mean reversals (Appendix K). The records therefore expose changes in input availability, target composition and predictive accuracy. Their measured quantity is report reconstruction; independent spatial temperature truth is unavailable.

7 Related work and conclusion

Exploration and experimental design. NDIGO (Azar et al., 2019) and CIG (Joseph et al., 2026) use predictive or conditional information signals for exploration. In-context curiosity derives a finite-horizon information residual for the complete predictive-improvement sum in Bayesian experimental design (Elmoznino et al., 2026). Our allocation laws characterize maximizing policies and their information tradeoff at fixed budgets. The fixed-design score is dual total correlation (Han, 1978; Austin, 2020). GP sensor placement (Krause et al., 2008) and prediction-oriented active learning (Bickford Smith et al., 2023) distinguish information objectives and prediction quality; our experiments connect these choices to learned acquisition.

Learning from policy records. Causally correct partial models (Rezende et al., 2020) and causal confusion in imitation learning (de Haan et al., 2019) explain why omitted observations and retained actions matter. Here that dependence changes the fitted reward: exact refitting can cycle or stabilize at a policy with lower information. Performative prediction (Perdomo et al., 2020) and performative reinforcement learning (Mandal et al., 2023) study feedback between fitting and deployment. Our mixture optimization connects conditional entropy to established log-score game theory (Grünwald and Dawid, 2004).

Predictive improvement gives discoveries value through later measurements. Under a finite budget, maximizing that credit can spend queries on confirmation while other unknowns remain unresolved. We characterize this cost and its dependence on scoring times and observation structure, show that reward optimization can redirect an effective learned policy, and explain how policy records change the masked prediction used for credit. The task consequences then depend on what the selected measurements actually reveal, as the sensor interventions illustrate. Designing predictive credit therefore includes choosing which measurements assign it and what the predictor may infer from retained actions.

REPRODUCIBILITY STATEMENT

The supplementary code archive includes experiment, training and audit scripts, their local dependencies, and instructions organized by experiment. Its README and reproduction guide specify the execution order; DATASETS.md links to the original Intel Lab and UCI Gas datasets and gives their expected locations. Datasets and checkpoints are not redistributed: simulated inputs and models are generated by the supplied scripts. The appendices give proofs, model definitions and evaluation protocols.

USE OF AI

The authors used OpenAI Codex to assist with research design, literature discovery, mathematical claims and proofs, software implementation, processing of public sensor data, experimentation, result interpretation and checks, visualization, and manuscript drafting and refinement. Sources were checked against the cited publications, and numerical results were checked by separate calculations and prediction replay. The authors take full responsibility for the final article.

REFERENCES

Tim Austin. **Multi-variate correlation and mixtures of product measures.** Kybernetika, 56(3):459–499, 2020. Paper.

Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo Avila Pires, Jean-Bastien Grill, Florent Altché, and Rémi Munos. **World Discovery Models.** arXiv:1902.07685, 2019. Paper.

Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. **Prediction-Oriented Bayesian Active Learning.** AISTATS, 2023. Paper.

Peter Bodik, Wei Hong, Carlos Guestrin, Sam Madden, Mark Paskin, and Romain Thibaux. **Intel Lab Data.** Sensor records collected in 2004. Dataset.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. **Causal Confusion in Imitation Learning.** NeurIPS, 2019. Paper.

Eric Elmoznino, Sangnie Bhardwaj, Johannes von Oswald, Rajai Nasser, Blaise Agüera y Arcas, João Sacramento, Rif A. Saurous, and Guillaume Lajoie. **Can In-Context Learning Support Intrinsic Curiosity?** arXiv:2606.19476, 2026. Paper.

Jordi Fonollosa. **Gas Sensor Array under Dynamic Gas Mixtures.** UCI Machine Learning Repository, 2015. Dataset.

Peter D. Grünwald and A. Philip Dawid. **Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory.** The Annals of Statistics, 32(4):1367–1433, 2004. Paper.

Te Sun Han. **Nonnegative entropy measures of multivariate symmetric correlations.** Information and Control, 36(2):133–156, 1978. Paper.

Tim Joseph, Marcus Fechner, Philipp Stegmaier, Karam Daaboul, and J. Marius Zöllner. **CIG: Exploration via Conditional Information Gain.** arXiv:2605.20878, 2026. Paper.

Andreas Krause, Ajit Singh, and Carlos Guestrin. **Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies.** Journal of Machine Learning Research, 9:235–284, 2008. Paper.

Debmalya Mandal, Stelios Triantafyllou, and Goran Radanovic. **Performative Reinforcement Learning.** ICML, 2023. Paper.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. **Performative Prediction.** ICML, 2020. Paper.

Danilo J. Rezende et al. **Causally Correct Partial Models for Reinforcement Learning.** arXiv:2002.02836, 2020. Paper.

A PREDICTIVE TARGETS AND LOG-SCORE IDENTITIES

A.1 FULL AND MASKED RECORDS

The stationary observation model has conditionally independent outcomes given the latent environment and queries. A trajectory prefix has density

$$p_\pi(\theta, h_{k-1}, a_k) = p(\theta) \prod_{t<k} p_\theta(y_t \mid a_t) \prod_{t \le k} \pi(a_t \mid h_{t-1}). \tag{A.1}$$

The last product is constant in θ when the full prediction record is observed. Normalization gives the common full conditional f on supported records. When Y_j is hidden, a completion z changes subsequent action probabilities. Integrating (A.1) over the missing outcome gives the conditional in (9). The mask location and query index are part of the record: different hidden locations define different conditional prediction problems.

The interventional target keeps visible query inputs fixed and integrates the omitted outcome before conditioning on later actions. Its likelihood product is (2). When retained action probabilities are independent of the omitted feedback conditional on the visible record, they factor out of the completion sum and the masked targets agree. The SS and RR policies in Section 5, and the uniform random Mastermind source, have this property.

To compute the observational conditional, restore the missing outcome to a possible value z, and write $Y_t^{(z)} = Y_t$ for $t \neq j$ and $Y_j^{(z)} = z$. Define

$$W_\mu(\theta, z; S_{jk}) = p(\theta) \prod_{t<k} p_\theta(Y_t^{(z)} \mid A_t) \prod_{t \le k} \mu(A_t \mid h_{t-1}^{(z)}).$$

Then

$$m_\mu(y \mid S_{jk}) = \frac{\int\int p_\theta(y \mid A_k) W_\mu(\theta, z; S_{jk})\, d\theta\, d\nu(z)}{\int\int W_\mu(\theta, z; S_{jk})\, d\theta\, d\nu(z)}. \tag{A.2}$$

Here ν is counting measure for discrete outcomes and the observation reference measure for continuous outcomes. Off-policy evaluation uses records supported by the source; conditionals are specified up to source-null sets.

A.2 SCORING DECOMPOSITION

Condition the full log score on H_k and the masked log score on S_{jk}. Proper log scoring gives

$$\begin{aligned} \mathbb{E}_\pi \log F(Y_k \mid H_k) &= \mathbb{E}_\pi \log f(Y_k \mid H_k) - \mathbb{E}_\pi \mathrm{KL}(f \| F), \\ \mathbb{E}_\pi \log M(Y_k \mid S_{jk}) &= \mathbb{E}_\pi \log m_\pi(Y_k \mid S_{jk}) - \mathbb{E}_\pi \mathrm{KL}(m_\pi \| M). \end{aligned} \tag{A.3}$$

Writing $A_\pi(F) = \sum_{j<k} \mathbb{E}_\pi \mathrm{KL}(f \| F)$ and $B_\pi(M) = \sum_{j<k} \mathbb{E}_\pi \mathrm{KL}(m_\pi \| M)$ gives $\mathbb{E}_\pi G(F, M) = R(\pi, m_\pi) + B_\pi(M) - A_\pi(F)$. This decomposition uses the target policy's conditional; the fixed-teacher decomposition in (F.1) separates teacher choice from approximation. Taking $F = f$ and $M = m_{\mathrm{do}}$ proves (10). The opposite signs follow from the numerator and denominator positions in the reward. Realized log scores and integrated conditional KL terms have different finite-sample fluctuations even when evaluated on the same trajectories.

B BUDGET ALLOCATION AND SHARED STRUCTURE

B.1 PATHWISE OPTIMUM AND INFORMATION BENCHMARK

For noiseless independent fair coordinates, (4) holds on every realized trajectory, including feedback-dependent choices. Every coordinate credited by the reward consumes at least two queries. With even T, attaining $(T/2)\log 2$ requires exactly two queries at each of $T/2$ coordinates; every such allocation attains the bound. Observing m distinct coordinates reduces posterior entropy by $m \log 2$, since unobserved coordinates remain independent fair bits. For odd T, $\lfloor T/2 \rfloor$ coordinates can be confirmed; one additional fresh query gives the greatest information among reward optima.

For $d \geq T$ and even T, (10) gives $V(\pi) \leq R(\pi, m_{\mathrm{do}}) \leq (T/2)\log 2$. A predetermined paired schedule has identical observational and interventional conditionals and attains the bound. Any self-fitted maximizer therefore also maximizes the interventional reward and must pair its queries. More precisely, self-fitted maximizers are the paired policies whose KL penalty in (10) vanishes. A paired policy that uses feedback to choose later coordinates can have a positive penalty.

For every path let D_T count distinct coordinates and C_T count coordinates queried at least twice. Then $D_T + C_T \leq T$, so $U(\pi) + R(\pi, m_{\mathrm{do}}) \leq T \log 2$. Since $R(\pi, m_{\mathrm{do}}) \geq V(\pi)$, the near-optimal implication in Section 5.1 follows for every $\delta \geq 0$. The same argument gives $U(\pi) \leq \min(d \log 2, T \log 2 - V(\pi))$ at any budget.

With binary symmetric noise q, the posterior after (a, b) reports is

$$\beta_{a,b} = \frac{(1-q)^a q^b}{(1-q)^a q^b + q^a (1-q)^b}. \tag{B.1}$$

Let $h(p) = -p \log p - (1-p)\log(1-p)$. Each query's conditional information is $h(p_{a,b}) - h(q) \leq \log 2 - h(q)$. A fresh coordinate attains equality. Since $d \geq T$, querying a fresh coordinate at every step attains the information optimum $T[\log 2 - h(q)]$, including adaptive alternatives.

B.2 FIXED-COUNT IDENTITY

Write $\Delta I_n = I_n - I_{n-1}$. Exchangeability implies that deleting any one of the $k-1$ earlier reports increases the expected next-report entropy by $\Delta I_{k-1} - \Delta I_k$. Thus

$$\begin{aligned} r_n &= \sum_{k=2}^{n} (k-1)(\Delta I_{k-1} - \Delta I_k) \\ &= I_{n-1} - (n-1)\Delta I_n = n I_{n-1} - (n-1) I_n. \end{aligned} \tag{B.2}$$

We compute I_n by summing $\log 2 - h(\beta_{a,n-a})$ over $a = 0, \ldots, n$ with probabilities

$$\frac{1}{2}\binom{n}{a}[(1-q)^a q^{n-a} + q^a (1-q)^{n-a}]. \tag{B.3}$$

The recursion $C(t) = \max_{1 \leq n \leq t}[r_n + C(t-n)]$, $C(0) = 0$, finds the fixed-count optimum. There are enough coordinates for every partition. Direct enumeration of report strings through $n = 6$ agrees with the formula to 4.56×10^{-15}.

B.3 ADAPTIVE BELLMAN RECURSION

Let s be the multiset of observed count pairs. An action c selects an existing pair (a, b) or creates $(0, 0)$. Let s_c^+ and s_c^- be the states after a report of one or zero. With r remaining queries,

$$V_r(s) = \max_c \{g(c) + p_c V_{r-1}(s_c^+) + (1-p_c) V_{r-1}(s_c^-)\}, \quad V_0(s) = 0. \tag{B.4}$$

The immediate reward is

$$g(a,b) = a\,\mathrm{KL}(\mathrm{Ber}(p_{a,b})\|\mathrm{Ber}(p_{a-1,b})) + b\,\mathrm{KL}(\mathrm{Ber}(p_{a,b})\|\mathrm{Ber}(p_{a,b-1}))\,. \tag{B.5}$$

Zero-multiplicity terms are omitted. The multiplicities count distinct deletion masks. Masks on other coordinates contribute zero. The queried coordinate's expected posterior information gain is

$$d(a,b) = h(\beta_{a,b}) - p_{a,b}h(\beta_{a+1,b}) - (1-p_{a,b})h(\beta_{a,b+1}). \tag{B.6}$$

Alongside V_r, we compute the least and greatest information acquired by reward-optimal policies, using (B.6) and the same transitions. Candidates within 10^{-11} of the greatest reward are treated as tied. We put each count pair in a canonical order by exchanging bit labels, then sort the multiset. These operations preserve gains, rewards and transitions. The noise-free endpoint uses reachable deterministic-report states; $q = .5$ has zero reward and information.

The grid uses $T \in \{2, 4, 6, 8, 10, 12, 16, 20\}$ and $q \in \{0, .05, .1, .2, .4\}$; the zero-information endpoint is checked at $T = 4$. A separate budget-four solver keeps labeled histories, enumerates latent bit vectors and recomputes each deleted-outcome posterior. Its expectations agree with (B.4). Appendix H tabulates the values, including information ranges across tied choices.

For the decisions in Section 3.3, one coordinate has been observed and four queries remain at $q = .1$. Continuing versus opening a fresh coordinate gives optimal remaining rewards .7279 versus .6853 after $(1, 0)$, 1.4912 versus 1.3637 after $(1, 1)$, and .4530 versus .5822 after $(2, 0)$. In each case, we also optimize all subsequent decisions.

B.4 Shared-structure control

The prior in (8) has uniform one-coordinate marginals for every ρ. We enumerate all 16 latent vectors and supported labeled histories at $T = 4$. For each query, the recursion constructs the full posterior, deletes each earlier outcome while keeping actions fixed, and integrates the reward. A separate recursion maximizes information under the same prior and noise law. Each ratio in Figure 2 compares two optima in the same environment.

The grid is $\rho \in \{0, .25, .5, .75, .9, 1\}$ and $q \in \{0, .1\}$. At $\rho = 0$, the result agrees with the independent-coordinate recursion. At $\rho = 1$, all coordinates share one latent bit and every query measures that bit, giving the two objectives the same information. Intermediate values quantify the changing allocation, including the noisy curve's decrease between .75 and .9.

B.5 Finite dimension and near-optimal rewards

Let $T \geq 1$ and $d \geq 1$. On a realized noiseless path, write $C = \sum_i \mathbf{1}\{N_i \geq 2\}$ for the number of confirmed coordinates and $D = \sum_i \mathbf{1}\{N_i \geq 1\}$ for the number discovered. Then $C \leq D \leq d$ and $C + D \leq T$. The reward is $C \log 2$, and posterior entropy reduction is $D \log 2$. These identities hold for adaptive choices because every unobserved coordinate remains a fair bit conditional on the full history.

Set $c^* = \min(d, \lfloor T/2 \rfloor)$. The greatest reward is $c^* \log 2$. For any threshold $0 \leq r \leq c^* \log 2$,

$$\max_{\pi:\, \mathbb{E}_\pi G \geq r} U(\pi) = \min\{d \log 2, T \log 2 - r\}. \tag{B.7}$$

The upper bound follows by taking expectations of $C + D \leq T$. To attain it when $T \leq d$, choose an integer c, query c coordinates twice, and query $T - 2c$ fresh coordinates once. This gives reward $c \log 2$ and information $(T - c) \log 2$. Randomizing between adjacent allocations attains the intermediate expectations.

When $T > d$, full discovery with one or more confirmed coordinates attains $d \log 2$ for every threshold up to $\min(d, T - d) \log 2$: spend the surplus queries on confirmed coordinates. For higher thresholds, the same paired-and-single allocations give $D = T - C \leq d$, and randomization again fills the intervals. A threshold is a lower bound on reward; it need not be attained with equality when full discovery already satisfies it.

At the reward optimum and $T \geq 2$, the least and greatest information are $c^* \log 2$ and $\min(d, T - c^*) \log 2$. The least is attained by spending surplus queries on confirmed coordinates; the greatest by discovering an additional coordinate when the budget allows. At $T = 1$, every policy acquires $\log 2$. Thus odd budgets may admit reward optimizers with different information, while $T \geq 2d$ forces every reward optimizer to confirm all d coordinates. For even $T \leq d$, allowing a reward shortfall δ permits at most δ additional nats of information until the information optimum is reached.

We enumerate all integer allocations for $d = 1, \ldots, 7$ and $T = 1, \ldots, 14$. For 17 reward thresholds per cell, a separate linear program optimizes information over randomized allocations. All 98 cells agree with (B.7), with maximum discrepancy 8.89×10^{-16} in units of $\log 2$.

B.6 Independent scoring probes

Keep the noiseless coordinate model and fixed-action masked posteriors. Let $N_i(t)$ count acquisitions of coordinate i through time t. A probe selects coordinate i with fixed probability p_i, independently of the acquisition history, and observes the same latent bit. Its outcome is withheld from the acquisition policy. Deleting an acquisition changes the probe prediction only when it was the unique observation of the probed coordinate: the full predictor then assigns probability one to its value and the masked predictor assigns one half.

If an independent probe replaces each scoring query at its original time $k = 2, \ldots, T$, the expected reward conditional on the acquisition path is

$$R_{\text{replace}} / \log 2 = \sum_{k=2}^{T} \sum_{i=1}^{d} p_i \mathbf{1}\{N_i(k-1) = 1\}. \tag{B.8}$$

Alternatively, one terminal probe can score all T acquisition deletions:

$$R_{\text{terminal}} / \log 2 = \sum_{i=1}^{d} p_i \mathbf{1}\{N_i(T) = 1\}. \tag{B.9}$$

For uniform probes and $d \geq T \geq 2$, (B.8) is maximized by making the first $T - 1$ acquisitions distinct, attaining $T(T-1)/(2d) \log 2$. The final acquisition receives no credit, so optimal policies acquire between $T - 1$ and T bits. In contrast, (B.9) requires all T acquisitions to be distinct and attains $(T/d) \log 2$ with T bits of information. Both results follow by bounding the number of singletons at each scored prefix by its length.

Terminal deletion credit differs from incremental information. If $T > d$, its maximum falls to $((d-1)/d) \log 2$: preserve $d-1$ singletons and spend the remaining queries on the last coordinate. These policies still discover all d bits. Incomplete coverage can leave information differences unscored: with $d = T = 3$ and $p = (1, 0, 0)$, paths $(0, 1, 1)$ and $(0, 1, 2)$ earn the same terminal reward but acquire two and three bits. Direct latent-posterior enumeration checks (B.9) on all 430 paths for $d = 1, \ldots, 3$ and $T = 1, \ldots, 5$, with zero discrepancy. Separate enumeration at $d = T = 2, \ldots, 5$ confirms the information range in (B.8).

Physical implementation requires $T - 1$ additional measurements for (B.8), or one for (B.9). The information above excludes these evaluation outcomes. A uniform terminal probe adds an expected $(d - T)/d$ bits if its outcome is also counted after T distinct acquisitions with $T \leq d$. Analytically integrating probes instead requires the known observation model. Thus this intervention identifies the role of scoring measurements with explicit costs and coverage conditions.

B.7 Learning the coordinate allocation

At $d = T = 8$, we train coordinate-selection policies under the original reward, the analytically integrated uniform terminal-probe reward, or information gain. A shared network scores each coordinate from seven inputs: its count class (zero, one, or at least two), the three global class fractions, and normalized time. A 32-unit tanh layer precedes the scalar output. The policy samples from the eight logits without action masking. These count classes suffice for the noiseless objectives, so report values are marginalized by bit symmetry.

The gradient estimator multiplies the complete episode return, minus a leave-one-out baseline, by the sum of all eight action log probabilities. Later actions therefore receive credit for their effect on scores of earlier discoveries. Each objective uses three seeds, 800 REINFORCE updates of 512 episodes, Adam with learning rate .003, and a leave-one-out batch reward baseline. There is no entropy bonus, predictor training, or refitting. We evaluate each final model on 16,384 new sampled episodes and independently propagate its exact terminal distribution over the three coordinate-count classes. All 27 reward/information comparisons lie within 1.61 Monte Carlo standard errors of the exact values.

Training objective	Exact information (bits)	Exact objective reward, divided by $\log 2$	Optimal-allocation probability
Original deletion	4.000686–4.000944	3.998048–3.998490	.998048–.998490
Terminal probe	7.999090–7.999253	.999773–.999813	.999091–.999253
Information	7.998926–7.999182	7.998926–7.999182	.998927–.999183

Ranges span the three seeds. Optimal allocation means four repeated pairs for the original objective and eight distinct coordinates for the other two. Sampled policy learning therefore realizes the exact allocation preferences in this sufficient-state environment.

B.8 Continuous spatial measurements

Let f be a zero-mean Gaussian field on $[0,1]$ with covariance $K(x,x') = \exp[-(x-x')^2/(2\ell^2)]$. Eight equally spaced sensor locations support a predetermined schedule of eight measurements, each with independent Gaussian noise of variance σ^2. Repeated measurements are allowed. We optimize the expected reward in (1) with m_{do}, terminal information about the field, and posterior reconstruction risk averaged over 201 equally spaced prediction locations.

For a fixed schedule, the expected contribution of deleting measurement j to prediction k is one half the log ratio of masked to full predictive variances. Write n_i for the previous count at location i and S_n for the field posterior covariance at the sensor locations. Deleting one measurement at i gives the rank-one covariance update $S_n + S_n e_i e_i^\top S_n/(\sigma^2 - S_{n,ii})$. The expected reward increment for the next measurement at a is consequently

$$c(n,a) = \frac{1}{2}\sum_{i:n_i>0} n_i \log\left[1 + \frac{S_{n,ai}^2}{(\sigma^2 - S_{n,ii})(\sigma^2 + S_{n,aa})}\right]. \tag{B.10}$$

A forward count-state dynamic program maximizes cumulative reward over all 8^8 schedules. There are 6,435 terminal count vectors; direct comparison gives the information and reconstruction optima. Reward ties within 10^{-10} nats are resolved by greatest terminal information. The complete expected reward depends only on the measurement counts, by (I.2); the recursion accumulates its increments along each prefix.

Length ℓ	Noise variance	Reward-optimal information / optimum	Reward-optimal reconstruction risk / minimum
.05	.01	.5746	1.8947
.05	.1	.6349	1.7037
.05	1	.5805	1.2939
.2	.01	.7606	14.2491
.2	.1	.8728	2.5271
.2	1	.7328	2.1373
1	.01	.9988	1.2496
1	.1	.9806	1.4955
1	1	1.0000	1.0695

At $\ell = .2$, $\sigma^2 = .01$, the reward-optimal counts are $(2,0,2,0,0,2,0,2)$; information and reconstruction both prefer one query at each location. Their information values are 10.4624 and 13.7554 nats, and reconstruction risks are $.116436$ and $.008171$. Repeated measurements reduce uncertainty

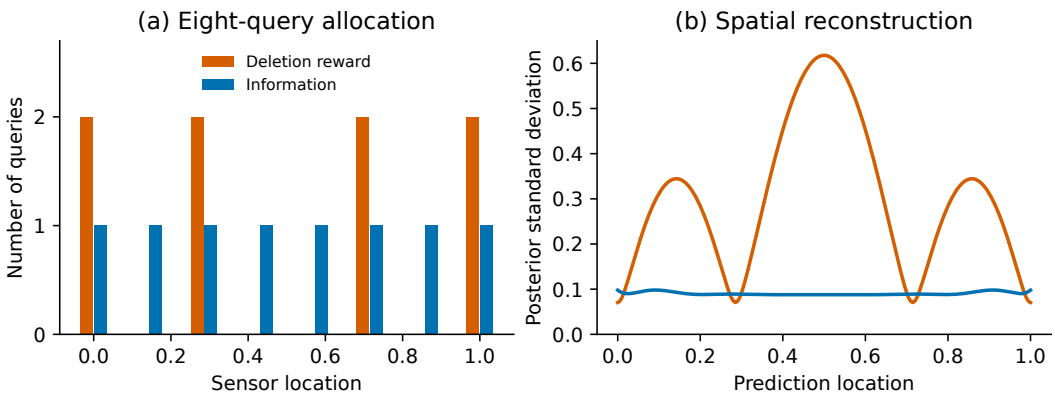


Figure 6: Continuous spatial measurements with length .2 and noise variance .01. (a) Deletion reward concentrates eight queries at four locations. (b) Repeated local measurements leave larger posterior uncertainty between the sensors. Values are exact Gaussian expectations for optimal predetermined schedules. The table reports all nine settings.

locally while leaving gaps between sensors. At $\ell = 1$, $\sigma^2 = 1$, reward and information both choose four measurements at each endpoint. The reconstruction optimum instead moves those measurements to the two neighboring interior locations. Thus the information and reconstruction criteria can also favor different designs.

An independent calculation forms the full covariance of each reported repeated-observation sequence and computes every masked Schur complement. All 27 schedule rewards agree within 2×10^{-11} nats. Exhaustive enumeration of all 81 schedules in a three-location, four-query instance agrees with the dynamic program within 10^{-10} nats. GP sensor placement provides the experimental-design context (Krause et al., 2008); here the comparison isolates the allocation induced by predictive-deletion credit.

C Complete search response and strictness

C.1 Joint distribution and policy class

Let N be the number of candidates and q the flip probability. Set $L(\theta, a, y) = 1-q$ if $y = \mathbf{1}\{\theta = a\}$, and q otherwise. For (u, v), write $r_0 = u$, $r_1 = v$ and

$$K_{u,v}(b \mid a, y) = \begin{cases} r_y, & b = a, \\ (1 - r_y)/(N - 1), & b \neq a. \end{cases} \tag{C.1}$$

The joint mass of (θ, a, y, b, z) is

$$P_{u,v}(\theta, a, y, b, z) = N^{-2} L(\theta, a, y) K_{u,v}(b \mid a, y) L(\theta, b, z). \tag{C.2}$$

This is the complete permutation-equivariant two-query class: symmetry forces a uniform first query and uniform switching among other candidates. The remaining choices are $u, v \in [\delta, 1 - \delta]$, where $0 < \delta < 1/2$.

The full predictor sums the product of the two likelihoods over θ and normalizes by the first likelihood sum. The masked predictor conditions (C.2) only on (a, b), summing over θ, y. For a fixed source predictor, its log scores are constant and the target mass is affine in (u, v). Target reward is therefore affine over the rectangle, so its four corners suffice. Information is also affine because the full posterior is common across target laws.

C.2 STRICT PRIMARY RESPONSE

At $N = 3$, $q = 1/10$ and $\delta = 1/20$, all masses and predictive probabilities are rational. To certify the comparisons independently, we bound each logarithm by rational numbers. For $x > 0$, set $t = (x-1)/(x+1)$. With $m = 160$ terms,

$$\log x = 2\sum_{k=0}^{m-1} \frac{t^{2k+1}}{2k+1} + e_m, \qquad |e_m| \le \frac{2|t|^{2m+1}}{(2m+1)(1-t^2)}. \tag{C.3}$$

Summing weighted intervals gives the rewards below. The strict margin subtracts each competitor's upper interval endpoint from the winner's lower endpoint, then takes the minimum.

Source	SS	SR	RS	RR	Strict margin
SS	.057188936	.130081758	.113843938	.186736760	.056655002
SR	.089919181	.027601703	.479809726	.417492248	.062317479
RS	.135763121	.421280920	.023663946	.309181744	.112099175
RR	.057188936	.130081758	.113843938	.186736760	.056655002

Every margin is positive. For m_{do}, no feedback remains and every candidate has noisy equality probability $q + (1-2q)/N = 11/30$. RR ignores the first report in selecting its second query, so its observational predictor equals this target. The RR row certifies its reward maximum; (10) proves its self-fitted optimum.

A floating-point grid additionally uses $N \in \{3, 4, 8\}$, uniform-replacement probability $\eta \in \{.05, .2, .5\}$ (flip probability $q = \eta/2$), and $\delta \in \{.01, .05, .1\}$. Each self-fitted optimization uses five initializations and checks all four corner responses. Appendix H reports the grid. The rational sign certificate applies to the primary cell.

C.3 WHICH INITIAL SOURCES ENTER THE CYCLE?

For the fixed three-candidate model, the first report is positive with probability $p = 11/30$. Conditional on a negative or positive first report, the probabilities of a positive second report are

$$(a_0, a_1) = (.1421052632, .7545454545), \qquad (b_0, b_1) = (.4789473684, .1727272727) \tag{C.4}$$

for repeating and switching, respectively. The exact values follow from $a_0 = q + (1-2q)q/[N(1-p)]$, $a_1 = q + (1-2q)(1-q)/(Np)$ and $b_y = (Np - a_y)/(N-1)$. For source (u, v), let $z = (1-p)u + pv$. Its masked probabilities after repeated and switched queries are

$$q_R = \frac{(1-p)ua_0 + pva_1}{z}, \qquad q_S = \frac{(1-p)(1-u)b_0 + p(1-v)b_1}{1-z}. \tag{C.5}$$

Writing $D(x\|y)$ for Bernoulli KL, the slopes of target reward with respect to its negative- and positive-report repeat probabilities are

$$\begin{aligned} c_0 &= (1-p)[D(a_0\|q_R) - D(b_0\|q_S)], \\ c_1 &= p[D(a_1\|q_R) - D(b_1\|q_S)]. \end{aligned} \tag{C.6}$$

A positive slope selects R in that coordinate and a negative slope selects S. Away from zero slopes, the first response is a unique corner. Proposition 3 then determines the entire subsequent sequence: SS or RR reaches RR within two responses, whereas SR or RS enters the strict two-cycle. At zero slopes, the maximizing response includes an edge or the rectangle; we retain this set rather than imposing a tie-breaking convention.

At $u = .5$, the numerical boundaries are $v = .2199802121$ and $.9322261809$. Between them the first response is RR; below the first it is SR and above the second it is RS. Formula (C.6) agrees with direct latent-state enumeration at 260 source points within 3.34×10^{-16}. The figure plots the sign regions on a 501×501 grid and locates boundary roots numerically.

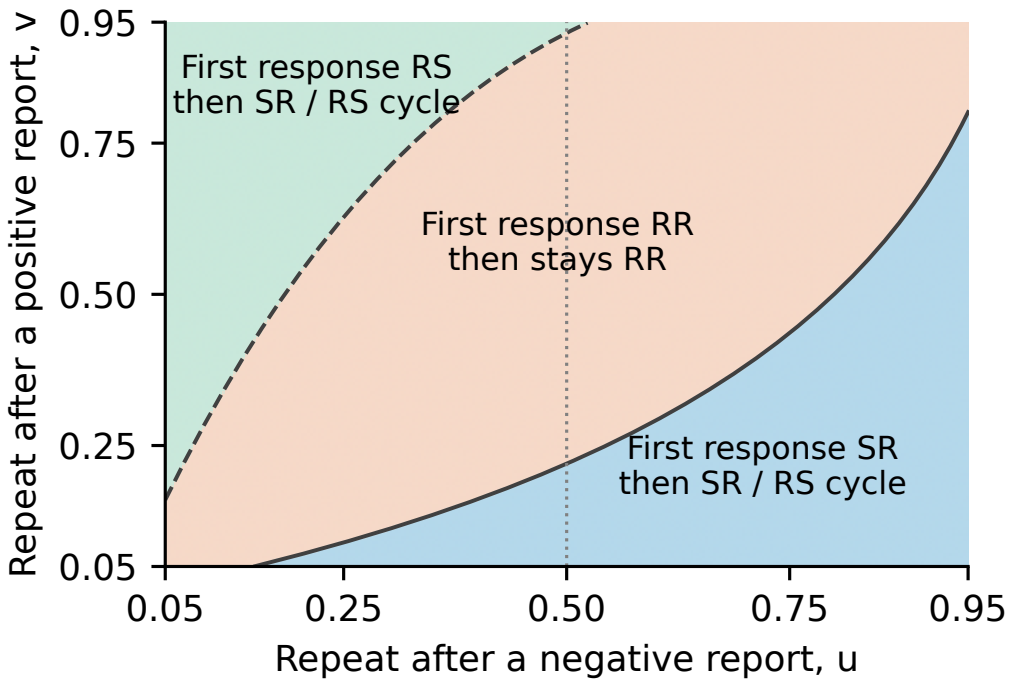


Figure 7: Initial sources lead to either the self-consistent policy or the strict cycle. Colors and labels give the first best response under exact fitting; lines mark zero reward slopes from (C.6). The dotted line is $u = .5$. The model and policy bounds are identical to Proposition 3. Responses on the boundary lines are set-valued.

D MASTERMIND CONDITIONALS AND EPISODE MIXTURES

D.1 OBSERVATION MODEL AND POLICIES

Secrets and guesses are vectors in $\{0, \dots, 5\}^7$. A clean feedback pair (b, w) gives exact-position matches and total matched color multiplicity. Its alphabet is $\{(b, w) : 0 \le b \le w \le 7\} \setminus \{(6, 7)\}$, with 35 elements. For clean feedback $c(\theta, a)$, the likelihood is

$$L_\theta(y \mid a) = .81\{y = c(\theta, a)\} + .2/35. \tag{D.1}$$

The prior is uniform on 6^7 secrets. The policy uses the exact full posterior b_t and kernel $\pi_\epsilon(a \mid h_t) = (1 - \epsilon)b_t(a) + \epsilon/6^7$. The first query is uniform. Every episode has eight queries, including episodes that have already queried the true secret. This fixed horizon is part of the objective.

For each mask/query pair, we restore each possible hidden report, update the posterior over secrets and accumulate the likelihood of the retained actions. Equation (9) sums predictive probabilities and record densities. The float64 vectorized GPU implementation is checked against an explicit CPU loop over hidden feedback and secret states. The four-policy calculation's largest checked predictive discrepancy is 6.65×10^{-14}, and record log-ratio discrepancy is 1.86×10^{-13}. All 16 pure-source/target rewards agree with the separate payoff map within 9.19×10^{-12}.

D.2 MIXTURE GEOMETRY

The relation between optimal prediction under log scoring, entropy and game-theoretic equilibrium is developed in robust Bayesian decision theory (Grünwald and Dawid, 2004). Here the mixture objective is conditional entropy minus an affine full-history term. We derive its concavity and source-payoff conditions below.

Let $Q_i^{jk}(s, y)$ be the joint law of (S_{jk}, Y_k) under source i, and $Q_w^{jk} = \sum_i w_i Q_i^{jk}$. The conditional $Q_w(s, y)/Q_w(s)$ gives

$$m_w(y \mid s) = \sum_i \alpha_i(s) m_i(y \mid s), \qquad \alpha_i(s) = \frac{w_i p_i(s)}{\sum_\ell w_\ell p_\ell(s)}. \tag{D.1a}$$

A mixture chooses its source once per episode and has record-dependent posterior weights. This differs from changing ϵ inside a per-action kernel.

The common full conditional gives

$$V(w) = \sum_{j<k} \left[H_w(Y_k \mid S_{jk}) - \sum_i w_i H_i(Y_k \mid H_k) \right]. \tag{D.2}$$

For a simplex tangent direction v, let $\Delta = \sum_i v_i Q_i$. When the joint probabilities are strictly positive,

$$D^2 V(w)[v, v] = \sum_{j<k} \left[-\sum_{s,y} \frac{\Delta(s, y)^2}{Q_w(s, y)} + \sum_s \frac{\Delta(s)^2}{Q_w(s)} \right] \le 0. \tag{D.3}$$

Cauchy–Schwarz within each s gives the inequality; continuity extends concavity to the boundary. Transferring weight from i to j has directional derivative $R_j(m_w) - R_i(m_w)$. A maximizing mixture equalizes returns for components with nonzero weight, and no other candidate has a greater return. These inequalities also suffice by concavity. Since reward is linear in the target law, the response comparison covers the convex hull of the four policies.

D.3 Estimation and uncertainty

We evaluate each target policy on 2,048 trajectories, paired across policies. We integrate the final query's reward over its conditional report distribution and use realized reports for earlier queries, summing all 28 mask/query terms per trajectory. This preserves expected reward and reduces final-query variance. Exact conditional probabilities are used inside these Monte Carlo expectations.

We check all six two-policy supports, solve for source log odds that equalize the two corresponding returns, and evaluate all candidates. The fitted mixture places weight on $.5/1$, with log odds -19.70725423 and .5 weight $2.76215618 \times 10^{-9}$. Returns for 0 and .1 fall below the common value by 3.0981305 and 2.4450172 nats. Fitting the first 1,024 trajectories gives log odds -20.05567433. On the second half, the return difference between .5 and 1 is .115639 (SE .105765); the differences of 0 and .1 from 1 are -3.01509 and -2.31497.

A paired trajectory bootstrap uses 4,000 replicates and interpolates the equal-return root on a 51-point local log-odds grid. Every replicate retains the $.5/1$ support, with returns for 0 and .1 below those for the two supported components. The 95% log-odds interval is $[-20.58120360, -18.85847315]$. We also resample the 8,192 paired trajectories used to evaluate information, propagating uncertainty in both the source weight and acquired information: the fitted mixture acquires .594805 fewer nats than posterior sampling, with interval $[.571676, .616755]$. For the independent evaluation below, we fix the computed weight before generating new trajectories.

D.4 Source changes and candidate selection

The endpoint comparison below evaluates the pure-random source predictor and the fitted mixture on the same trajectories and with the same exact full predictor. Appendix H gives all rewards and standard errors. Holding the first-half weight fixed at $1.94953621 \times 10^{-9}$, the second-half means for the random-source predictor are 8.7198, 8.5181, 7.6078, 5.9913; for the mixture they are 2.9762, 3.6763, 6.1070, 5.9913. The first comparison favors 0 over .1 by .201739 nats (paired SE .059366). The second favors the $.5/1$ group over $0/.1$, with uncertainty over which of .5 and 1 is best.

A paired bootstrap of the 1,024 evaluation trajectories, holding this weight fixed, selects 0 in 99.975% of 4,000 replicates for the random-source predictor. For the mixture it selects .5 in 86.1%

and 1 in 13.9%. These comparisons concern the reward defined by each source predictor; the information acquired by a candidate remains a separate quantity.

D.5 Source-value proximity and response curves

Let $C \in \{\text{base}, r\}$ identify the episode source, with probability w of the rare source. The common full conditional makes its entropy contribution affine in w. The conditional-entropy mixture identity therefore gives

$$V(w) - [(1-w)V_{\text{base}} + wV_r] = \sum_{j<k} I(C; Y_k \mid S_{jk}). \tag{D.4}$$

Each summand lies in $[0, h(w)]$. Each source's own value is a sum of L conditional mutual informations and lies in $[0, L \log K]$. Consequently,

$$-Lw \log K \le V(w) - V_{\text{base}} \le L[h(w) + w \log K], \tag{D.5}$$

which establishes the source-value bound for this episode-mixture construction. At the fitted weight, the upper bound is 1.877×10^{-6} nats. This is a population comparison between source values, not a numerical certificate that the estimated equal-return weight is the population maximizer.

We evaluate 247 weights along the same $.5/1$ source edge. The three changes in the empirical maximizing candidate occur at log odds -47.2202, -38.7891 and -19.7073, corresponding to weights 3.1084×10^{-21}, 1.4259×10^{-17} and 2.7622×10^{-9}. Separately solving on each half preserves the sequence $0 \to .1 \to .5 \to 1$; the respective half-sample weight pairs are $[4.039, 2.281] \times 10^{-21}$, $[2.886, .628] \times 10^{-17}$ and $[1.950, 3.851] \times 10^{-9}$. This checks stability within the retained bank, rather than providing another independent replication.

At the four representative weights in the source-path figure in Appendix D.5, the winner's estimated margins over the strongest competitor are .2189, .1426, 1.1699 and .3021 nats. A 2,000-replicate paired bootstrap gives intervals $[.1322, .3050]$, $[-.0059, .2794]$, $[1.0290, 1.2847]$ and $[.1537, .4489]$. At the second weight, .1 wins 96.75% of replicates and .5 wins the remainder. All four reported winners agree across the two halves and when changes in the masked score are integrated under the target's own conditional distribution. Appendix H gives the values and information outcomes.

The fitted weight balances a specific payoff change. Relative to the random-source predictor, the .5 candidate initially earns 1.5474 more nats than candidate 1. At the root, conditioning on the rare source reduces its reward by that amount while leaving the random candidate's reward almost unchanged. Predictions at steps seven and eight contribute 1.2649 nats of this reduction. Thus the root is determined by the aggregate effect of record likelihood ratios on candidate rewards, with most of the adjustment occurring in late predictions.

For stable numerical evaluation, we compute the change in masked log score as $\log[1 + \alpha(s)(m_r/m_{\text{base}} - 1)]$ using `log1p`. Direct two-component mixture arithmetic agrees within 9.33×10^{-15} on checked weights. At the fitted root, the paired estimate of $V(w) - V_{\text{base}}$ is 5.28×10^{-10} (SE 8.03×10^{-10}); integrating the masked outcomes conditionally instead gives -1.57×10^{-10} (SE 2.15×10^{-10}). These estimates do not resolve the sign. The source-value comparison in the main text uses (D.5).

D.6 Budget dependence of the fitted source scale

To vary the acquisition budget without changing the environment or policies, we truncate the retained trajectory banks to $T = 2, \ldots, 8$. Prefix conditionals remain valid because the policies do not depend on the horizon. At each budget we integrate its final report, fit the $.5/1$ source edge, and check the other two candidates' rewards. These are paired prefix comparisons; the horizons are not independent replications.

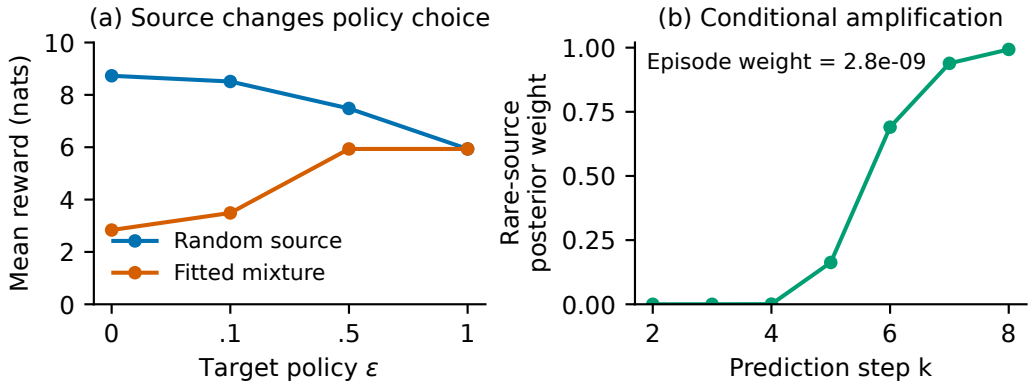


Figure 8: Four-candidate episode-mixture analysis: endpoint rewards and independent conditional amplification. The left panel compares the random-source predictor with the fitted mixture on the original paired trajectories. The right panel uses 1,024 independent posterior-sampling trajectories and averages rare-source posterior weights across eligible masks at each prediction step. The episode weight is fixed at 2.762×10^{-9}.

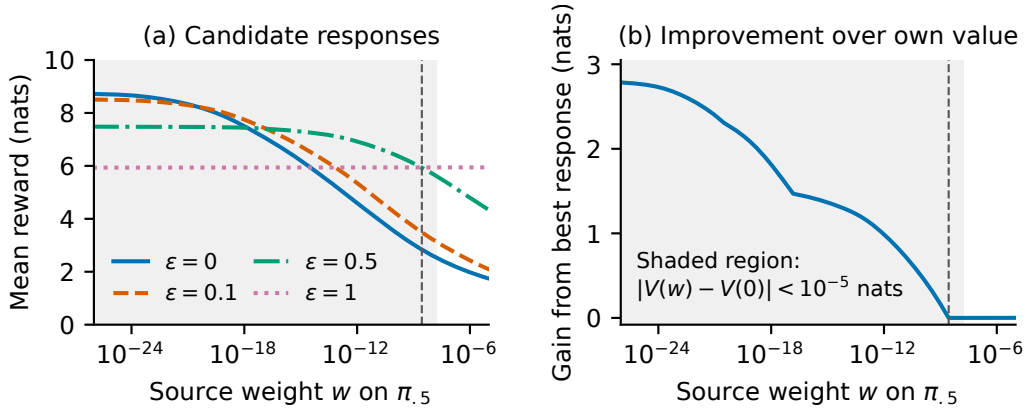


Figure 9: Four-candidate episode-mixture analysis: similar source values induce different acquisition responses. (a) Rewards for four candidates as the weight on source .5 changes; the other source is random. (b) The greatest candidate reward minus the source's own reward. Shading marks weights for which (D.5) bounds the difference from the random source's value by 10^{-5} nats. The dashed line marks the fitted equal-return weight. Curves use the same 2,048 paired trajectories per candidate.

Budget	Full-bank weight on .5	First-half weight	Second-half weight
4	2.5066×10^{-4}	4.6646×10^{-5}	7.2993×10^{-4}
5	3.0650×10^{-5}	1.2783×10^{-5}	6.9288×10^{-5}
6	2.5039×10^{-6}	1.9789×10^{-6}	3.3079×10^{-6}
7	1.1564×10^{-7}	1.2894×10^{-7}	1.0278×10^{-7}
8	2.7622×10^{-9}	1.9495×10^{-9}	3.8511×10^{-9}

For these five budgets, all three fits satisfy the empirical four-candidate payoff conditions. At budgets two and three, the full-bank edge calculation instead selects the random endpoint, which fails the excluded-candidate check. At budget three, the second half alone has an interior root. The short-budget results therefore do not establish the same support or a full-hull optimum.

On posterior-sampling records, median log likelihood ratios at prediction steps two through eight are approximately $0, 3.00, 7.89, 14.51, 23.27, 33.61, 45.21$. On target .5, the step-eight median is

18.10, compared with -4.45 on random-target records. The weight balances rewards on the $.5/1$ targets, while the larger predictive amplification is measured on target 0. These different target laws explain why a single typical likelihood ratio does not determine the equal-return weight. The paired budget trend establishes finite-horizon scale dependence, not an asymptotic scaling law.

D.7 Expanding targets under fixed source predictors

We test the original candidate restriction using an independent bank of 1,024 paired trajectories for each target $\epsilon \in \{0, .025, .1, .25, .5, .75, 1\}$. Targets and five source weights were fixed before generation. This comparison holds source predictors at mixtures of .5 and 1 with log odds $-\infty, -40, -25, -19.7072542334, -15$. Reward evaluation integrates the final report. Three independent CPU hidden-feedback calculations on the new targets agree with the GPU conditional predictions and record log likelihood ratios within 2×10^{-9}.

Target ϵ	Random-source reward	Reward at fitted source weight
0	8.6864	2.8389
.025	8.6227	2.9461
.1	8.4874	3.2703
.25	8.2083	4.3048
.5	7.6044	5.8376
.75	6.9738	6.6908
1	6.1398	6.1398

At the previously fitted weight, target .75 exceeds targets $0, .1, .5, 1$ by $3.8519, 3.4205, .8531, .5510$ nats. Their paired standard errors are $.1196, .1189, .0977, .0761$. Bonferroni-adjusted normal intervals, simultaneous over all 12 new-versus-original comparisons at this fixed predictor, have lower endpoints $3.5092, 3.0798, .5732, .3328$. These are asymptotic intervals rather than finite-sample certificates. The original source mixture therefore admits an improving response in the expanded target class; its original hull restriction is consequential. Target .75 acquires .3550 fewer nats than target 0 (paired SE .0250), estimated on this same bank with terminal-report integration.

Across the five fixed predictors the empirical winning targets are $0, .25, .75, .75, .75$. At log odds -40 and -25, the new winner's adjusted comparison with the strongest original competitor includes zero. At -15, target .75 exceeds every original candidate under the same per-predictor correction, with smallest lower endpoint .0104. Each interval family covers the 12 comparisons for its fixed predictor. This improving response motivates optimization over the continuous source family below.

D.8 Optimization over the continuous source family

For $\epsilon \in [0, 1]$, each action likelihood relative to uniform querying is $\epsilon + (1 - \epsilon)Np(\theta = A_t \mid h_{t-1})$. The first action is uniform, so an eight-step full-record density is a polynomial of degree at most seven in ϵ. Summing over a deleted report preserves that degree for the masked joint law. We compute its coefficients in the nonnegative basis $\epsilon^d(1 - \epsilon)^{k-d}$ at prediction step $k + 1$. The source mixture enters through expectations of these basis functions.

Let $J_{e,p}(s, y)$ be the masked joint density under source e, divided by the masked-record density of an equal mixture of the seven proposal policies. Let $C_{e,p}(H)$ be the corresponding full-record density ratio, and $h_f(H)$ the full predictive entropy. For an episode mixture W, the empirical self-fitted objective is

$$\widehat{V}(W) = \frac{1}{M} \sum_{b,p} \left[- \sum_{y} J_{W,p}(S_{bp}, y) \log m_W(y \mid S_{bp}) - C_{W,p}(H_{bp}) h_f(H_{bp}) \right]. \tag{D.6}$$

Balanced sampling from the seven proposals estimates their mixture expectations. Each entropy perspective in (D.6) is concave in W. Its directional score $\widehat{g}_W(e)$ replaces J_W, C_W by J_e, C_e while holding m_W fixed. This is a degree-seven polynomial, globally maximized by evaluating endpoints and real stationary roots in $[0, 1]$. Concavity gives the empirical certificate

$$\max_{W'} \widehat{V}(W') - \widehat{V}(W) \le \max_{e\in[0,1]} \widehat{g}_W(e) - \widehat{V}(W). \tag{D.7}$$

We first optimize mixture weights by fully corrective conditional-gradient updates and then refine the resulting near-uniform single-policy source. The two halves use 512 common-seed clusters each, with seven trajectories per cluster. The full fit uses all 1,024 clusters. Every refined source has its own parameter as the maximizing response; the residual in (D.7) is below 2×10^{-10} nats.

Fit bank	Source ϵ	Posterior-sampling probability $1-\epsilon$
First half	.9998209695	1.7903×10^{-4}
Second half	.9999514244	4.8576×10^{-5}
Full	.9999078056	9.2194×10^{-5}

The continuous family therefore admits an empirically optimal source that mixes query rules within every episode. The rare episode mixture in the earlier construction and this nearly uniform per-query policy provide different source laws. Coefficient reconstruction reproduces the independently implemented predictors for sources $.5, .75, 1$ within 2×10^{-10}; new-source validation uses direct hidden-report enumeration.

D.9 Fresh response and terminal decision utility

We freeze the first-half source at $\epsilon_s = .9998209694929553$ and generate 2,048 new trajectories for each target below, using common random numbers across targets. Exact reference evaluation integrates the final feedback report. Independent CPU enumeration agrees with all seven new-source predictive checks within 4×10^{-13} and log density ratios within 6×10^{-13}.

Target ϵ	Reward under frozen source	Paired reward advantage over source	Acquired information
0	3.8471	$-2.0046 \pm .0790$	11.8744
.5	4.7286	$-1.1231 \pm .0695$	11.6453
.75	5.3464	$-.5054 \pm .0573$	11.5021
.9	5.7272	$-.1245 \pm .0386$	11.3643
.99	5.8499	$-.0018 \pm .0119$	11.2723
ϵ_s	5.8517	0	11.2623
1	5.8517	0	11.2623

Entries after $\pm$ are one standard error. Bonferroni-adjusted normal intervals over the six frozen comparisons place targets $0, .5, .75, .9$ below the source. The realized draws contain zero posterior-branch selections under ϵ_s, so its paired trajectories coincide with those of 1. To estimate this small difference using the richer proposal records, we also form the degree-seven target-response polynomial by balanced importance sampling of complete-record densities. The response curve peaks at 1, whose estimated advantage over ϵ_s is .0004266 nats (SE .0001925). Integrating every step's score gives .0005035 (SE .0001666). The fitted near-uniform region is stable, while the precise interior parameter varies across samples.

Terminal decision utility is $\max_\theta p(\theta \mid h_8)$, the conditional success probability of the Bayes MAP decision. Replaying the exact saved draws gives mean success .814828 for posterior sampling and .604843 for the frozen source. The paired difference is .209984 (SE .007742). The corresponding information difference, with terminal-feedback integration, is .612104 nats (SE .018182). Thus the reward preference has a substantial consequence for identifying the secret, even though both policies collect much of the prior information.

D.10 Conditional action rules under the frozen source

We analyze the source frozen before the independent evaluation, $e = .9998209694929553$, on exactly the saved trajectories of Appendix D.9. At each query after the first, let a latent coin select

the posterior rule with probability $1 - e$. The first query is uniform under either rule. At prediction step k, the event that all $k - 1$ subsequent query choices used the uniform rule has prior probability e^{k-1}. Conditional on this event, the source has the uniform-record law $p_1(s)$. Writing $r_e(s) = p_e(s)/p_1(s)$, the posterior probability that at least one choice used the posterior rule is

$$\alpha_e(s) = 1 - \frac{e^{k-1}}{r_e(s)}. \tag{D.8}$$

Consequently $m_e = (1 - \alpha_e)m_1 + \alpha_e m_{e,+}$, where $m_{e,+}$ conditions on at least one posterior-rule choice. This proves $\mathrm{TV}(m_e, m_1) \leq \alpha_e$. Under the source's own record law, its expectation is at most $1 - e^{k-1}$; another target law can emphasize records with large conditional weights. Saved likelihood ratios and predictive arrays satisfy the TV bound and the nonnegative mixture residual within 10^{-10} on every checked record. These are latent rule choices: a posterior draw can happen to equal a uniform draw, and the chosen rule is unobserved.

The episode probability of at least one such choice is .00125254. On posterior-sampling target records, the mean posterior probability at prediction steps 2 through 8 is .000179, .00403, .0512, .291, .619, .840 and .943. On uniform target records, the corresponding final value is .000834. Thus actions that are unlikely under uniform querying make the small adaptive component dominate the conditional interpretation of a record.

Target policy	Reward with m_1	Reward with m_e	Paired change	Mean predictive TV
Posterior sampling	8.65252	3.84714	$-4.80539 \pm .07623$	$.122692 \pm .001090$
Uniform	5.85152	5.85170	$.000185 \pm .000256$	$.0000752 \pm .0000157$

Entries after $\pm$ are one standard error across 2,048 independent seed groups. Each score integrates the final report, as in Appendix D.9. Recomputing the uniform score directly agrees with subtracting the masked-score change within 10^{-10}. The last three prediction steps contribute -1.27947, -1.52178 and -1.40930 nats on posterior-sampling trajectories. Separate halves give changes -4.729 and -4.882 nats. The largest 1% of trajectories by absolute change account for 2.39% of the absolute total. The tiny change on uniform trajectories is much more concentrated: the largest 1% account for 51.4%, and the two halves have opposite signs. The large ranking reversal and the small endpoint effects therefore have different statistical resolutions.

For the source-fitting bank, proposals are $\{0, .025, .1, .25, .5, .75, 1\}$, with one trajectory per proposal in each common-seed group. For the frozen source, masked importance ratios have maximum 6.113 across all 28 pairs. Defining weight ESS as $(\sum_b w_b)^2 / \sum_b w_b^2$ after averaging the seven proposal weights within each group gives at least 764.7 of 1,024 groups for every pair. The fresh full-record response calculation has maximum ratio 1.967 and weight ESS $1,920.1$ of 2,048 groups. Weight ESS diagnoses concentration; it is not a confidence bound for a fitted optimum. The directional response SEs use the paired group contributions, retaining their covariance across proposals.

E Rare-source amplification and finite-data recovery

E.1 Independent target evaluation

At fixed log odds -19.7072542334, we generate 1,024 new independent eight-query $\epsilon = 0$ trajectories using seed 2026092501. Each supplies all 28 eligible mask/query records. We define the target metric by first drawing a trajectory, then choosing one of its 28 pairs uniformly. The mean TV across a trajectory's pairs lies in $[0, 1]$. We average by trajectory to account for dependence among masks.

Mean TV is .15314044 (SE .00187082). The rare component's mean posterior weight is .59604034, and its weight exceeds one half on .59629604 of records. The total reward change from the random-source predictor to the mixture is -5.74227321 nats (SE .11989916). The amplification plot in Appendix D.5 averages eligible masks within each prediction step. Four fresh CPU checks cover

two trajectories, two mask/query locations and both sources, with maximum predictive discrepancy 1.38×10^{-13} and log-density-ratio discrepancy 9.06×10^{-14}.

E.2 Two-source lower bound

The source is either P_{base} or $P_w = (1-w)P_{\text{base}} + wP_r$. The estimator knows the candidate problem but not which law generated its n independent episodes. Its respective targets are observational conditionals m_{base} and m_w. Evaluation uses a common fixed retained-record law Q. The estimator may use internal randomness or side information that is identical under the two hypotheses.

Define the separation between the two target predictors by

$$D = \mathbb{E}_{S\sim Q}\,\mathrm{TV}(m_{\text{base}}(\cdot \mid S), m_w(\cdot \mid S)).$$

Proposition 4 (finite-source recovery). Every estimator $\widehat{m}$ in this information setting satisfies

$$\max_{i\in\{\text{base},w\}} \mathbb{E}_{\mathcal{D}\sim P_i^n}\mathbb{E}_{S\sim Q}\,\mathrm{TV}(\widehat{m}(\mathcal{D};S), m_i(S)) \geq \frac{(1-nw)_+ D}{2}. \tag{E.1}$$

Couple training datasets by first sampling the P_{base} episodes, then replacing each independently by a P_r episode with probability w for the mixture dataset. They coincide with probability at least $(1-w)^n \geq (1-nw)_+$. Couple estimator randomness identically. On coincident datasets its outputs coincide; the triangle inequality, averaged over Q, gives a sum of target losses at least D. Averaging over the coupling and dividing by two proves (E.1). Providing the actual source identity or its exact weights changes this information setting.

For $N_Q = 1,024$ target trajectories and $\delta = .05$, bounded-variable concentration gives

$$D \geq \widehat{D} - \sqrt{\frac{\log(1/\delta)}{2N_Q}} = .11489440 \tag{E.2}$$

with probability at least .95 over target validation data, conditional on the fixed weight. Equation (E.1) then gives .05744590 at $n = 8,192$, .05728852 at one million and .05586042 at ten million source episodes. These quantify target recovery when a nearly unobserved source component changes the conditional predictor on evaluation records.

E.3 Recovering a policy with small reward regret

Use the same two possible training sources as in Proposition 4. Fix a finite candidate class Π and let $r_i(\pi) = R(\pi, m_i)$, $r_i^* = \max_{\pi\in\Pi} r_i(\pi)$, for $i \in \{\text{base}, w\}$. The regret of selecting π under source i is $L_i(\pi) = r_i^* - r_i(\pi)$. Define the separation

$$\Gamma = \min_{\pi\in\Pi}\big[L_{\text{base}}(\pi) + L_w(\pi)\big] = r^*_{\text{base}} + r^*_w - \max_{\pi\in\Pi}\big[r_{\text{base}}(\pi) + r_w(\pi)\big]. \tag{E.3}$$

For any procedure that uses n source episodes to select a policy from this class,

$$\max_{i\in\{\text{base},w\}} \mathbb{E}_{\mathcal{D}\sim P_i^n} L_i(\widehat{\pi}(\mathcal{D})) \geq \frac{(1-nw)_+\Gamma}{2}. \tag{E.4}$$

On the coincident datasets in the coupling, the procedure makes the same choice under both sources; couple any internal randomization identically. The sum of its two regrets is at least Γ. On all other datasets regrets are nonnegative. Averaging and taking the larger source risk proves (E.4). For a finite class, $\Gamma > 0$ precisely when the two sources have no common optimal candidate. The result also covers episode mixtures of the candidates: both returns are linear, so maximizing their sum over the convex hull reduces to a vertex.

The full-bank estimates in the source-path figure in Appendix D.5 give $\widehat{\Gamma} = 1.2481$ nats. Using the first-half weight and second-half rewards gives 1.1120, with paired bootstrap interval $[.9018, 1.3399]$. These estimate the population separation in (E.3); they are not finite-sample certificates for a numerical regret bound. Equation (E.4) concerns regret relative to the reward associated with the actual source, rather than regret in acquired information.

E.4 Finite support and target distribution shift

For one mask/query record law, write $r = p_r/p_{\text{base}}$ and $\delta(s) = \mathrm{TV}(m_r(\cdot \mid s), m_{\text{base}}(\cdot \mid s))$. The mixture identity gives $\mathrm{TV}(m_w, m_{\text{base}}) = \alpha_w \delta$. If r is finite Q-almost surely, dominated convergence implies that its target expectation tends to zero as $w \to 0$. This includes every fixed finite common-support model.

If additionally $dQ/dP_{\text{base}} \le M$, the target separation satisfies $D \le Mw$. Indeed, $\delta \le 1$ and concavity of $x \mapsto wx/(1 - w + wx)$ give $\mathbb{E}_{P_{\text{base}}} \alpha_w \le w$, since $\mathbb{E}_{P_{\text{base}}} r \le 1$. A nonvanishing effect at vanishing weights consequently requires changing record distributions without a uniform such bound, or a failure of absolute continuity.

In Mastermind, each complete-history action likelihood ratio for source .5 versus the random source is $.5 + .5Nb_t(a)$. With t reports and the uniform prior, $Nb_t(a) \le \min(N, 141^t)$, since the largest report likelihood ratio is 141. Marginalizing the hidden report preserves the product bound, so at prediction step k,

$$\log r(S_{jk}) \le B_k = \sum_{t=1}^{k-1} \log[.5 + .5 \min(N, 141^t)]. \tag{E.5}$$

Thus $\alpha_w \le \mathrm{sigmoid}(\mathrm{logit}\, w + B_k)$. All retained records satisfy (E.5). The bound $B_8 = 72.713$ permits strong but finite amplification; it is too loose to predict the fitted log odds -19.707. The budget comparison in Appendix D measures that scale directly.

F Neural intervention and reward attribution

F.1 Predictor and training

The causal Transformer has six blocks, width 256, eight attention heads, feed-forward width 1,024 and 4,772,899 parameters. Seven concatenated six-way one-hot vectors represent each guess and are projected to the model width. Each step contributes a query token and a guess/report token with learned type and position embeddings. A dedicated report embedding marks deletion. Predictions are read from query tokens, exposing preceding reports and the current query through causal attention.

We pretrain the shared model for 10,000 AdamW updates, with batch size 256, on generated random-query trajectories. Training masks one of the first seven reports with probability one half and uses observed-report cross entropy; the seed is 20260914. We initialize the shared full predictor from this checkpoint and adapt it by maximizing the likelihood of reports given their full histories. We use 8,192 trajectories generated by $\pi_{.1}$ and fix the predictor at update 5,000 for all teacher arms and adaptation runs.

The teacher intervention uses 8,192 separate training trajectories and 512 validation trajectories, all generated by $\pi_{.1}$ with seed 2026092201. A deletion location is sampled uniformly from the first seven reports. Exact observational and interventional soft labels are computed for every following prediction. Each arm minimizes cross entropy to its teacher over eligible queries. The two arms share initialization, batch indices and symmetry augmentation; only the teacher differs.

Both arms use AdamW with learning rate .0003, weight decay .01, batch size 256 and gradient-norm clipping at 1 for 5,000 updates. Forward passes use bfloat16 autocasting, with float32 log-softmax and loss; references are float64. Each batch independently permutes colors and positions. We evaluate checkpoints at updates $0, 1,000, 2,500, 5,000$ and select the one with the lowest validation KL divergence from its training teacher. Both arms select update 1,000. The primary adaptation seed

Table 1: Selected policy's ϵ for the primary adaptation run. Selected checkpoints minimize validation KL divergence from each model's training teacher; both are at update 1,000. Final checkpoints are fixed at update 5,000. Each cell reports the candidate with the highest estimated reward among the same four policies. The endpoint reward curves below use the selected checkpoints; Appendix H includes all adaptation runs.

Masked teacher	Selected, learned numerator	Selected, exact numerator	Final, learned numerator	Final, exact numerator
Observational	.5	1	.5	1
Interventional	0	1	.1	1

is 2026092202; two further runs use seeds 2026092204 and 2026092205, keeping the pretrained parameters and data fixed.

Teacher	Selected update	KL to observational target	KL to interventional target
Observational	1,000	.374031	.379543
Interventional	1,000	.470533	.310068

These values average KL divergence over eligible queries. The reward decomposition below sums all 28 mask/query terms.

F.2 Selection and uncertainty

We evaluate each of the four candidate policies on 2,048 paired trajectories and select the one with the greatest estimated reward. We estimate acquired information on 8,192 separate paired trajectories. For the primary validation-selected checkpoints, the observational-teacher model selects $.5$ and the interventional-teacher model selects 0. The latter policy acquires $.226189$ more nats of information (conditional SE $.010125$). Jointly bootstrapping policy selection and the information evaluation gives $[.179425, .599137]$, conditional on this pretrained model and adaptation run. For the interventional-teacher model, bootstrapping the reward evaluation selects 0 with frequency $.617$ and $.1$ with $.383$.

At the final checkpoints, the observational- and interventional-teacher models select $.5$ and $.1$, respectively. The latter policy acquires $.193379$ more nats, with joint interval $[.173176, .214267]$. Both additional runs select the same pair under both checkpoint rules. Replacing the full predictor with the exact one makes all 12 combinations of teacher, checkpoint and adaptation run select 1. Appendix H provides these choices and the rewards for the primary run.

F.3 Quantitative attribution

Across targets $0, .1, .5, 1$, A_π is 8.984265, 10.229576, 14.353404, 18.477908 nats. Changing the selected teacher from observational to interventional changes B_π by 3.101533, 2.802759, $.812644$, -2.718986. Measured reward changes are 3.277090, 2.939396, $.872785$, -2.614770. The respective finite-sample differences are $.175557$, $.136637$, $.060141$, $.104216$, with trajectory-level SE $.134596$, $.135977$, $.116931$, $.084818$. All four lie within 1.31 estimated standard errors of zero. The scoring-identity residuals for all four combinations of teacher and checkpoint are all within 1.13 estimated standard errors of zero.

Thus the masked teacher redistributes the positive discrepancy term, while replacing the full predictor removes a policy-dependent negative term. Full-predictor error is greater on random trajectories, so removing it raises the random candidate's reward more.

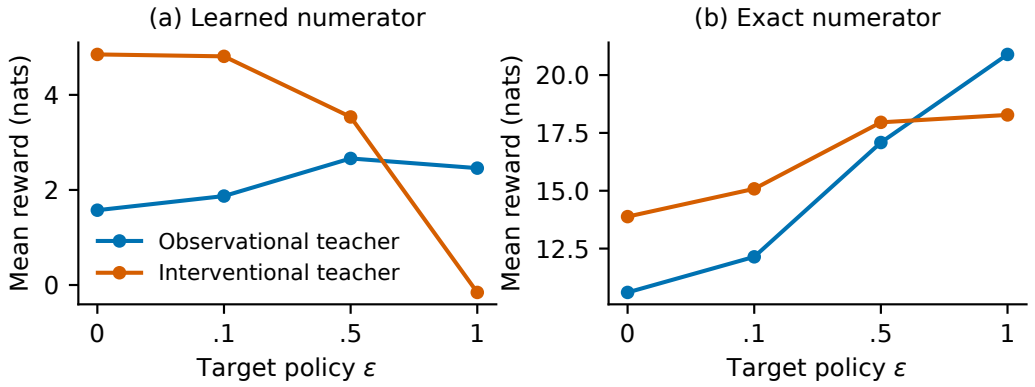


Figure 10: Exact full predictions change the preferred policy for both learned masked models. Curves use the primary validation-selected checkpoints and the same 2,048 paired trajectories per candidate. The left panel uses the shared learned full predictor; the right replaces it with the exact predictor.

Table 2: Policy selected by each combination of full and masked predictors. For the learned masked predictors, we use the primary run's checkpoints selected on the validation set. The entries are the candidate's ϵ.

Masked predictor	Learned full predictor	Exact full predictor
Exact observational teacher	0	1
Exact interventional teacher	0	0
Learned observational model	.5	1
Learned interventional model	0	1

F.4 Exact teachers and effects of approximation

The fixed observational teacher is $m_{.1}$ on every target policy's records; it is not replaced by that target policy's own conditional. The interventional teacher is m_{do}. We recover both from the stored source-conditioned predictive arrays on the neural evaluation trajectories. The common full log numerator agrees with the independent reference file within 1.10×10^{-11}. Every comparison below uses the same realized outcomes in all 28 log-score terms. Appendix H reports complete rewards. the source-path figure in Appendix D.5 instead integrates the final outcome, as specified in Appendix D.3; the two estimators have the same expectation.

With exact predictors in both positions, changing the masked target changes the choice from 1 to 0. For either learned masked model, replacing the full predictor changes its choice to 1 (the main-text crossing table). Holding the learned full predictor fixed, replacing the exact observational teacher by its neural approximation changes the choice from 0 to .5. Each comparison holds the other reward component fixed.

For a teacher m_τ and its learned approximation M_τ, an identity on every trajectory separates three signed effects:

$$\begin{aligned} G(F, M_\tau) = G(f, m_{\mathrm{do}}) &+ \sum_{j<k} \log \frac{m_{\mathrm{do}}(Y_k \mid S_{jk})}{m_\tau(Y_k \mid S_{jk})} \\ &+ \sum_{j<k} \log \frac{m_\tau(Y_k \mid S_{jk})}{M_\tau(Y_k \mid S_{jk})} - \sum_{j<k} \log \frac{f(Y_k \mid H_k)}{F(Y_k \mid H_k)}. \end{aligned} \tag{F.1}$$

The first added term changes the target definition, the second replaces that target by its neural approximation, and the last replaces the full predictor. The identity reconstructs saved rewards within

2×10^{-10}. The approximation term is a signed log-score difference; under a different target policy it need not equal KL to the training teacher. Appendix H gives its value separately from the target and full-predictor terms.

F.5 Checkpoint paths and candidate margins

We evaluate all retained primary masked checkpoints at updates $0, 1,000, 2,500, 5,000$, and all full-only checkpoints at $0, 2,000, 5,000, 10,000, 20,000$. The evaluation uses the same four policy banks and no further training. Per-mask/query log scores and KL values permit every full/masked combination. The previously reported selected and final rewards are reproduced exactly. These diagnostics do not select a new checkpoint.

With the observational masked model fixed at update 1,000, the full checkpoints select $1, 1, .5, .5, 0$ in that order. Bootstrap frequencies for the .5 winner at full updates 5,000 and 10,000 are 88.7% and 92.8%; at update 2,000, 1 wins 99.8%. With the interventional masked model fixed at 1,000, the middle checkpoints favor the $0/.1$ group with uncertain internal ordering. All 40 full/masked combinations and error values are retained; Appendix H tabulates the primary axes.

For a fixed masked model, let $P_\pi = \sum_{j<k} \log[f(Y_k \mid H_k)/F(Y_k \mid H_k)]$ on a trajectory. The .5-minus-1 reward gap equals the exact-full gap plus the mean of $P_1 - P_{.5}$. Between full updates 2,000 and 5,000, this penalty contrast changes by .6682 nats (paired SE .1313). Its contributions at prediction steps two through eight are .0014, $-.0004$, .0273, .0481, .1612, .1171 and .3134 nats. Each step includes all earlier deletion masks, so a full-prediction error at step k is counted $k - 1$ times.

Full KL on the training-source policy .1 decreases from 10.3489 to 10.2296 between those checkpoints, while full KL on the random target increases from 17.5787 to 18.4779. At update 20,000, full KL is 41.4399, 44.6465, 57.5161 and 69.9151 nats on the four targets. The final change of preferred policy therefore occurs with greater full-prediction error, not with a monotonic approach to the exact predictor.

For masked training with the full predictor fixed at 5,000, the exact observational teacher gives a .5-minus-0 gap of -3.6811 nats. The signed approximation contribution in (F.1) changes from -18.1020 at update 0 to 4.7700 at update 1,000. Their sum changes the reward gap from -21.7831 to 1.0889. The largest contributions occur at steps seven and eight: 6.9413 and 10.6396 of the 22.8720-nat change. This comparison holds the target definition and full predictor fixed; it measures how fitting that target redistributes reward across candidates.

All uncertainty here concerns evaluation trajectories conditional on the saved models. Information differences use the same independent utility bank as the earlier intervention. For the .5 and 1 policies, the difference is .3686 nats; this is the consequence of choosing those candidates, not an additional utility replication.

G Executable sources and reproduction

Paths are relative to the supplementary code archive. Paths under `results/`, `data/` and `models/` identify locally generated or downloaded files, not bundled datasets or checkpoints. Numerical analysis uses Python, NumPy and SciPy; neural training and vectorized Mastermind references use PyTorch with CUDA. Experiments used one Ryzen 5800H workstation with 32 GB RAM and an RTX 3070 Laptop GPU with 8 GB memory. Appendix F distinguishes float64 references from neural mixed-precision forward passes.

The archive contains the generation, training, analysis and audit scripts, including their local imports and protocol files. `DATASETS.md` links to the original sensor datasets and gives their expected locations. Synthetic inputs and model checkpoints are generated locally. Fresh runs should use a clean extraction because several scripts protect existing output directories.

The execution order and required inputs are specified in `paper/predictive_curiosity/REPRODUCE.md` and `workflows.json`. The depth-analysis audit checks input hashes, reconstructs all 40 neural combinations, and independently verifies the entropy bound. It writes

`results/predictive_curiosity_depth_summary/audit.json` after the preceding stages have completed.

The package root provides the following entry points:

Build stage	Python entry point
Package integrity	`check_package.py`
Experiment sequences	`reproduce.py`
Figure and table regeneration	`reproduce.pyfigures--run`
Optional Markdown-to-PDF build	`paper/predictive_curiosity/build.py`

Figure and table regeneration requires outputs from the corresponding experiment stages. The supplied LaTeX and vector figures can instead be compiled directly without running experiments. The reproduction guide distinguishes package checks and small exact calculations from the longer GPU and sensor-data stages.

H COMPLETE NUMERICAL TABLES

All rewards and information values are in nats. Tables are generated from retained analysis files. For the coordinate and shared-structure grids, minimum and maximum information among tied reward-optimal actions coincide to the displayed precision. In the Mastermind tables, target labels give the candidate policy's ϵ.

H.1 INDEPENDENT COORDINATES: FLIP PROBABILITY 0

Adaptive and fixed-count reward optima are evaluated against the same information optimum.

Budget	Adaptive reward	Adaptive info.	Fixed reward	Fixed info.	Info. optimum
2	0.6931	0.6931	0.6931	0.6931	1.3863
4	1.3863	1.3863	1.3863	1.3863	2.7726
6	2.0794	2.0794	2.0794	2.0794	4.1589
8	2.7726	2.7726	2.7726	2.7726	5.5452
10	3.4657	3.4657	3.4657	3.4657	6.9315
12	4.1589	4.1589	4.1589	4.1589	8.3178
16	5.5452	5.5452	5.5452	5.5452	11.0904
20	6.9315	6.9315	6.9315	6.9315	13.8629

H.2 INDEPENDENT COORDINATES: FLIP PROBABILITY 0.05

Adaptive and fixed-count reward optima are evaluated against the same information optimum.

Budget	Adaptive reward	Adaptive info.	Fixed reward	Fixed info.	Info. optimum
2	0.3792	0.6101	0.3792	0.6101	0.9893
4	0.8972	1.2201	0.7584	1.2201	1.9785
6	1.4284	1.8302	1.1376	1.8302	2.9678
8	1.9609	2.4403	1.5168	2.4403	3.9571
10	2.4935	3.0504	1.8960	3.0504	4.9463
12	3.0261	3.6604	2.2751	3.6604	5.9356
16	4.0913	4.8806	3.0335	4.8806	7.9141
20	5.1565	6.1007	3.7919	6.1007	9.8926

H.3 INDEPENDENT COORDINATES: FLIP PROBABILITY 0.1

Adaptive and fixed-count reward optima are evaluated against the same information optimum.

Budget	Adaptive reward	Adaptive info.	Fixed reward	Fixed info.	Info. optimum
2	0.2218	0.5144	0.2218	0.5144	0.7361
4	0.5822	1.0287	0.4740	0.6390	1.4723
6	0.9675	1.5431	0.6958	1.1534	2.2084
8	1.3574	2.0575	0.9480	1.2781	2.9445
10	1.7481	2.5696	1.1698	1.7925	3.6806
12	2.1395	3.0681	1.4220	1.9171	4.4168
16	2.9231	4.0659	1.8960	2.5562	5.8890
20	3.7073	5.0646	2.3701	3.1952	7.3613

H.4 Independent coordinates: flip probability 0.2

Adaptive and fixed-count reward optima are evaluated against the same information optimum.

Budget	Adaptive reward	Adaptive info.	Fixed reward	Fixed info.	Info. optimum
2	0.0663	0.3192	0.0663	0.3192	0.3855
4	0.2100	0.4788	0.2100	0.4788	0.7710
6	0.3935	0.7630	0.3402	0.5673	1.1565
8	0.5910	1.0403	0.4426	0.6182	1.5420
10	0.7943	1.2843	0.5529	1.0589	1.9274
12	1.0020	1.5314	0.6803	1.1346	2.3129
16	1.4221	2.0405	0.8931	1.6262	3.0839
20	1.8442	2.5388	1.1278	1.7597	3.8549

H.5 Independent coordinates: flip probability 0.4

Adaptive and fixed-count reward optima are evaluated against the same information optimum.

Budget	Adaptive reward	Adaptive info.	Fixed reward	Fixed info.	Info. optimum
2	0.0008	0.0395	0.0008	0.0395	0.0403
4	0.0043	0.0760	0.0043	0.0760	0.0805
6	0.0100	0.1099	0.0100	0.1099	0.1208
8	0.0173	0.1415	0.0173	0.1415	0.1611
10	0.0259	0.1737	0.0258	0.1710	0.2014
12	0.0358	0.2081	0.0355	0.1987	0.2416
16	0.0587	0.2715	0.0571	0.2489	0.3222
20	0.0853	0.3308	0.0811	0.2934	0.4027

H.6 Independent coordinates: flip probability 0.5

Adaptive and fixed-count reward optima are evaluated against the same information optimum.

Budget	Adaptive reward	Adaptive info.	Fixed reward	Fixed info.	Info. optimum
4	0.0000	0.0000	0.0000	0.0000	0.0000

H.7 Shared structure at budget four

Noise denotes flip probability q; prior mass denotes ρ in (8).

Noise	Prior mass	Reward	Acquired info.	Info. optimum	Fraction
0.0	0	1.3863	1.3863	2.7726	0.5000
0.0	0.25	1.3547	1.3547	2.6136	0.5183
0.0	0.5	1.2555	1.2555	2.2298	0.5630
0.0	0.75	1.0880	1.2951	1.6441	0.7877
0.0	0.9	0.9233	1.0073	1.1601	0.8682
0.0	1.0	0.6931	0.6931	0.6931	1.0000
0.1	0	0.5822	1.0287	1.4723	0.6988
0.1	0.25	0.5713	1.0095	1.4061	0.7180
0.1	0.5	0.5392	0.9494	1.2393	0.7661
0.1	0.75	0.4909	0.8387	1.0094	0.8309
0.1	0.9	0.4740	0.6390	0.8138	0.7853
0.1	1.0	0.4740	0.6390	0.6390	1.0000

H.8 SEARCH GRID: 3 CANDIDATES

Noise is uniform-replacement probability (twice the flip probability); floor denotes δ. Responses are listed in source order SS, SR, RS, RR. Fitted information is acquired by the policy that maximizes the self-fitted reward, found using multiple initializations; the benchmark maximizes information over the same rectangle.

Noise	Floor	Responses	Fitted info.	Info. optimum
0.05	0.01	RR / RS / SR / RR	0.6037	0.9487
0.05	0.05	RR / RS / SR / RR	0.6169	0.9337
0.05	0.1	RR / RS / SR / RR	0.6334	0.9150
0.2	0.01	RR / RS / SR / RR	0.4717	0.6476
0.2	0.05	RR / RS / SR / RR	0.4774	0.6390
0.2	0.1	RR / RS / SR / RR	0.4846	0.6282
0.5	0.01	RR / RS / SR / RR	0.2077	0.2436
0.5	0.05	RR / RS / SR / RR	0.2085	0.2414
0.5	0.1	RR / RS / SR / RR	0.2095	0.2387

H.9 SEARCH GRID: 4 CANDIDATES

Noise is uniform-replacement probability (twice the flip probability); floor denotes δ. Responses are listed in source order SS, SR, RS, RR. Fitted information is acquired by the policy that maximizes the self-fitted reward, found using multiple initializations; the benchmark maximizes information over the same rectangle.

Noise	Floor	Responses	Fitted info.	Info. optimum
0.05	0.01	RR / RS / SR / RR	0.5334	0.8913
0.05	0.05	RR / RS / SR / RR	0.5468	0.8754
0.05	0.1	RR / RS / SR / RR	0.5635	0.8556
0.2	0.01	RR / RS / SR / RR	0.4139	0.5978
0.2	0.05	RR / RS / SR / RR	0.4195	0.5884
0.2	0.1	RR / RS / SR / RR	0.4265	0.5766
0.5	0.01	RR / RS / SR / RR	0.1788	0.2169
0.5	0.05	RR / RS / SR / RR	0.1795	0.2145
0.5	0.1	RR / RS / SR / RR	0.1804	0.2114

H.10 SEARCH GRID: 8 CANDIDATES

Noise is uniform-replacement probability (twice the flip probability); floor denotes δ. Responses are listed in source order SS, SR, RS, RR. Fitted information is acquired by the policy that maximizes the self-fitted reward, found using multiple initializations; the benchmark maximizes information over the same rectangle.

Noise	Floor	Responses	Fitted info.	Info. optimum
0.05	0.01	RR / RS / SR / RR	0.3564	0.6171
0.05	0.05	RR / RR / SR / RR	0.3655	0.6049
0.05	0.1	RR / RR / SR / RR	0.3768	0.5896
0.2	0.01	RR / RS / SR / RR	0.2700	0.3998
0.2	0.05	RR / RS / SR / RR	0.2732	0.3924
0.2	0.1	RR / RR / SR / RR	0.2772	0.3831
0.5	0.01	RR / RS / SR / RR	0.1097	0.1363
0.5	0.05	RR / RS / SR / RR	0.1100	0.1344
0.5	0.1	RR / RR / SR / RR	0.1104	0.1321

H.11 Neural rewards: primary adaptation run

O/I indicate observational/interventional teachers; selected/final specify the checkpoint rule. Learned numerator.

Teacher / rule	Target 0	Target .1	Target .5	Target 1
O / selected	1.5738	1.8713	2.6627	2.4587
O / final	3.9057	4.4998	5.7280	4.8185
I / selected	4.8509	4.8107	3.5355	-0.1561
I / final	5.5366	5.8340	5.2970	2.1831

Exact numerator.

Teacher / rule	Target 0	Target .1	Target .5	Target 1
O / selected	10.6043	12.1430	17.0841	20.8923
O / final	12.9362	14.7715	20.1494	23.2521
I / selected	13.8814	15.0824	17.9569	18.2776
I / final	14.5671	16.1058	19.7183	20.6168

H.12 Choices across adaptation runs

Each cell lists the selected policies for the observational/interventional teachers. Runs share pretrained parameters and data; learned/exact refers to the full predictor.

Adaptation seed	Selected, learned	Final, learned	Selected, exact	Final, exact
2026092202	0.5 / 0	0.5 / 0.1	1 / 1	1 / 1
2026092204	0.5 / 0.1	0.5 / 0.1	1 / 1	1 / 1
2026092205	0.5 / 0.1	0.5 / 0.1	1 / 1	1 / 1

H.13 Source comparison: full bank

The weight is fitted and rewards are evaluated on all 2,048 trajectories per candidate. Parentheses give trajectory-level standard errors.

Target	Random source	Fitted mixture
0	8.7295 (0.0968)	2.8359 (0.0710)
0.1	8.5107 (0.0979)	3.4890 (0.0795)
0.5	7.4814 (0.0972)	5.9340 (0.0938)
1	5.9340 (0.0867)	5.9340 (0.0867)

H.14 Source comparison: held half

The weight is fitted on the first 1,024 trajectories per candidate; rewards are evaluated on the remaining 1,024. Parentheses give trajectory-level standard errors.

Target	Random source	Fitted mixture
0	8.7198 (0.1386)	2.9762 (0.1019)
0.1	8.5181 (0.1411)	3.6763 (0.1129)
0.5	7.6078 (0.1364)	6.1070 (0.1337)
1	5.9913 (0.1220)	5.9913 (0.1220)

H.15 EXACT MASKED TEACHERS: ALL CANDIDATE REWARDS

The observational teacher uses the conditional distribution induced by the fixed source $\pi_{.1}$ for every target policy. All 28 terms use realized log scores.

Full predictor / teacher	Target 0	Target .1	Target .5	Target 1
Learned / Observational	-7.9365	-8.9437	-11.6176	-11.3999
Learned / Interventional	-0.4441	-1.8851	-6.8920	-12.4319
Exact / Observational	1.0940	1.3280	2.8037	7.0338
Exact / Interventional	8.5864	8.3866	7.5293	6.0018

H.16 SIGNED EFFECTS ON THE NEURAL REWARD

We use the primary run's checkpoints selected on the validation set. Effects of the target definition and masked approximation are signed log-score changes, as defined in Appendix F.4.

Effect	Target 0	Target .1	Target .5	Target 1
Observational target	-7.4925	-7.0586	-4.7256	1.0321
Approx. observational	9.5103	10.8150	14.2803	13.8585
Approx. interventional	5.2950	6.6958	10.4276	12.2758
Full-predictor penalty	9.0305	10.2718	14.4214	18.4337

H.17 SIMILAR SOURCE VALUES AND CANDIDATE RESPONSES

The value column bounds the absolute population difference from the random source using (D.5). Margins and best responses are estimated on the paired trajectory bank. Response gain is the greatest candidate reward minus the source's own reward. Information uses the existing independent utility trajectories.

Source weight	Value bound	Chosen epsilon	Reward margin	Response gain	Information
0	0	0	0.2189	2.7955	11.8586
4.25e-18	5.3e-15	0.1	0.1426	1.6332	11.8258
9.36e-14	9.05e-11	0.5	1.1699	1.1699	11.6324
1.52e-08	9.62e-06	1	0.3021	$< 10^{-8}$	11.2638

H.18 FULL-PREDICTOR UPDATES: FIXED OBSERVATIONAL MASKED MODEL

The masked model is fixed at update 1,000. KL sums the 28 mask/query terms. The margin is reward on .5 minus reward on 1, not necessarily the winner's margin.

Full update	Chosen epsilon	KL on .1	KL on 1	Margin (SE)
0	1	42.2823	14.5216	-18.0826 (0.5452)
2000	1	10.3489	17.5787	-0.4641 (0.1523)
5000	0.5	10.2296	18.4779	0.2041 (0.1740)
10000	0.5	11.3463	19.3659	0.3765 (0.2498)
20000	0	44.6465	69.9151	10.1044 (0.9605)

H.19 Full-predictor updates: fixed interventional masked model

The masked model is fixed at update 1,000. KL sums the 28 mask/query terms. The margin is reward on 0 minus reward on .1, not necessarily the winner's margin.

Full update	Chosen epsilon	KL on .1	KL on 1	Margin (SE)
0	1	42.2823	14.5216	-4.9309 (0.3416)
2000	0.1	10.3489	17.5787	-0.0398 (0.1124)
5000	0	10.2296	18.4779	0.0402 (0.1331)
10000	0	11.3463	19.3659	0.0053 (0.1673)
20000	0	44.6465	69.9151	2.0092 (0.5576)

H.20 Masked-predictor updates: observational teacher

The full model is fixed at update 5,000. KL is measured relative to the fixed training teacher on each target policy. The margin is reward on .5 minus reward on 0.

Masked update	Chosen epsilon	Teacher KL on .1	Teacher KL on 1	Margin (SE)
0	0	41.6569	10.2354	-21.7831 (0.6631)
1000	0.5	10.7842	15.0681	1.0889 (0.1681)
2500	0.5	12.9101	17.2164	1.9309 (0.2115)
5000	0.5	13.4017	17.4737	1.8223 (0.2106)

H.21 Masked-predictor updates: interventional teacher

The full model is fixed at update 5,000. KL is measured relative to the fixed training teacher on each target policy. The margin is reward on .5 minus reward on 0.

Masked update	Chosen epsilon	Teacher KL on .1	Teacher KL on 1	Margin (SE)
0	0	31.0510	9.7881	-21.7831 (0.6631)
1000	0	9.1711	12.2446	-1.3154 (0.2125)
2500	0.1	10.5065	14.1390	-0.4097 (0.2171)
5000	0.1	10.8396	14.6199	-0.2396 (0.2197)

I SCORING TIMES AND THE JOINT PREDICTIVE LAW

I.1 COMPLETE DELETION AND DUAL TOTAL CORRELATION

Fix a query sequence $a = (a_1, \dots, a_T)$ and its coherent joint predictive law $p_a(y_{1:T})$. All conditionals in this subsection use this law, and the required log expectations are assumed finite. For each j, applying the chain rule to the future predictions gives

$$\sum_{k>j} \log \frac{p_a(y_k \mid y_{<k})}{p_a(y_k \mid y_{<k} \setminus y_j)} = \log \frac{p_a(y_{>j} \mid y_{\le j})}{p_a(y_{>j} \mid y_{<j})} = \log \frac{p_a(y_j \mid y_{-j})}{p_a(y_j \mid y_{<j})}. \tag{I.1}$$

The last equality is Bayes' rule. Summing j, including the zero term at $j = T$, yields

$$G(a, y) = \sum_{j=1}^{T} \log p_a(y_j \mid y_{-j}) - \log p_a(y_{1:T}). \tag{I.2}$$

Its expectation is $H(Y_{1:T}) - \sum_j H(Y_j \mid Y_{-j})$, the dual total correlation (Han, 1978; Austin, 2020). Under conditional independence of the observations given θ and a, expand

$$I(\theta; Y_{1:T}) - \sum_j I(\theta; Y_j \mid Y_{-j}).$$

The terms conditional on θ cancel because $H(Y_{1:T} \mid \theta) = \sum_j H(Y_j \mid \theta)$ and $H(Y_j \mid \theta, Y_{-j}) = H(Y_j \mid \theta)$. This proves (3). For continuous observations the same calculation uses differential entropies when they are finite. The formulation through log densities in (I.2) specifies the score directly.

For a zero-mean Gaussian joint law with covariance C, let $Q = C^{-1}$ and $u = Qy$. Its full conditional distributions give

$$G(a, y) = \frac{1}{2} \left[\log\det C + \sum_j \log Q_{jj} + y^\top Q y - \sum_j \frac{u_j^2}{Q_{jj}} \right]. \tag{I.3}$$

Under the fixed-design law, the last two terms both have expectation T. Thus the expected score is $\frac{1}{2}[\log\det C + \sum_j \log Q_{jj}]$. For adaptive acquisition, (I.3) remains a pathwise identity with the realized actions supplied as interventions. The outcomes conditional on the selected sequence generally differ from p_a, so the quadratic terms must be retained. The GP learning experiments use (I.3) in full.

Independent enumeration checks the pathwise identity, its expectation and the latent-information decomposition on 80 discrete distributions. The largest discrepancy is below 6×10^{-15}. Gaussian implementations were also checked against direct prefix and deletion conditionals, including repeated actions and heterogeneous noise. Code and retained checks are `experiments/audit_predictive_curiosity_joint_score.py` and `results/predictive_curiosity_joint_score_20260909/audit.json`; the GP trajectory checks are specified below.

I.2 WEIGHTED ACQUISITION OVER ALL ADAPTIVE POLICIES

Let $d \ge T$ independent fair bits be queried without noise, and let $w_{jk} \ge 0$ be fixed. Keeping observation j changes the prediction at time k precisely when j is the only earlier query of coordinate A_k. Thus j and k must be its first and second query times. The improvement is $\log 2$, proving the pathwise expression in (6).

For different coordinates these first/second pairs use disjoint time points, so they form a matching. Each path's reward is bounded by the maximum weight of a matching times $\log 2$, independently of

how actions depend on past observations. Conversely, assign a distinct coordinate to each edge of any matching and another distinct coordinate to every unmatched time. This fixed schedule realizes its weight and proves attainability over adaptive randomized policies.

If T is even and every $w_{jk} > 0$, a nonperfect matching has at least two unmatched vertices. Joining them strictly increases its weight. Hence every maximum matching is perfect. A policy attaining the expected maximum must attain the pathwise maximum almost surely, and therefore must query exactly $T/2$ coordinates twice. This establishes the information loss for arbitrary strictly positive time weights. When weights can approach zero, this argument gives no uniform quantitative bound for near-optimal policies across all such weight choices; (5) concerns the unweighted objective.

For $w_{jk} = \mathbf{1}\{k = j + H\}$ with integer $H \geq 1$, the positive-weight graph is the union of chains on residues modulo H. A chain of n_r vertices admits $\lfloor n_r/2 \rfloor$ edges. Thus the maximum reward is $m_H \log 2$, where $m_H = \sum_r \lfloor n_r/2 \rfloor$. Any optimizer needs m_H repeat queries, so it observes at most $T - m_H$ distinct coordinates. A maximum matching with fresh unmatched coordinates attains this bound. For fixed H, the information fraction tends to one half as T increases.

The fixed-interval prefix score in NDIGO (Azar et al., 2019) compares predictions before and after observing Y_j, without adding intervening outcomes to either prefix. Here future actions are supplied as interventional query inputs, as in the bit-model oracle above. In this model the score is positive precisely when j first reveals the coordinate queried at $j + H$. Each coordinate produces at most one credited edge, and the credited edges are again disjoint. Any lag-H matching can be realized, giving the same reward optimum and greatest information among its maximizers. This derivation uses the prefix score's own conditioning sets. If prefix improvements are instead summed over every later time, one first observation earns credit at every repeat, giving $(T - D_T) \log 2$ with D_T distinct coordinates; repeating one coordinate then maximizes reward. This last objective is a separate all-horizon variant.

For the Brier score $S(p, y) = -\sum_b (p_b - \mathbf{1}\{b = y\})^2$, a revealed fair bit improves the score by $1/2$. The same pathwise counting and matching proofs apply with $\log 2$ replaced by $1/2$.

I.3 Weighted self-fitting and verification

With log scores and the observational conditional fitted to the collecting policy, proper scoring applies separately to each term:

$$V_w(\pi) = R_w(\pi, m_{\mathrm{do}}) - \sum_{j<k} w_{jk} \mathbb{E}_\pi \mathrm{KL}(m_\pi(\cdot \mid S_{jk}) \| m_{\mathrm{do}}(\cdot \mid S_{jk})). \tag{I.4}$$

The penalty is nonnegative. A fixed maximum-matching schedule makes both masked targets agree and attains the upper bound. Every weighted self-fit maximizer therefore attains the interventional maximum. Under strictly positive weights and even T, each acquires exactly $T/2$ bits. With weights that include zeros, the bound on the greatest information among fixed-interval optima also carries over because its fixed attaining schedule has zero penalty.

`experiments/predictive_curiosity_weighted_rewards.py` enumerates all unlabeled query partitions for $T = 2, \ldots, 8$ and independently compares their scores with graph matching. For $T \leq 6$, it also computes predictive log and Brier scores directly. The retained results are `results/predictive_curiosity_weighted_rewards_20260909/analysis.json`. These checks corroborate the pathwise proof; the adaptive statement follows from that proof rather than finite enumeration.

J CONTINUOUS ACQUISITION AND LEARNED PREDICTORS

J.1 FIELD, MEASUREMENTS AND EVALUATION

The GP acquisition environment follows the public description in (Elmoznino et al., 2026), Section 5.1 and Appendix E.1. We use a GRU policy in place of Griffin, place queries at tile centers, and match the oracle predictor to the finite-grid interpolation used to generate data. Let Z contain the 10×10 grid on $[-10, 10]^2$ and X the 8×8 grid of tile centers. Define

$$k(z, z') = \left(1 + \frac{\|z - z'\|^2}{32}\right)^{-1}, \quad K = k(Z, Z), \quad P = k(X, Z)K^{-1}, \quad A = PKP^\top. \tag{J.1}$$

Draw the latent field $\theta \sim \mathcal{N}(0, K)$. A query at site i returns $Y = (P\theta)_i + \eta$, with independent $\eta \sim \mathcal{N}(0, n_i)$. Checkerboard cells alternate between $n_i = .001$ and 5. Each observation is a new noisy measurement, including repeated queries. For action sequence a, set $C = A_{a,a} + \mathrm{diag}(n_a)$. The exact deletion reward is (I.3); acquired information on that record is $\frac{1}{2}(\log\det C - \sum_t \log n_{a_t})$. Averaging this posterior entropy reduction over adaptive records gives $I(\theta; h_T)$.

The full-grid posterior has mean $\widehat{\theta} = KP_a^\top C^{-1} y$ and covariance $K - KP_a^\top C^{-1} P_a K$. Let v_i be its marginal variances. The task training objective is $-\frac{1}{200}\sum_i[\log(2\pi) + 1 + \log v_i]$, the expected marginal log score. We record mean posterior variance and realized reconstruction error $\|\theta - \widehat{\theta}\|^2/100$ separately. Reported test MSE uses the latter. Latent values are used for sampling observations and evaluating reconstruction; they are not policy inputs.

J.2 POLICY TRAINING AND OPTIMIZATION PROTOCOL

A GRU receives the previous query coordinates divided by 10, the previous observation divided by 3, the time fraction t/T, and an indicator for the first step. Its 128 hidden units feed a 64-action linear actor and a linear value head. Actor weights and biases start at zero. Actions are sampled from its categorical distribution. Each update uses 128 new fields and full episodes. All objectives use Adam with learning rate 3×10^{-4}, gradient norm clipping at 1 and entropy coefficient .005.

Let R_b be terminal return, v_{bt} the value prediction, and $s = \max(\mathrm{sd}_b\, R_b, .01)$ the current batch scale. The implemented loss is

$$\begin{aligned}\mathcal{L} = &-\frac{1}{BT}\sum_{b,t}\log\pi(A_{bt} \mid h_{b,t-1})\frac{R_b - \mathrm{stopgrad}(v_{bt})}{s} \\ &+ \frac{1}{2BT}\sum_{b,t}\left(\frac{v_{bt} - R_b}{s}\right)^2 - \frac{.005}{BT}\sum_{b,t} H(\pi(\cdot \mid h_{b,t-1})).\end{aligned} \tag{J.2}$$

The actor and critic share the GRU. The batch-dependent scale and entropy term are part of the training algorithm. Different reward distributions can change their effective optimization behavior, even with the same hyperparameters. These experiments compare learned policies under a common protocol; global acquisition optima are established only where the theory or exact search supplies them.

The exact-predictor comparison uses seeds 2026091001, 2026091002 and 2026091003, budgets 13 and 26, and 2,000 updates per objective. No intermediate checkpoint is selected by test performance. At each budget and seed, evaluation uses 2,048 new latent fields and a shared measurement-noise bank across policies. Action sampling remains stochastic. Random querying and greedy maximization of the current signal-to-noise variance ratio use the same fields. Curves record one training batch at step 1 and every 200 updates, rather than evaluation on a fixed validation bank. Shading spans the three seeds. The learned-covariance continuations extend to 4,000 total updates.

Exact-predictor policy learning. Entries average three training seeds, each evaluated on 2,048 new fields. MSE measures reconstruction of the latent field. The task policy maximizes a terminal

prediction score; the greedy information policy is an exact-model reference. Per-seed estimates and paired uncertainty follow below.

Budget	Policy objective	Deletion reward	MSE	Distinct sites
13	Deletion	12.405	.5607	4.95
13	Task score	7.207	.3404	10.85
13	Random	2.466	.4916	11.85
13	Greedy information	1.806	.2192	13.00
26	Deletion	24.758	.2925	11.78
26	Task score	20.063	.1974	18.14
26	Random	7.620	.3220	21.50
26	Greedy information	6.583	.0775	26.00

Both learned objectives largely avoid high-noise locations: deletion policies make 99.88% and 99.95% of queries at low-noise sites, compared with 99.15% and 98.70% for task policies. All six seed/budget comparisons favor task training in reconstruction MSE.

The following per-seed table reports independent evaluation. Each row has 2,048 test fields. We give MSE and its conditional Monte Carlo standard error; variation across training seeds remains a separate source of uncertainty.

Budget	Seed suffix	Policy	MSE	MC SE
13	1001	greedy	0.21730	0.00164
13	1001	random	0.48706	0.00470
13	1001	reward	0.54420	0.00505
13	1001	task	0.33567	0.00315
13	1002	greedy	0.22062	0.00163
13	1002	random	0.49508	0.00497
13	1002	reward	0.57004	0.00524
13	1002	task	0.34376	0.00323
13	1003	greedy	0.21958	0.00161
13	1003	random	0.49258	0.00464
13	1003	reward	0.56799	0.00490
13	1003	task	0.34177	0.00320
26	1001	greedy	0.07726	0.00049
26	1001	random	0.32055	0.00303
26	1001	reward	0.28958	0.00255
26	1001	task	0.19826	0.00191
26	1002	greedy	0.07752	0.00051
26	1002	random	0.32326	0.00298
26	1002	reward	0.30513	0.00270
26	1002	task	0.19559	0.00191
26	1003	greedy	0.07780	0.00050
26	1003	random	0.32228	0.00309
26	1003	reward	0.28292	0.00246
26	1003	task	0.19839	0.00183

For paired policy comparisons, each shared field/noise pair is one sampling unit. We compute differences before their standard errors. The first comparison's full results and per-trajectory arrays are in `results/predictive_curiosity_published_gp_20260909/`. All six deletion-versus-task comparisons have higher deletion reward and higher MSE. The exact-model greedy reference has lower MSE than both learned objectives at either budget.

The paired differences below subtract the task policy from the deletion policy. Each standard error describes Monte Carlo uncertainty conditional on the two trained policies, using the same 2,048 fields; it does not measure variation across training seeds.

Budget	Seed suffix	Reward difference	Paired SE	MSE difference	Paired SE
13	1001	5.47479	.08288	.20853	.00473
13	1002	5.37827	.08739	.22627	.00501
13	1003	4.74305	.07773	.22622	.00465
26	1001	4.55824	.11217	.09132	.00266
26	1002	4.66037	.11425	.10953	.00290
26	1003	4.86597	.11266	.08453	.00261

J.3 Switching the objective from the same learned policy

We freeze the three budget-26 task checkpoints from the preceding comparison at update 2,000. Each supplies the identical model parameters to two continuations, one maximizing deletion reward and the other the task score. The field, observation model, GRU inputs and reward computations remain as above. Both branches receive 2,000 further updates of 128 new fields. We specify two continuation algorithms before running them.

The primary algorithm uses actor-only REINFORCE. For each episode, subtract the mean return of the other 127 episodes, then divide by a fixed positive scale. The scale is the objective's return standard deviation on 2,048 separate calibration trajectories from that seed's initial policy, floored at .01. Calibration gives deletion scales 4.17–4.23 and task scales .0739–.0787. The loss is the negative mean action log probability times this normalized advantage, averaged over episodes and query times. Adam starts with fresh state at learning rate 3×10^{-4}; gradient norm clipping remains 1. There is no value loss or entropy term. Thus the continuation removes both the inherited critic target and current-batch normalization from the policy loss.

The secondary algorithm retains (J.2), including its shared critic, batch scale and entropy term. Each branch receives a separate copy of the same source optimizer state. Its critic initially predicts the old task score, including in the branch switched to deletion reward. This comparison tests continuation under the original workflow; the actor-only comparison isolates the switch without that critic mismatch.

All branches begin with the same training random-number state within a seed. Evaluation preserves training RNG state. At update zero and every 200 updates, we evaluate on 512 fixed diagnostic fields per seed, shared across branches. No diagnostic checkpoint is selected for final reporting. The final policies, unchanged initial policy and original deletion policy are evaluated on 4,096 fresh shared fields and noise draws per seed, with matched action-sampling streams. These fields are separate from calibration and diagnostics. The same initial model gives exactly identical diagnostic results in all four branches.

Policy	Deletion reward	MSE	Information	Distinct sites
Initial task	20.0000	.1959	57.5575	18.1696
Actor only, deletion	24.9667	.2692	43.2354	12.1790
Actor only, task	20.2084	.1856	58.7695	18.2738
Original protocol, deletion	24.3607	.3399	37.1868	10.2825
Original protocol, task	20.1801	.1916	58.1552	18.1519
Original deletion policy	24.6728	.2915	41.4562	11.7498
Greedy information	6.5038	.0774	78.8163	26.0000

Entries average the three seeds. MSE uses realized reconstruction error. The ratio of mean deletion/task MSE after continuation is 1.451 for actor-only updates and 1.774 for the original protocol. Within each algorithm, all three switched policies increase deletion reward and MSE relative to both their initial policy and the task continuation. On average, the actor-only switch also improves both reward and reconstruction relative to the original deletion policy. Initialization and optimization therefore affect the final solution, while switching the objective from the common task policy worsens reconstruction under both tested algorithms.

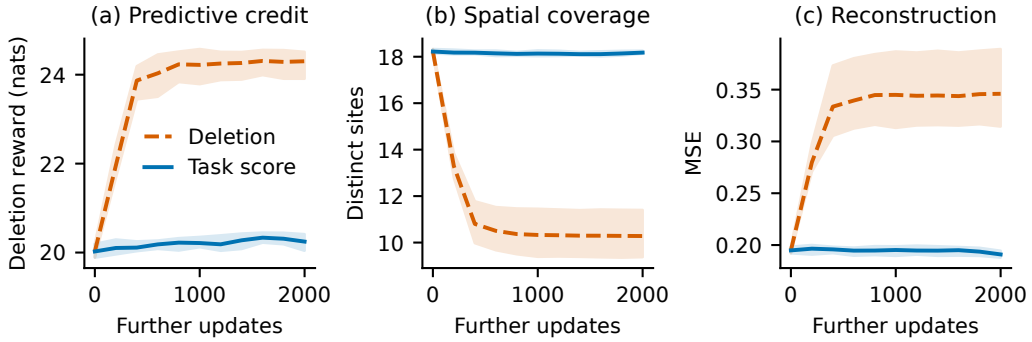


Figure 11: The original actor/critic protocol also trades task performance for deletion reward after the switch. Both branches inherit the same task-policy and optimizer checkpoint. Lines average three seeds on the fixed diagnostic banks and shading spans the seeds. Figure 3 gives the actor-only comparison, which removes the inherited critic target.

The following differences subtract task continuation from deletion continuation. Standard errors are conditional Monte Carlo errors over 4,096 paired fields; the three rows within each algorithm are separate training seeds.

Protocol	Seed suffix	Reward difference	Paired SE	MSE difference	Paired SE
Actor only	1001	4.54824	.07781	.07195	.00163
Actor only	1002	5.08436	.07815	.08374	.00177
Actor only	1003	4.64233	.07933	.09514	.00188
Original	1001	4.37583	.07898	.13697	.00211
Original	1002	4.57839	.07838	.11994	.00215
Original	1003	3.58762	.07617	.18789	.00235

Implementation and independent checks are `experiments/gp_objective_switch.py`, `experiments/check_gp_objective_switch.py` and `experiments/audit_gp_objective_switch.py`. Results are retained in `results/gp_objective_switch_20260909/`. The replay audit checks 24 final trajectories for each of seven policy groups and three seeds, totaling 504 trajectories. Direct NumPy prefix/deletion scores differ by at most 2.60×10^{-11}; realized MSE differs by at most 1.91×10^{-13}. Saved table means and paired standard errors reproduce exactly from the full trajectory arrays. Source checkpoint and prospective protocol hashes are retained with the run.

J.4 Learning a covariance and continuing acquisition

For each seed, collect 8,192 random 26-measurement trajectories. The first 6,144 train a free covariance $\widehat{A} = LL^\top + 10^{-5}I$; the remaining 2,048 select it by Gaussian validation NLL. The diagonal of L is a softplus transform, with free lower-triangular entries. Measurement noise is supplied. Adam uses learning rate .01, batch 128 and gradient clipping at 10 for 1,000 updates; validation is checked every 100 updates. The selected model supplies both full and deleted-record predictions by Gaussian conditioning. NLL omits the common $\frac{1}{2}\log(2\pi)$ term and is normalized by trajectory length.

Initial validation excess NLL is .0073–.0079 nats per observation. Train one policy on its deletion reward and one on its information reward for 2,000 updates. Collect 8,192 new trajectories from each policy, split them 6,144/2,048, and concatenate each split with its corresponding initial random split. This gives equal initial and policy-generated records for refitting and validation. Refit from the selected initial covariance using the same 1,000-update procedure. At the unchanged policy, compare old and refitted predictors on exactly the same fresh records. Then continue policy learning for 2,000 updates under the refitted predictor. A control continues from the identical policy, optimizer and random-number states while retaining the old predictor.

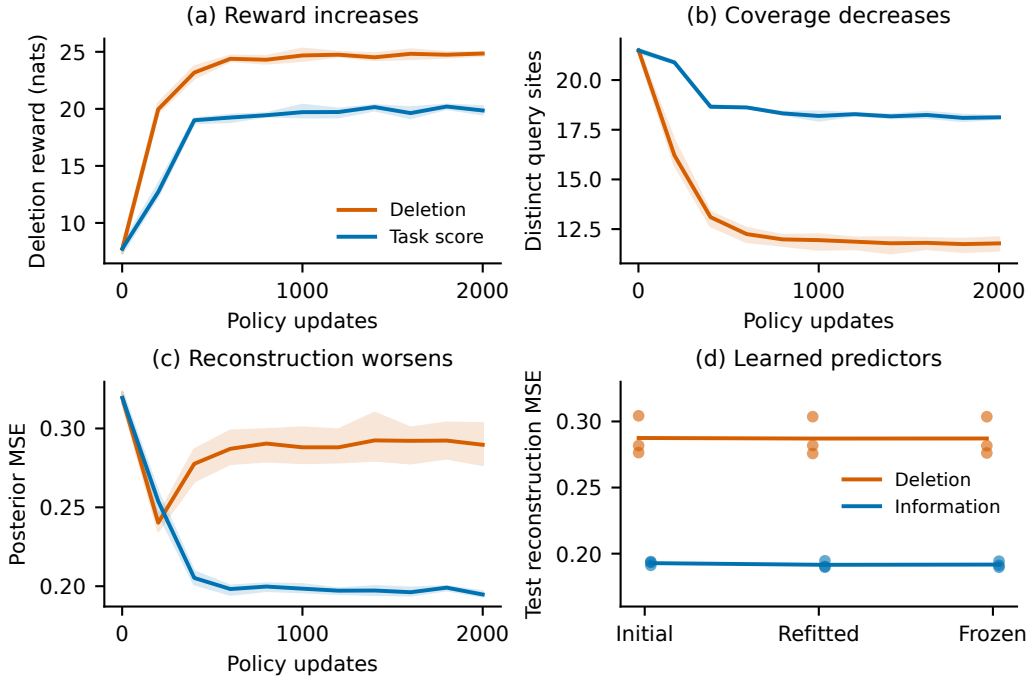


Figure 12: Original GP training and predictor refitting. (a–c) Budget-26 training from the original initialization: lines average three seeds and shading spans their range; MSE is mean posterior variance on each sampled training batch. Between updates 200 and 2,000, deletion reward rises from 19.96 to 24.85, distinct sites fall from 16.20 to 11.78, and MSE rises from .240 to .290. (d) Independent evaluation with learned predictors before and after 2,000 further updates. Points are individual seeds; refitted and frozen branches start from the same policy and optimizer.

The full-grid MSE below evaluates the information collected using the exact model. Learned-site MSE reconstructs the 64 candidate values using the fitted model itself. Their grids differ, so the two columns are different evaluation quantities.

Reward	Stage	Full-grid MSE	Learned-site MSE	Distinct sites
Deletion	Initial	.2875	.2948	11.7565
Deletion	Refit, continue	.2871	.2939	11.7438
Deletion	Frozen, continue	.2871	.2942	11.7459
Information	Initial	.1928	.2129	17.9653
Information	Refit, continue	.1916	.2132	17.9386
Information	Frozen, continue	.1917	.2127	17.9318

Refitting lowers same-record NLL by .0014–.0037 nats per observation on the three deletion-policy banks and .0070–.0106 on the information-policy banks. Improved likelihood does not improve every individual learned-site reconstruction estimate. The aggregate acquisition gap remains after both continuations. The covariance model treats actions as query inputs throughout; observational conditioning on the source's action probabilities is studied separately in Section 5.

J.5 Existing rewards and paid scoring measurements

The information objective uses posterior entropy reduction. The description-length objective used in (Elmoznino et al., 2026) sums the log-predictive improvement at the current outcome after incorporating that outcome into the posterior. In the BED model, it is computed by a second likelihood evaluation at the same value, using the updated parameter posterior. It requires no additional observation in this implementation. Direct Gaussian posterior calculations independently verify the formula used in training.

For each probe policy, acquisition ends after 25 measurements. A 26th site A^* is sampled uniformly from the 64 sites and yields a real independent noisy observation Y^* from the same field. It is unavailable to the acquisition policy. With h the acquisition record and h_{-j} deleting acquired outcome j while retaining actions, define

$$R_{\text{full probe}} = \log \frac{p(Y^* \mid h, A^*)}{p(Y^* \mid A^*)}, \qquad R_{\text{deletion probe}} = \sum_{j=1}^{25} \log \frac{p(Y^* \mid h, A^*)}{p(Y^* \mid h_{-j}, A^*)}. \tag{J.3}$$

Both scores are sampled during training, with the same 2,000-update policy protocol. All final reconstruction metrics incorporate all 26 measurements. Probe training curves record reconstruction before this last observation; final evaluation also saves the before-probe quantities. We evaluate all six objectives on a fresh common bank, reusing the original deletion/task checkpoints. The slight change from the original comparison above reflects different test draws.

Objective	MSE	Marginal log score	Information	Distinct sites
Deletion	.2917	-.6144	41.4890	11.7617
Task score	.1956	-.3191	57.5373	18.1528
Information	.1930	-.3185	57.8568	17.9261
Description length	.1923	-.3207	57.8674	17.9352
Full probe	.2275	-.4142	50.9581	19.8345
Deletion probe	.2846	-.5681	40.6191	21.3660

Rows average three seeds, each with 2,048 new fields. The full probe's mean pre-probe posterior MSE falls from .2725 at update 1,000 to .2342 at 2,000; deletion probe falls from .3061 to .2940. These continuing changes motivate interpreting the table at its finite training budget. More distinct sites alone also does not order reconstruction: measurement noise and the specific spatial locations matter.

J.6 Numerical checks and executable sources

The exact-predictor implementation is `experiments/predictive_curiosity_published_gp.py`. Its independent NumPy audit replays the first 24 test trajectories for each budget, seed and policy, reconstructs observations, directly sums prefix/deletion log scores, and evaluates posteriors. Maximum discrepancies are 1.40×10^{-11} for reward and 1.34×10^{-13} for MSE.

`experiments/predictive_curiosity_paid_probe.py` implements the added objectives. Its independent audit checks observations, the charged budget, before/after-probe posteriors and all scores for the first 24 test trajectories per seed/objective; maximum discrepancy is below 3.1×10^{-10}. `experiments/learned_gp_acquisition.py` implements fitting, collection and both continuations. Independent replay of its ten evaluation groups per seed differs by less than 3×10^{-11}, and the fixed-policy refitting comparisons use identical trajectories. These are numerical audit samples; full saved arrays and checkpoint hashes are retained for reproduction.

Outputs are in `results/predictive_curiosity_published_gp_20260909/`, `results/predictive_curiosity_paid_probe_20260909/`, the three directories `results/learned_gp_acquisition_seed1001_20260909/`, `results/learned_gp_acquisition_seed1002_20260909/`, `results/learned_gp_acquisition_seed1003_20260909/`, and `results/learned_gp_acquisition_summary_20260909/`. Each contains per-seed tables or trajectory arrays, learning curves and source/checkpoint digests. Main GP figures are generated from these saved results without retraining.

K SENSOR RECORDS AND TASK DEPENDENCE

K.1 REPORT SELECTION WITH MISSING TEMPERATURE RECORDS

Intel Lab Data (Bodik et al., 2004) contains temperature reports from a deployed sensor network. We aggregate to five-minute medians, select 16 candidate sensors using training availability, and fit a Gaussian model on a rolling seven-day training window. For each following test day, we enumerate subsets of four or eight reports. The objectives are deletion reward, observation entropy, expected reconstruction MSE, and mutual information between selected and unselected reports (Krause et al., 2008); random subsets provide a reference. The budget counts requested aggregated reports. These objectives are optimized for the fitted model; test errors measure prediction of observed reports.

The complete-record analysis has 13 evaluable test days from 26 candidate dates, with 2,872 complete intervals. Deletion has higher daily MSE than reconstruction selection on 12/13 days at budget four and 11/13 at budget eight; median daily MSE ratios are 2.12 and 2.61. The follow-up fits incomplete training records by regularized Gaussian EM (50 iterations). It charges every requested report, including failed requests, and conditions only on received measurements. All methods are scored on the same available target readings. This expands evaluation to 21 days and changes both the record population and fitting procedure.

Coverage	Budget	Days	Deletion	Entropy	Recon.	MI	Random
Available	4	21	24.9059	28.0920	26.3346	19.0589	25.1227
Available	8	21	5.8098	18.2646	17.4934	10.1217	15.2076
At least 8	4	18	15.0365	11.6336	12.2603	9.7465	13.8442
At least 8	8	18	6.0771	6.6863	7.2214	5.1602	6.9771

Entries are means of daily MSE. With available targets, deletion exceeds reconstruction MSE on 14/21 days at both budgets, with median ratios 1.451 and 1.392. The means reverse because of tail days. Requiring at least eight valid targets was a prespecified coverage diagnostic. It retains 18 days, and still gives different mean orderings at the two budgets. Reported readings have no independent spatial temperature reference; missingness, device failures and model drift complicate interpreting error as fidelity to the physical field. Days and time intervals are not treated as independent identically distributed experimental repetitions.

The retained missing-report output includes all daily comparisons and the temperature quality filter. Its audit independently replays scattered test records for each date, budget and objective, checking received-only conditioning, shared targets and direct deletion scores; maximum discrepancy is below 8.5×10^{-12}. Results and protocol are `results/intel_lab_missing_reports_20260909/` and `notes/intel_lab_missing_reports_protocol_20260909.md`.

K.2 LOCATING THE REVERSAL ON MATCHED RECORDS

We match candidate sensor order, dates and exact timestamps between the original and missing-report analyses. The following sequence changes the fit and selected sets on identical complete test records, adds incomplete intervals on the same dates, then adds the remaining evaluable days. Each entry averages daily MSE gaps, deletion minus reconstruction selection, in squared degrees Celsius.

Evaluation	Days	Four-request gap	Eight-request gap
Original complete records	13	1.0961	0.1135
Change fit and sets	13	0.9608	2.2562
Add incomplete intervals	13	0.6188	1.0005
Include all evaluable days	21	-1.4288	-11.6836

This is an ordered accounting decomposition; the factors can interact. The complete-record model uses Ledoit–Wolf estimation, while the missing-report model uses regularized Gaussian EM. We therefore also cross their selected sets with both reconstruction models. All eight mean gaps per

budget, covering the two sources of selected sets, two decoders and two evaluation populations on the common 13 dates, remain positive. All predictions condition only on the received subset, and every comparison within a population uses the same available targets.

The daily gaps below retain all 21 dates. An asterisk marks a date added by the missing-report analysis. The number of targets is averaged over that day's evaluated intervals.

Date in 2004	Intervals	Mean targets	Four-request gap	Eight-request gap
03-07	288	16.00	2.1152	-0.0003
03-08	288	16.00	0.1672	0.2783
03-09	288	16.00	0.9880	0.2411
03-10	110	15.99	2.2035	0.1009
03-11	174	15.17	-0.2923	0.6003
03-12	288	15.97	0.0834	0.1183
03-13	288	15.99	0.1110	0.0890
03-14	288	15.95	0.3891	0.2513
03-15	193	15.81	0.7661	0.4872
03-16 *	156	12.76	1.4711	0.1658
03-17	288	15.97	0.1454	0.0331
03-18	288	13.33	-1.4930	0.0868
03-19 *	288	14.98	0.2468	0.0173
03-20	288	15.97	0.3432	-0.1859
03-21	288	14.58	2.5180	10.9063
03-22 *	288	12.63	-12.5322	-1.9723
03-23 *	288	9.35	-5.8837	10.1925
03-24 *	288	6.20	43.8803	-25.2104
03-25 *	201	1.30	-65.2314	-194.0834
03-28 *	288	1.00	0.0000	0.0000
03-29 *	86	1.00	0.0000	-47.4723

On March 25 the eight-request deletion set contains every sensor that still provides an evaluated report. It receives all available targets in every evaluated interval, so report-reconstruction error is zero. The task set receives $.3184$ reports per interval and leaves MSE 194.0834. At four requests, the 21-day gap is -1.4288; a leave-one-day-out diagnostic omitting March 25 gives $+1.7614$. All reported aggregate comparisons retain that date.

On March 29 each evaluated interval has one valid target. The eight-request advantage of 47.4723 comes from sensor 31 in four intervals, where deletion directly receives the target. Its fitted mean is 34.1454 degrees Celsius, while those four test readings average 2.1977. The original quality rule retains these readings. Their physical accuracy cannot be resolved from the report records alone.

K.3 Direct target measurements and prediction of other reports

To separate these two uses of acquisition, hold the model, selected sets, other received inputs and evaluation targets fixed. When predicting each available target k, withhold its own report from the decoder even if the requested set includes k. All other received selected reports remain. The requested set and its budget are unchanged; this is a prediction diagnostic rather than a competing acquisition policy.

For this report model, receiving a target gives its exact recorded value. Its squared reconstruction error is then zero. Per interval, original MSE equals the error after withholding each target's own report minus the reduction obtained by directly receiving that target. Both methods are evaluated on the same observed targets, including targets whose own reports were withheld. Averaging preserves this identity.

Requests	Selection	Original MSE	Own report withheld	Direct-report reduction
4	Deletion	24.9059	30.6344	5.7285
4	Task	26.3346	32.7998	6.4652
8	Deletion	5.8098	25.5398	19.7300
8	Task	17.4934	30.1466	12.6532

At eight requests, the difference in direct-report reductions is 7.0769, compared with the original MSE advantage of 11.6836. Withholding own reports reduces this advantage but preserves both mean reversals. We also retain prediction errors on available targets outside the union of the two requested sets, using the same targets for both methods and recording intervals with no eligible target. Those target sets depend on the compared selections, so their errors serve as a separate diagnostic.

K.4 Intervening on the selector's availability assumptions

Let C be the retained 16-report covariance and S a requested subset. For a received set A, write

$$V_C(A) = C - C_{:,A} C_{A,A}^{-1} C_{A,:}, \qquad V_C(\varnothing) = C. \tag{K.1}$$

For a bank of n availability patterns, let M_t be the assumed available inputs and O_t the assumed available targets. The selection criterion is

$$J_C(S) = \frac{1}{n} \sum_{t=1}^{n} \frac{1}{|O_t|} \sum_{k \in O_t} [V_C(S \cap M_t)]_{kk}. \tag{K.2}$$

The original task selector sets both M_t and O_t to all 16 sensors. The training-pattern selector uses the preceding seven days' empirical masks for both. The test-pattern selector uses the test day's empirical masks, with no test temperature values entering selection. It has advance knowledge of availability and serves as an oracle diagnostic. Every subset of size four or eight is enumerated under the same retained EM covariance. Actual evaluation always uses received inputs and available targets, regardless of the assumptions used for selection.

We further cross the two assumptions: use test-period input availability with all 16 targets, or test-period target availability while assuming all requested inputs arrive. These intermediate criteria isolate changes to the selection objective. Full-target conditional variance is a model expectation; no naturally missing target value is imputed for actual scoring.

Selection	Four-request MSE	Eight-request MSE
Original deletion	24.9059	5.8098
Original task	26.3346	17.4934
Task: training patterns	26.8779	20.6642
Task: test inputs, all targets	6.7285	2.2167
Task: complete inputs, test targets	8.0483	3.0133
Task: test inputs and targets	8.2640	2.4181

The test-pattern criteria all restore the task selector's mean advantage over deletion; the training-pattern criterion does not. Thus the change can be produced by altering the criterion's availability assumptions while keeping the temperature model fixed. The two test-mask factors come from the same records and interact; their improvements cannot be added as independent effects. These selectors optimize the fitted model's conditional variance, and do not establish optimality for actual report error.

K.5 Model predictions and realized task errors

For the original deletion and task sets, we also compute the model's expected error using the actual received subsets and actual target masks. These predictions differ substantially from realized errors:

Requests	Selection	Model-expected MSE	Realized MSE
4	Deletion	1.3571	24.9059
4	Task	0.9888	26.3346
8	Deletion	0.1803	5.8098
8	Task	0.6673	17.4934

At four requests, the model still predicts that the task set is better after actual masks are supplied, whereas realized mean error favors deletion. At eight requests, the model predicts the observed direction but greatly understates both errors. Correcting both selection assumptions gives lower model-expected error than correcting only inputs, yet higher realized error at both budgets. Availability knowledge therefore does not remove task-model error. Per-sensor training/test means and scales, received counts and target contributions are retained for all dates.

K.6 Reproduction and checks for the Intel interventions

Each intervention was specified after the preceding diagnosis and before its outcomes were inspected. All reuse the same test records and are exploratory attribution, rather than new independent data. No additional quality filter or favorable date subset is introduced. Code and stage-specific protocols retain this sequence.

Entry points are `experiments/intel_reversal_attribution.py`, `experiments/intel_reversal_target_intervention.py`, `experiments/intel_reversal_pattern_selection.py` and `experiments/intel_reversal_pattern_factorial.py`. Their corresponding result directories under `results/` have the same stems and suffix `_20260909`. They contain crossed daily comparisons, sensor contributions, selected sets, all enumerated selection objectives, per-interval predictions and availability counts. `experiments/audit_intel_reversal.py` independently reconstructs all 21 days from the original five-minute table and verifies candidate order, training records and test timestamps.

The independent replay checks 4,032 prediction vectors, 9,408 predictions with the target's own report withheld, 3,822 subset objectives and 644 summary rows. Maximum discrepancies are 4.13×10^{-13} for predictions, 2.48×10^{-12} for selection objectives and 2.85×10^{-14} for summary means. Original complete/missing endpoints reproduce within 1.46×10^{-12}, and the direct-report decomposition within 1.14×10^{-13}. The audit and input/protocol hashes are in `results/intel_reversal_attribution_20260909/independent_audit.json`.

K.7 Gas channels, causal history and concentration targets

The UCI dynamic gas dataset (Fonollosa, 2015) contains two records of a 16-channel array exposed to ethylene/CO and ethylene/methane mixtures. The task predicts the supplied concentration setpoints, rather than an independently measured instantaneous chamber concentration. We subsample to one-second readings, use chronological train/validation/test fractions $.5/.2/.3$, and leave 120-second gaps after split boundaries. Scaling is fitted on training data. A budget of four or eight means that many continuously active channels; all their past readings are available.

An initial selection uses the current signal and its ten-second change for each channel. A Gaussian covariance defines deletion, entropy, target-MSE and target-MI selection criteria, and all channel subsets of the permitted size are enumerated. Sensor covariance shrinkage is fitted only from sensor data; the target-informed criteria additionally use training labels. The retained sensor-only shrinkage run supersedes an early joint-shrinkage implementation. To test decoder dependence, we freeze these selected subsets and fit gradient-boosted regressors using either the short history or current readings and changes at lags 1, 5, 10, 30, 60 and 120 seconds. Validation chooses 100 or 300 boosting iterations and 15 or 31 leaf nodes, with learning rate $.05$, minimum leaf size 30 and L2 regularization 1. No past targets enter the decoder.

Mixture	History	Channels	Deletion	Entropy	Task MSE	Task MI
CO	Short	4	.3962	.3601	.3543	.3543
CO	Short	8	.3738	.4041	.3960	.3760
CO	Long	4	.3118	.3103	.3446	.3446
CO	Long	8	.3384	.4045	.3525	.3442
Methane	Short	4	.3337	.3462	.3124	.3124
Methane	Short	8	.3152	.3383	.3144	.3144
Methane	Long	4	.3317	.3202	.3201	.3201
Methane	Long	8	.3373	.3034	.3485	.3485

Each mixture also contains ethylene. Entries are normalized test MSE averaged over its two concentration targets. All methods use common eligible times. The corresponding all-16-channel errors are $.3778/.3455$ for short/long CO histories and $.3339/.2996$ for methane. Greater channel access does not improve this finite decoder in every comparison.

The ranking changes motivated an exploratory follow-up that matches the selector's history to the long-history decoder. It refits the Gaussian covariance on all 112 channel/history features and deletes each channel's seven-feature block as a unit. Shrinkage is again fitted only to sensor data and then applied to the joint sensor/target covariance; the sensor block is checked to remain independent of target labels. The same subset enumeration, decoder grid, chronological splits and test interval are reused.

Mixture	Channels	Deletion	Entropy	Task MSE	Task MI
CO	4	.3795	.3382	.3604	.3280
CO	8	.3137	.3907	.3626	.3581
Methane	4	.3233	.3651	.3183	.3136
Methane	8	.3373	.2974	.2949	.2949

The matched-history analysis shares the test period with the preceding diagnostic, so it is exploratory evidence rather than an independent confirmation. It retains both directions of the task comparison. Ten-minute block errors are saved descriptively; they are not treated as independent repetitions. Input reconstruction from timestamps and prediction replay have zero discrepancy; a direct conditional-entropy check of block deletion differs by at most 7.674×10^{-13}.

The formal initial run is `results/gas_sensor_selection_sensor_fit_20260909/`. Dynamic and matched-history results are `results/gas_dynamic_predictor_20260909/` and `results/gas_matched_history_20260909/`. Their `complete_results.md` files, saved predictions and audits retain all comparisons. Entry points are `experiments/gas_dynamic_predictor.py` and `experiments/gas_matched_history.py`; the reproduction guide maps the protocols and initial selection code.