WINNING COMPARISONS, LOSING PREDICTIONS: COMPARISON FEEDBACK MISLEADS WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

A world model can correct its next-step predictions from comparison feedback. It samples several candidate outcomes, and an oracle that sees the actual outcome reports which one came closest. Existing methods typically fit these answers with the pairwise Bradley–Terry (BT) objective from preference learning, which raises the likelihood of the chosen candidate over each of the others. However, does winning more comparisons always make a world model better? To answer it, we use an exact oracle and no learned reward model, so any gap we find can only come from the learning rule. We first show in theory that winning more comparisons can make a world model predict worse, and trace this failure to three gaps between feedback and prediction. BT training converges to a law biased by how often each outcome is repeated among the candidates. A correctly specified feedback loss removes this bias, yet it can still mis-rank predictors near the truth, because each answer supervises only the probability of a set of outcomes. The law that BT fits can also flip a planner's optimal action. Experiments with IRIS on three Atari games and Delta-IRIS on Crafter confirm each gap. BT feedback worsens prediction when the true outcome is itself a candidate, and equalizing how often candidates are repeated removes most of this damage on the Atari games. On Crafter, the checkpoint that BT feedback ranks best is the worst predictor of these outcomes in every fit. Planning with it loses most of the original model's return.

1 INTRODUCTION

By learning to predict the consequences of actions, world models let agents plan and learn policies in imagination, with remarkable results from Atari games to Minecraft (Ha & Schmidhuber, 2018; Micheli et al., 2023; Alonso et al., 2024; Hafner et al., 2025). However, where training data are sparse, world models err and produce physically implausible futures. Several works have therefore sought to correct world models with feedback on their sampled predictions. For instance, RLVR-World rewards each sampled prediction according to how closely it matches the observed outcome (Wu et al., 2025). RENEW instead learns from comparisons between predicted futures to repair the model errors that a policy exploits in imagination (Bhamidipaty et al., 2026). Under such comparison feedback, the model samples several candidate outcomes, and an oracle that observes the realized outcome reports which candidate is nearest to it (Figure 1).

Existing methods such as RENEW train on these answers with the pairwise Bradley–Terry (BT) objective, the standard comparison model in preference learning (Bradley & Terry, 1952; Christiano et al., 2017; Ouyang et al., 2022). BT treats each answer as a preference for the selected candidate over every other candidate and raises its likelihood relative to theirs. A lower BT loss therefore indicates that the model wins more comparisons by assigning higher probability to the selected candidates. Without outcomes, the same loss is also the natural criterion for checkpoint selection.

Research Question

However, does winning more comparisons always make a world model better? If not, why?

To answer this question, we use an exact oracle and no learned reward model, so any failure we observe can only come from the learning rule. Unlike in reward-model overoptimization (Gao et al., 2023), no learned proxy is involved. We evaluate each adapted model by its negative log-likelihood (NLL) of the realized outcome. A query is covered when its true outcome is among the candidates, so that the oracle selects the truth itself as the winner.

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

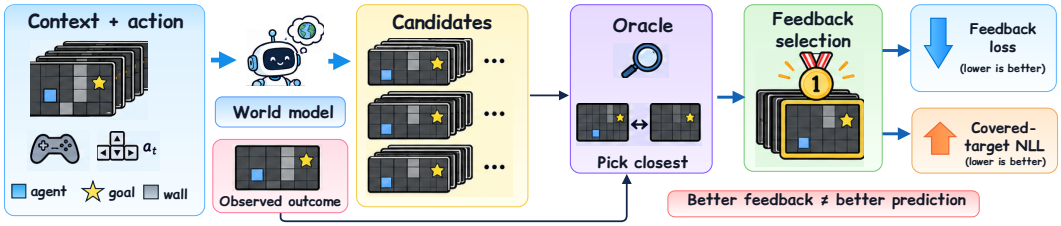

Figure 1: **Selecting a world model by comparison feedback picks a worse predictor.** The world model proposes candidate outcomes for a context and action, and an oracle that sees the observed outcome picks the closest one. Choosing checkpoints by this feedback lowers the feedback loss but raises the NLL on covered targets, those among the candidates.

We first prove in theory that winning more comparisons can make a world model predict worse, and from this analysis we derive three gaps between feedback and prediction. BT training converges to a law biased by candidate multiplicity, the number of times each outcome appears among the candidates. This bias arises because every copy of a loser contributes a separate comparison and each batch is normalized by its number of comparisons. We derive the exact optima of the BT loss, which equal the true law with three or more outcomes only under equal multiplicities. When candidates are sampled from the model itself, even the true law can fail to be stationary, and additional candidates do not remove the bias (Figure 3a). A correctly specified feedback loss removes this bias, yet it can still mis-rank predictors near the truth. Scoring the model by the probability of the oracle's answer yields such a loss, the channel likelihood. This failure arises because each answer constrains only the probability of the winner's cell, the set of outcomes closer to the winner than to any other candidate. The allocation of probability within the cell therefore remains unsupervised, and in a three-outcome example the channel likelihood reverses the ranking of some pairs of predictors near every interior truth. Beyond prediction, the law fitted by BT can also flip the optimal action of a planner. In a minimal control task, it overestimates the probability that a risky action continues, so the planner prefers the risky action even when the safe action is optimal.

Experiments with IRIS on three Atari games and Delta-IRIS (Micheli et al., 2024) on Crafter confirm each gap. BT feedback raises covered NLL above that of the original model in every world model (Table 1). Equalizing only candidate multiplicities removes most of this increase on the Atari games, identifying multiplicity as its main source. Figure 1 illustrates that BT feedback also misleads checkpoint selection. On Crafter, its loss is lowest exactly where covered NLL is highest in every fit. Under the channel likelihood on Breakout, the increase in covered NLL lies within the winner's cell, which feedback leaves unsupervised. In the control task, every BT fit selects the suboptimal action. Planning with the checkpoint that BT feedback selects on Crafter retains less than half of the original model's return. Our contributions are twofold.

- We reveal that winning more comparisons does not necessarily make a world model better. Our theory derives three gaps that cause this failure, namely a multiplicity-biased law under BT, mis-ranking under a correctly specified loss, and action flips under this law.
- We verify the three gaps with four pretrained world models across different tasks, from three Atari games to the open-world game Crafter, and in a minimal control task.

2 Related Work

World models. World models are trained to predict observed transitions and then used to plan or to train policies in imagination (Ha & Schmidhuber, 2018; Micheli et al., 2023; 2024; Hafner et al., 2025; Alonso et al., 2024). Likelihood need not track control performance (Lambert et al., 2020), and value-aware losses target control directly (Farahmand et al., 2017). We keep likelihood as the goal, change the training signal to comparisons, and ask whether a better comparison score still means better prediction and a better action (Section 4.3).

Comparison feedback. BT models (Bradley & Terry, 1952) turn pairwise judgments into training signals for preference learning (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2023). There, BT is an assumption about human judges. Its known failures include reward-model overoptimization (Gao et al., 2023), overfitting to deterministic preferences (Gheshlaghi Azar et al., 2024) and a falling likelihood of preferred responses (Razin et al., 2025). RENEW brings comparison feedback

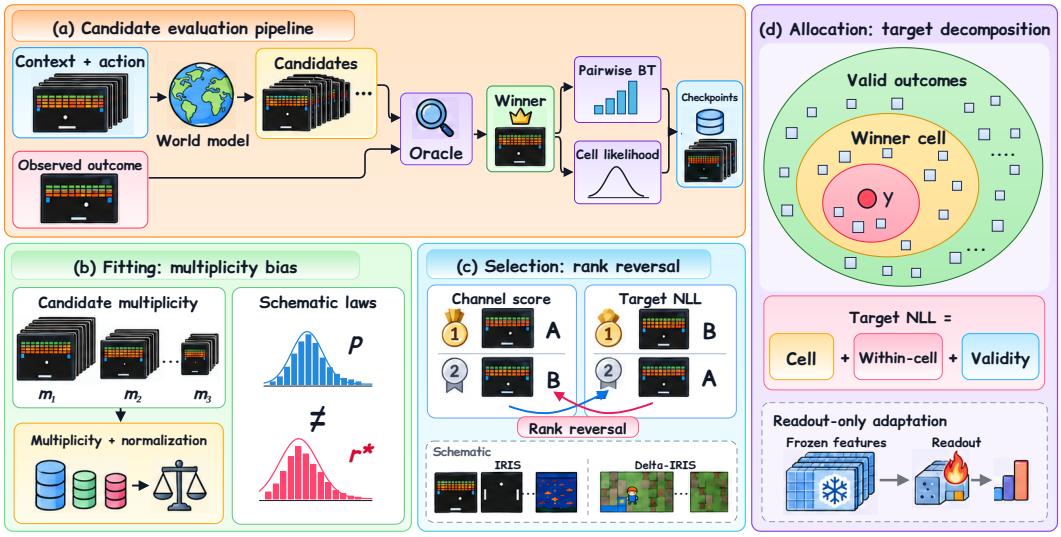

Figure 2: **Comparison feedback departs from prediction in fitting and in selection.** (a) The oracle returns the candidate nearest to the observed outcome, and this winner trains either pairwise Bradley–Terry or the channel likelihood of the winner's cell. (b) Candidate multiplicity and comparison-count normalization make Bradley–Terry fit a law that differs from the true outcome law. (c) Even the correctly specified channel likelihood can rank two predictors in the opposite order of their target NLL. (d) Each answer reveals only the cell that contains the observed outcome, so it supervises the cell term of the target NLL and leaves the within-cell term free. Adapting only the readout nearly removes the within-cell loss, which therefore arises mostly in the layers below it.

to world models and evaluates it with an exact oracle, whose answers follow from distances to the realized outcome rather than from a BT model (Bhamidipaty et al., 2026). We thus ask what BT fits under this known judgment rule, where no learned proxy can explain the gap. The fitted law extends the density-ratio bias of uncorrected negative sampling with an exact finite-batch factor (Gutmann & Hyvärinen, 2012; Ma & Collins, 2018). Its correctly specified alternative is a set-valued likelihood, as in partial-label learning (Jin & Ghahramani, 2002; Liu & Dietterich, 2012), yet it can rank imperfect predictors differently from prediction (Gneiting & Raftery, 2007).

3 Problem Setup

Under comparison feedback, the world model never sees the next outcome; it sees which of its own samples came closest (Figure 2a). This section formalizes it. Every later result concerns what the oracle reveals, the two losses that learn from it, or the prediction loss that judges them.

Outcomes and predictions. At a context H, a state–action history, the next outcome Y takes one of n values $\mathcal{Y} = \{y_1, \dots, y_n\}$, as in tokenized world models whose outcomes are codebook entries (Micheli et al., 2023). We drop H from the notation. The environment draws Y from a true law p with full support, and the world model predicts a law $r = \mathrm{softmax}(z)$ with logits z. We write $p_i = p(y_i)$, $r_i = r(y_i)$, and $r(S) = \sum_{y \in S} r(y)$ for the probability that the model gives to a set S of outcomes; feedback will reveal such sets rather than single outcomes.

What the oracle reveals. Before Y is observed, the model proposes a bank $C = (\hat{y}_1, \dots, \hat{y}_K)$ of K candidate outcomes, so C is drawn from a law ν independently of Y. Sampling can repeat an outcome, and m_j counts the copies of y_j in the bank. An oracle that sees Y compares it with the candidates through a distance d, as the programmatic oracle of RENEW does (Bhamidipaty et al., 2026), and names the closest one as the winner.

Definition 3.1 (Nearest-candidate feedback)**.** *The oracle returns the* winner y_w*, and* S_w *denotes its* cell*,*

$$y_w := \arg\min_{\hat{y} \in C} d(\hat{y}, Y), \qquad S_w := \{y \in \mathcal{Y} : \; d(y_w, y) < d(\hat{y}, y) \;\; \forall\, \hat{y} \in C,\; \hat{y} \neq y_w\}. \tag{1}$$

The cell collects every outcome that would produce the same answer, so y_w wins exactly when $Y \in S_w$. Observing the winner is the same as observing that Y lies in S_w, a set that partial-label learning calls a candidate label set (Cour et al., 2011; Liu & Dietterich, 2012). For example, let $\mathcal{Y} = \{-1, 0, 1\}$ with $d(y, y') = |y - y'|$ and bank $C = (0, 1)$. Candidate 0 has cell $\{-1, 0\}$ and candidate 1 has cell $\{1\}$, so the answer "0 wins" cannot tell $Y = -1$ from $Y = 0$. This holds even when $Y = 0$ and the winner is the truth itself. A query is *valid* when the bank holds at least two distinct outcomes and only one of them is nearest to Y. Every candidate with a different outcome is then a *loser* y_l. Ties between distinct outcomes carry no clean answer, so we discard them; A denotes the outcomes that give a valid query.

Win probability. Because the answer is the event $Y \in S_w$, its probability under any law is that law's mass on the cell, renormalized to valid queries. Under the model, y_w wins with probability

$$Q_r(y_w) := \frac{r(S_w)}{r(A)}, \tag{2}$$

and Q_p is the same ratio under the true law. The bank is drawn independently of Y, so Q_p is how often each candidate actually wins, and Q_r is the model's forecast of the oracle's answer. In the example, no outcome ties, so $r(A) = 1$, $Q_r(0) = r_{-1} + r_0$ and $Q_r(1) = r_1$. The model is credited for any mass in the winner's cell, however it is split, and the channel likelihood maximizes Q_r.

Pairwise Bradley–Terry loss. RENEW learns from the answer as preference learning does (Christiano et al., 2017; Ouyang et al., 2022). It reads "y_w is nearest" as "y_w is preferred to every loser" and fits a BT model that scores each outcome by its log-likelihood. One comparison costs

$$-\log \sigma\big(\log r(y_w) - \log r(y_l)\big) = -\log \frac{r(y_w)}{r(y_w) + r(y_l)}, \qquad \sigma(u) = \frac{1}{1 + e^{-u}}, \tag{3}$$

and this cost falls as the winner gains probability relative to that loser. Two choices in RENEW's released code then shape what BT learns. First, every copy of a loser is a separate comparison, so an outcome sampled m times is pushed down m times; copies of the winner add none. Second, the loss averages over all comparisons in a batch. Let N_{bij} count the times y_i beats y_j in context b, and let $V_b = \sum_{i \neq j} N_{bij}$ count the comparisons of that context. The *pairwise BT loss* of a batch of B contexts is

$$L_B(z) := \frac{\sum_b \sum_{i \neq j} N_{bij} \log(1 + e^{z_j - z_i})}{\sum_b V_b + \epsilon}. \tag{4}$$

Each term is (3) written in logits, and $\epsilon \geq 0$ guards against a batch without comparisons. Because all contexts share one denominator, a context with more comparisons carries more weight. In the example with bank $(0, 1, 1)$ and $Y = 0$, the winner 0 beats the loser 1 twice, so this context adds two comparisons against 1 where bank $(0, 1)$ adds one. Section 4.1 derives the law that BT then fits.

Channel likelihood. The pairwise loss ranks candidates against each other, whereas the oracle reports a cell. We therefore treat the oracle as a channel from Y to the winner and score the model by the likelihood $Q_r(y_w)$ of the winner it reports. This defines the *channel likelihood*, whose risk is the expected negative log-likelihood over valid queries,

$$\mathcal{L}_{\mathrm{ch}}(r) := \mathbb{E}\big[-\log Q_r(y_w) \mid Y \in A\big]. \tag{5}$$

Because Q_p is the actual law of the answer, the true law minimizes $\mathcal{L}_{\mathrm{ch}}$; as in partial-label learning, the channel likelihood is correctly specified. In the example, the answer "0 wins" costs $-\log(r_{-1} + r_0)$. This loss moves mass into the cell $\{-1, 0\}$ but is indifferent to how that mass is split. Section 4.2 shows when this indifference still identifies p and when it mis-ranks models.

Prediction and the cell decomposition. Both rules learn from feedback alone, yet a world model is used to predict what happens next, so it is judged by the probability it gives to the actual outcome. We measure prediction by the risk $\mathcal{L}_{\mathrm{pred}}(r) := \mathbb{E}[-\log r(Y)]$, the expected negative log-likelihood (NLL) of the outcome. A query is *covered* when Y is among the candidates; there the winner is Y itself, so feedback points straight at the truth. If feedback helps prediction anywhere, it should help

on covered queries. The following identity locates the part of the NLL that feedback can reach. For every valid query with outcome y and winner y_w,

$$-\log r(y) = \underbrace{-\log Q_r(y_w)}_{\text{winner cell}} + \underbrace{-\log \frac{r(y)}{r(S_w)}}_{\text{within cell}} + \underbrace{-\log r(A)}_{\text{validity mass}}. \tag{6}$$

Feedback supervises the first term, whose average over valid queries is $\mathcal{L}_{\text{ch}}$. The within-cell term measures how the model divides the cell's mass between y and the rest of the cell, and this answer does not constrain it (Figure 2d). Its third term depends only on $r(A)$, the probability of a valid query. In the example with $Y = 0$, a covered query, the identity reads $-\log r_0 = -\log(r_{-1} + r_0) - \log\big(r_0/(r_{-1} + r_0)\big)$. Feedback can raise $r_{-1} + r_0$ while mass drifts from 0 to -1, so the feedback loss falls while the NLL of the truth rises. Section 5.4 measures both terms on Breakout.

4 Three Gaps between Feedback and Prediction

Winning more comparisons does not always make a world model better. On Crafter, the checkpoint with the lowest BT feedback loss has the highest covered NLL, about three times that of the original model (Section 5.2). We explain why this happens when fitting a model (Section 4.1), selecting a checkpoint (Section 4.2) and choosing an action (Section 4.3).

Key Findings

- **What does Bradley–Terry training fit?** A law biased by the bank. Outcomes with more copies lose probability, and the normalization adds a factor that depends on the winner. With two model-sampled candidates, the true law is never stationary (Section 4.1).
- **Can a correct feedback loss rank checkpoints?** Not always. The channel likelihood removes the fitting bias, yet it can prefer the worse of two nearby predictors (Section 4.2).
- **Does the bias change decisions?** Yes. The law fitted by Bradley–Terry training can flip a planner's optimal action, whereas the channel likelihood of the same comparisons acts optimally (Section 4.3).

4.1 Fitting Gap: Bradley–Terry Fits a Biased Law

A sound training loss should be minimized by the true law, at least with an exact oracle. BT instead fits the odds of two outcomes to how often one beats the other. These win counts also depend on the bank, because each copy of a loser adds a comparison and each batch is normalized by its number of comparisons. The next theorem shows the effect when every bank contains every outcome.

Theorem 4.1 (Multiplicity bias)**.** *Let every bank contain $m_i \geq 1$ copies of every outcome y_i. The expected pairwise BT loss (4) is then uniquely minimized by*

$$r_i^\star \propto \frac{p_i\, h_i}{m_i}, \qquad h_i = \mathbb{E}\big[(\textit{comparisons in the batch} + \epsilon)^{-1} \;\big|\; y_i \textit{ wins a given context}\big]. \tag{7}$$

For $n \geq 3$, $r^\star = p$ if and only if all multiplicities are equal. (Proof in Sections A.4 and A.5.)

Theorem 4.1 shows that BT is biased even when the winner reveals the outcome. An outcome with more copies loses more comparisons, so BT divides its probability by its number of copies (Figure 2b). The normalization adds a factor that depends on which outcome won; in large batches this factor is nearly constant, and the bias is the density-ratio distortion of uncorrected negative sampling (Gutmann & Hyvärinen, 2012; Ma & Collins, 2018). Giving every outcome the same number of copies removes both factors, which Balanced BT in Section 5 uses as a control. Without full support, multiplicity still weights each loser's gradient (Section A.8), as Section 5.3 tests.

Model-sampled candidates. RENEW samples the bank from the model being trained and does not differentiate through the sampling. The loss therefore changes with the model, and training can only settle at a stationary point of the update (Section A.3).

Theorem 4.2 (Two sampled candidates)**.** *Let $\mathcal{Y} = \{-1, 0, 1\}$ with $d(y, y') = |y - y'|$ and $p = (a, 1-2a, a)$ with $0 < a < \frac{1}{2}$, and draw each bank as $K = 2$ independent samples from the model r. For every batch size $B \geq 1$ and every $\epsilon \geq 0$, the unique stationary point of this update with $r_{-1} = r_1$ is*

$$r^\star = \tfrac{1}{1+a}\,(a,\ 1-a,\ a). \tag{8}$$

The true law is never stationary; $r^\star$ overstates the center by $2a^2/(1+a)$. (Proof in Section B.3.)

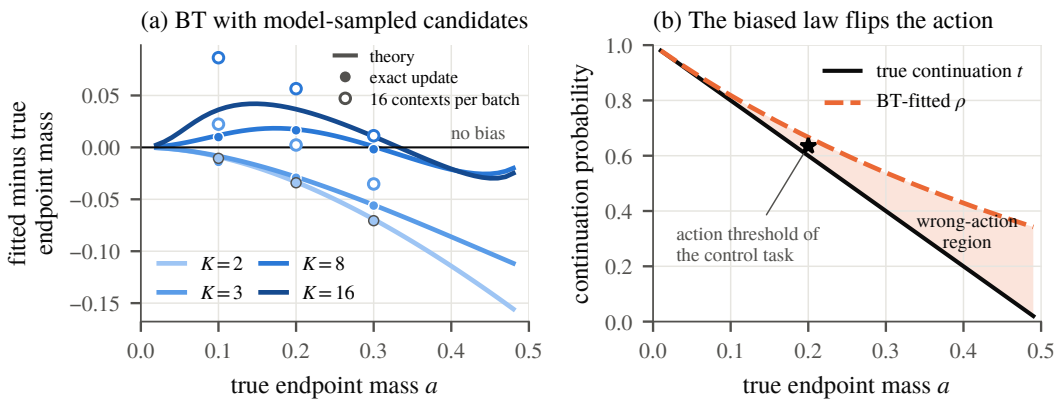

Figure 3: **Bradley–Terry training fits a biased law, and the bias can flip an action.** (a) More candidates do not remove the bias. The stationary law departs from the true endpoint mass except at isolated values, and the sign of the error changes with the number of candidates and the endpoint mass. Simulated updates follow the curves, and sharing one comparison count across sixteen contexts shifts the stationary law further. (b) With two candidates, the fitted continuation probability always exceeds the true one, so an action threshold anywhere in the shaded band leads to the wrong action.

Theorem 4.2 shows that BT with model-sampled candidates never settles at the truth, for any batch size. The cause is geometric. In a bank with the center and one endpoint, the center also wins when the outcome is the opposite endpoint, because it lies between the two. BT counts these wins as evidence for the center. Each valid query yields a single comparison, so the normalization does not move the stationary point. With more candidates the bias persists and can change sign (Figure 3a).

4.2 Selection Gap: A Correct Loss Can Mis-rank Models

The channel likelihood removes the fitting gap. Its excess risk is an average Kullback–Leibler divergence between true and modeled win probabilities (Section A.1), so the true law minimizes it, uniquely when these probabilities determine the law. Checkpoint selection, however, ranks imperfect models (Figure 2c), and a loss minimized at the truth can rank them differently from the prediction risk (Gneiting & Raftery, 2007). The next proposition shows this happens arbitrarily near the truth.

Proposition 4.3 (Near-truth ordinal reversal)**.** *Let $\mathcal{Y} = \{0, 1, 2\}$ with squared distance, and draw bank $(0, 1)$ with probability $\lambda \in (0, 1)$, else $(1, 2)$, independently of Y. The channel likelihood identifies every full-support p, yet for all such p and λ, every neighborhood of p contains predictors r, s with $\mathcal{L}_{\mathrm{pred}}(r) > \mathcal{L}_{\mathrm{pred}}(s)$ but $\mathcal{L}_{\mathrm{ch}}(r) < \mathcal{L}_{\mathrm{ch}}(s)$. (Proof in Section A.7.)*

Proposition 4.3 shows that a correctly specified feedback loss can prefer the worse of two nearby predictors. Each bank reveals whether the outcome is one particular end, so the channel risk scores the two end probabilities separately. The prediction risk also scores the middle probability, which absorbs the errors of both ends. Same-sign end errors therefore add up in the middle and raise the prediction risk alone. A predictor with small same-sign errors can thus win on channel risk and lose on prediction. The reversal persists at finite distance (Section A.7).

4.3 Decision Gap: A Biased Law Can Flip Actions

A biased law matters when it changes what the agent does. In the smallest such task, a safe action ends the episode with a fixed reward; a risky action pays a reward and repeats the state if the central outcome occurs. The risky action is optimal exactly when the central outcome is likely enough.

Proposition 4.4 (Decision flip)**.** *In this task with $p = (a, 1-2a, a)$, let the rewards be $R_{\mathrm{risk}}, R_{\mathrm{safe}} > 0$ and the discount $\gamma \in (0, 1)$, and let BT learn from the adjacent banks $(-1, 0), (0, 1)$. Let $t = 1 - 2a$ and $\rho = (1 - a)/(1 + a)$ be the true and fitted continuation probabilities. Any reward in*

$$\frac{R_{\mathrm{risk}}}{1 - \gamma t} < R_{\mathrm{safe}} < \frac{R_{\mathrm{risk}}}{1 - \gamma \rho} \tag{9}$$

makes the safe action optimal while the BT-fitted model picks the risky one. (Proof in Section A.6.)

Proposition 4.4 turns the overstated center into a wrong decision. The fitted model overestimates how often the risky action continues and so overvalues it. Any safe reward between the true and believed values flips the choice, and such rewards exist because BT overstates the center (Figure 3b). The channel likelihood recovers the true law from the same comparisons and acts optimally (Section 5.5).

5 Experiments

We answer four questions. **Q1:** Does feedback select good checkpoints? **Q2:** Does candidate multiplicity shift the fitted law? **Q3:** Where is prediction lost? **Q4:** Do the gaps change decisions?

5.1 Experimental Setup

Datasets. We use the Atari games Breakout, Pong and Seaquest, with recorded Minari expert trajectories (Younis et al., 2024), and Crafter (Hafner, 2022), with procedurally generated worlds. Test data come from held-out Atari episodes and new Crafter worlds played by the released agent (Section C). For each target token, a programmatic oracle sees the true token and answers a nearest-candidate query (Definition 3.1) over $K = 8$ candidates drawn once from the original model.

Models. We adapt the released IRIS (Micheli et al., 2023) checkpoints on the Atari games and the Delta-IRIS (Micheli et al., 2024) checkpoint on Crafter, and call them the *original models*. Both predict the next frame as discrete tokens. Each outcome is thus one token, $\mathcal{Y}$ is the tokenizer's codebook, and d is the squared Euclidean distance between codewords. The tokenizers stay frozen, and the transformer is adapted for three epochs on IRIS and 500 updates on Delta-IRIS.

Objectives. **Sampled BT** applies the pairwise loss (4) to the banks as drawn, with the pairs and normalization of RENEW. Two controls learn from the same banks and winners and each change one part of this rule. **Balanced BT** gives every distinct candidate equal multiplicity and thus isolates the effect of multiplicity. **Channel** maximizes the channel likelihood, which removes the fitting bias and exposes the part of prediction that feedback does not supervise. For comparison, we also report the original model and **Supervised** next-token likelihood, which observes the true token. All objectives share data, banks, initialization, optimizer and budget, with three fitting seeds, reported as mean and standard deviation. We report token NLL on covered queries and on all tokens.

5.2 Q1: Does Feedback Select Good Checkpoints?

We record every checkpoint of each objective on Crafter, every 50 updates over three seeds, and score it by validation feedback loss and validation covered NLL. Each fit keeps the fitted checkpoint with the lowest feedback loss, a rule fixed before we generated new test worlds.

Under Sampled BT, covered NLL rises as feedback loss falls (Figure 4), so the pick is the worst checkpoint (Figure 6). On new test worlds, its covered NLL exceeds three times that of the original model. Balanced BT flattens this relation, yet its pick is also the worst. Under the channel likelihood, both scores fall together and the original model ranks first. On Breakout and Pong, feedback selection also costs test NLL, more under Sampled BT than under channel feedback (Table 4).

Takeaway

Bradley–Terry feedback selects the worst Crafter checkpoint for covered prediction. The channel likelihood orders checkpoints like prediction, with the original model first.

5.3 Q2: Does Multiplicity Shift the Fitted Law?

We adapt each original model with each feedback objective for a fixed budget and report token NLL on covered queries and on all target tokens in Table 1. To isolate multiplicity, we also build two views of the same banks. Both keep the candidates, winners and discarded ties, but they give paired candidates one and seven copies, swapped between the views, and we train on each view.

Sampled BT raises covered NLL above the original model in every world model, even on Pong and Seaquest, where it lowers all-token NLL. Supervised training moves covered NLL far less, so this increase comes from the feedback objective. Balanced BT differs only in candidate multiplicity and

Table 1: **Pairwise feedback degrades covered prediction in every world model.** Balancing multiplicities removes most of the increase of Sampled BT on the Atari games, where the channel likelihood attains the lowest covered NLL among the feedback objectives.

NLL ↓	Breakout		Pong		Seaquest		Crafter	
(nats)	Covered	All	Covered	All	Covered	All	Covered	All
Original	0.560	1.982	0.441	4.678	1.513	3.516	0.486	1.196
Comparison feedback								
Sampled BT	$2.981_{\pm.123}$	$3.431_{\pm.094}$	$0.953_{\pm.041}$	$1.949_{\pm.015}$	$2.697_{\pm.005}$	$2.991_{\pm.018}$	$0.792_{\pm.007}$	$\mathbf{1.752}_{\pm.029}$
Balanced BT	$1.316_{\pm.016}$	$2.113_{\pm.023}$	$0.547_{\pm.028}$	$1.732_{\pm.015}$	$1.960_{\pm.025}$	$2.412_{\pm.010}$	$\mathbf{0.713}_{\pm.005}$	$1.769_{\pm.015}$
Channel	$\mathbf{0.740}_{\pm.026}$	$\mathbf{1.773}_{\pm.014}$	$\mathbf{0.531}_{\pm.021}$	$\mathbf{1.516}_{\pm.057}$	$\mathbf{1.486}_{\pm.044}$	$\mathbf{1.968}_{\pm.026}$	$0.724_{\pm.020}$	$1.849_{\pm.009}$
Outcome supervision								
Supervised	$0.637_{\pm.004}$	$1.276_{\pm.004}$	$0.456_{\pm.001}$	$1.089_{\pm.007}$	$1.276_{\pm.002}$	$1.414_{\pm.004}$	$0.500_{\pm.002}$	$1.076_{\pm.004}$

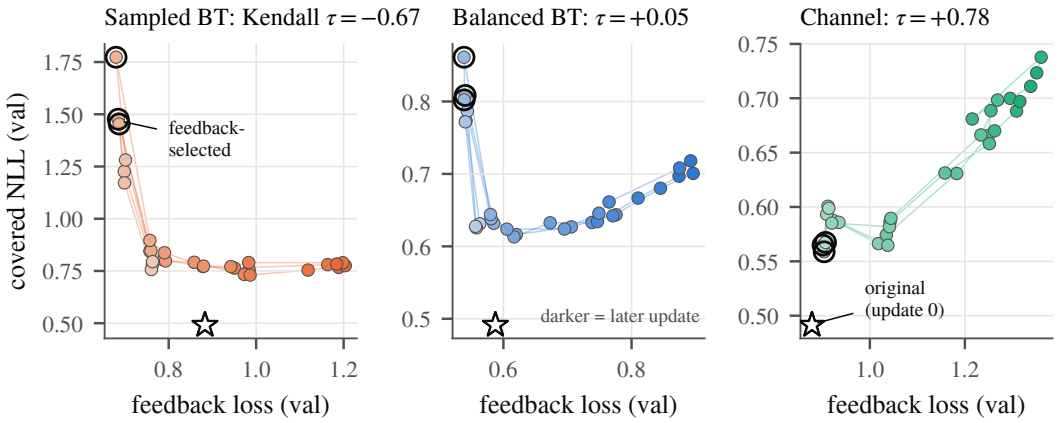

Figure 4: **Sampled BT ranks Crafter checkpoints against covered prediction; the channel likelihood ranks them with it.** Under Sampled BT, lower feedback loss tends to mean higher covered NLL, so its picks predict worst. Balanced BT flattens this trend, yet its picks still predict worst. The channel likelihood orders checkpoints like covered NLL, with the original model first.

removes most of this increase on the Atari games. In the two views, candidates on average end with less probability in the view that gives them more copies, in every world model, episode and seed (Sections D.2 and D.3). This is the direction that Theorem 4.1 gives for full-support banks.

Takeaway

Candidate multiplicity causally shifts the fitted law. Candidates with more copies lose probability, and balancing multiplicities removes most of the covered damage on the Atari games.

5.4 Q3: Where Is Prediction Lost?

On covered valid Breakout targets, we split the change in NLL into the winner-cell and within-cell terms of (6), as in Figure 2d. We adapt either all layers or only the final readout layer, with channel feedback or with supervised next-token training, and report each term in Table 2.

With all layers adapted, channel feedback improves the winner-cell term but worsens the within-cell term by more, so covered NLL rises. Supervised training of all layers also loses within the cell, but less. Adapting only the readout keeps the cell gain and nearly removes the within-cell loss.

Takeaway

Prediction is lost inside the winner cell, which comparison feedback leaves unsupervised. On Breakout, most of this loss arises from updating the layers below the readout.

Table 2: **Channel feedback loses covered prediction inside the winner cell, mostly through the layers below the readout.** On covered valid Breakout targets, adapting all layers improves the cell term but loses more within the cell, where supervised training loses less. Adapting only the readout achieves the largest cell gain and nearly removes the within-cell loss.

Objective	Adapted layers	Δ target NLL	Δ cell NLL	Δ within-cell NLL
Channel	All	$+0.185_{\pm.019}$	$-0.031_{\pm.009}$	$+0.216_{\pm.012}$
Channel	Readout	$\mathbf{-0.033}_{\pm.002}$	$\mathbf{-0.051}_{\pm.001}$	$+0.018_{\pm.001}$
Supervised	All	$+0.127_{\pm.006}$	$-0.042_{\pm.002}$	$+0.169_{\pm.004}$
Supervised	Readout	$+0.002_{\pm.002}$	$-0.046_{\pm.001}$	$+0.049_{\pm.001}$

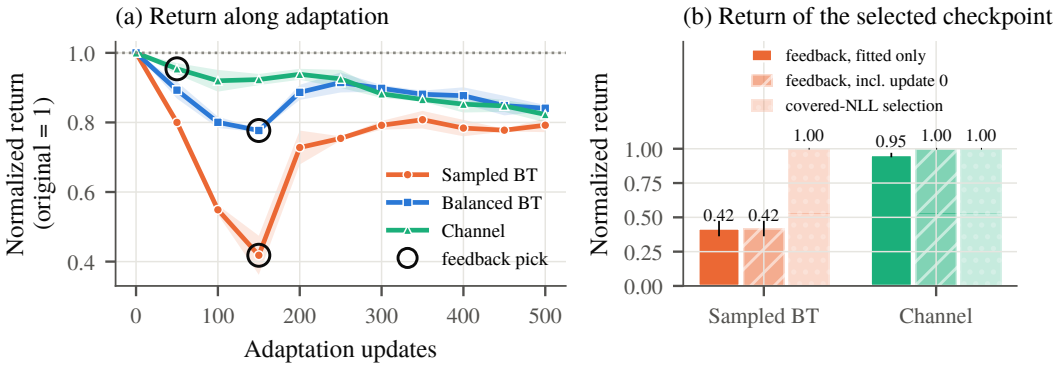

Figure 5: **Feedback selection costs downstream return on Crafter.** (a) Return falls most exactly where both Bradley–Terry objectives select and stays below the original model afterwards. (b) The Sampled BT pick loses most of the original model's return. The channel likelihood keeps nearly all of it, and all of it when the original model is admitted, like selection by covered NLL.

5.5 Q4: Do the Gaps Change Decisions?

In the control task of Section 4.3, we place the action threshold inside the band of Figure 3b. BT and the channel likelihood learn from the same adjacent comparisons and start from two initializations with three seeds each (Sections C.5 and D.4). On Crafter, we plan with every recorded checkpoint (Section C.3) and report its return relative to the original model in Figure 5.

In the control task, every BT fit chooses the risky action, whereas every channel fit chooses the optimal safe one. On Crafter, return falls most at the checkpoints that BT feedback selects. The Sampled BT pick loses most of the original model's return, whereas the channel pick keeps nearly all of it. In both settings, the fitting bias of BT therefore turns into wrong actions and lost return.

Takeaway

The fitting bias changes decisions. From the same comparisons, Bradley–Terry training picks the wrong action and loses return, whereas the channel likelihood keeps the right action and nearly all of the return.

6 Conclusion

The central contribution of this work is a systematic investigation into whether, and why, winning more comparisons can fail to make a world model better. We find that comparison feedback and prediction come apart at three points, even with an exact oracle and no learned reward model. Pairwise BT training converges to a law biased by candidate multiplicity and comparison-count normalization. The channel likelihood removes this bias but can still mis-rank models, because each answer leaves the split of probability inside the winner cell unsupervised. In turn, the fitted bias can flip an agent's action. A lower feedback loss can therefore mean better agreement with comparisons and worse prediction. More broadly, comparison feedback is evidence about sets of outcomes, so a feedback objective should be judged by the outcome law it recovers.

Ethics Statement

No human subjects took part in this work. Comparison labels come from a programmatic oracle, and the finite control task uses synthetic data. The visual experiments use public expert trajectories, the Crafter environment and released IRIS and Delta-IRIS weights, none of which contain personal data. These findings concern the safe use of world models. Selecting or adapting a world model by comparison feedback can deploy a worse predictor, and an agent that plans with it can choose a worse action. Extending the study to human judges would require informed consent and ethics review.

Reproducibility Statement

The supplementary material contains anonymized code for the experiments. Sections 3 and 4 state the theoretical results with their assumptions, and Sections A and B give complete proofs. Section B.4 checks the closed-form stationary points against simulations of the exact stochastic update. Section C specifies the data, released model weights, candidate banks, seeds, optimizer, batch size and budget of each experiment, and records which endpoints were fixed before the test data were decoded. Section D reports the per-seed and per-episode results. Rerunning the experiments requires no human labels, because they use released weights, public trajectories and environments, and a programmatic oracle in place of human judges.

AI Use Statement

In this work, we used generative AI tools to implement and run the experiments. We have not used generative AI tools to propose or refine hypotheses, to develop the theory, or to formulate or prove mathematical claims. Nor have we used them to generate synthetic data, clean or reformat data, or translate text. Qualitative and thematic data analysis is not applicable to this work. We also used generative AI tools to search and summarize the literature and to polish the writing. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems*, 2024.

Logan Mondal Bhamidipaty, Mykel Kochenderfer, and Subramanian Ramamoorthy. RENEW: Towards learning world models and repairing model exploitation from preferences. arXiv preprint arXiv:2607.14180, 2026. URL https://arxiv.org/abs/2607.14180.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 39(3–4):324–345, 1952. doi: 10.1093/biomet/39.3-4.324.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12(42):1501–1536, 2011. URL https://jmlr.org/papers/v12/cour11a.html.

Amir-massoud Farahmand, André Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455, 2024.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://sites.stat.washington.edu/raftery/Research/PDF/Gneiting2007jasa.pdf.

Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13 (11):307–361, 2012. URL https://jmlr.org/papers/v13/gutmann12a.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1W0z96MFEoH.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems*, volume 15, 2002. URL https://papers.nips.cc/paper_files/paper/2002/hash/653ac11ca60b3e021a8c609c7198acfc-Abstract.html.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, 2020.

Liping Liu and Thomas G. Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://papers.nips.cc/paper_files/paper/2012/hash/aaebdb8bb6b0e73f6c3c54a0ab0c6415-Abstract.html.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3698–3707. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1405. URL https://aclanthology.org/D18-1405/.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Vincent Micheli, Eloi Alonso, and François Fleuret. Efficient world models with context-aware tokenization. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=BiWIERWBFX.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations*, 2025.

Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. RLVR-World: Training world models with reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K. Terry. Minari. Zenodo, version 0.5.0, 2024. URL https://doi.org/10.5281/zenodo.13767625.

A Proofs for Fixed Banks

A.1 Formal Conventions and the Cell Decomposition

We use the notation of Section 3 and make the bank explicit where needed: $S_{C,w}$ is the cell of the winner and A_C the valid region of bank C. $\Delta^\circ(\mathcal{Y})$ denotes the laws on $\mathcal{Y}$ with full support. Softmax ignores a common shift of the logits, so we normalize $\sum_i z_i = 0$; uniqueness and stationarity statements refer to this contrast space.

Win probabilities. Because C is independent of Y given H, a distinct candidate v wins a valid query with probability $p(S_{C,v})/p(A_C) = Q_p(v)$. The channel loss $-\log Q_r(y_w)$ is therefore the negative log-likelihood of the observed winner under the model, and it is correctly specified. When valid queries have positive probability, averaging over contexts and banks gives the excess risk

$$\mathcal{L}_{\mathrm{ch}}(r) - \mathcal{L}_{\mathrm{ch}}(p) = \frac{\mathbb{E}\big[p(A_C)\, D_{\mathrm{KL}}(Q_p \,\|\, Q_r)\big]}{\mathbb{E}[p(A_C)]}, \tag{10}$$

which vanishes exactly when $Q_r = Q_p$ on every bank that can produce a valid query.

Ties and empty batches. A query is invalid when two distinct candidates tie for nearest or when all candidates coincide, as in our IRIS and Delta-IRIS implementations (Section C.6). RENEW instead keeps the pairs between the first nearest candidate and strictly farther ones; in every analytical construction of this appendix, the two conventions yield the same pairs. With $\epsilon = 0$, a batch without valid pairs contributes zero loss and zero gradient.

Derivation of (6). On a valid query, $y \in S_{C,w} \subseteq A_C$, so $r(y) = r(A_C) \cdot \frac{r(S_{C,w})}{r(A_C)} \cdot \frac{r(y)}{r(S_{C,w})}$. Taking $-\log$ gives (6).

A.2 Full-Support Multiset Statement and Consequences

Consider a deterministic candidate multiset containing $m_i \geq 1$ copies of every support outcome, repeated in each context. Put $K = \sum_i m_i$ and $v_i = K - m_i$. Because the realized outcome is present, its support is the winner and all its duplicate copies tie. It produces m_j valid comparisons against outcome $j \neq i$, for a total of v_i pairs. The winner therefore reveals the realized outcome, even when the fitted loss is distorted.

Theorem A.1 (Finite-batch multiset optimum)**.** *Let $p \in \Delta^\circ(\mathcal{Y})$, $B \geq 1$, and use the full-support multiset and tie-excluded loss of (4). For $V_{-1} = \sum_{b=2}^{B} v_{Y_b}$, with independent $Y_b \sim p$, define*

$$h_i^{(B)} = \mathbb{E}\left[(v_i + V_{-1} + \epsilon)^{-1}\right]. \tag{11}$$

The exact normalized pair weights and unique population optimum are

$$A_{ij} = B p_i m_j h_i^{(B)}, \qquad r_i^{*,B} = \frac{p_i h_i^{(B)}/m_i}{\sum_\ell p_\ell h_\ell^{(B)}/m_\ell}. \tag{12}$$

For $n \geq 3$, every finite B and every $\epsilon \geq 0$,

$$r^{*,B} = p \quad \Longleftrightarrow \quad m_1 = \cdots = m_n. \tag{13}$$

For $n = 2$, the same equivalence holds if $B > 1$ or $\epsilon > 0$. When $n = 2, B = 1, \epsilon = 0$, every positive multiplicity pair recovers p.

The result describes a distortion of probabilities despite complete observation of the winner. For $n \geq 3$, outcomes with more copies have smaller ratios of fitted to true probability than outcomes with fewer copies. The random normalization modifies this distortion through $h_i^{(B)}$; it does not generally remove it. The equivalence for three or more outcomes holds at zero offset, so it does not depend on the small numerical constant in the denominator.

Batch size changes the exact incorrect law. For one-context batches,

$$r_i^{*,1} \propto \frac{p_i}{m_i(K - m_i + \epsilon)}. \tag{14}$$

As $B \to \infty$, bounded convergence gives $Bh_i^{(B)} \to 1/\sum_j p_j v_j$, and

$$r_i^{*,B} \longrightarrow \frac{p_i/m_i}{\sum_j p_j/m_j}. \tag{15}$$

This inverse-multiplicity limit is the familiar density-ratio distortion of uncorrected negative sampling (Gutmann & Hyvärinen, 2012; Ma & Collins, 2018). The factor $h_i^{(B)}$ gives the exact adjustment for our finite-batch, tie-excluded normalization rule.

A.3 Exact Score and the Unique Conditional Optimum

Averaging the batch loss (4) over outcomes and banks drawn from a fixed law ν gives the expected pairwise loss

$$\mathcal{L}_{\mathrm{BT}}(z; p, \nu) = \sum_{i \neq j} A_{ij} \log\bigl(1 + e^{z_j - z_i}\bigr), \qquad A_{ij}(p, \nu) = \mathbb{E}\left[\frac{\sum_b N_{bij}}{\sum_b V_b + \epsilon}\right], \tag{16}$$

where A_{ij} is the expected share of comparisons won by y_i against y_j.

Proposition A.2 (Exact score balance)**.** *For fixed p and ν, the gradient of $\mathcal{L}_{\mathrm{BT}}$ in (16) with respect to z_i is*

$$g_i(r; p, \nu) = \sum_{j \neq i} \frac{A_{ji} r_i - A_{ij} r_j}{r_i + r_j}. \tag{17}$$

If the candidate co-occurrence graph is connected, $\mathcal{L}_{\mathrm{BT}}$ has a unique interior minimizer, the solution of $g(r) = 0$, and this minimizer equals p if and only if $g(p; p, \nu) = 0$.

Each term of (17) vanishes when $r_i/r_j = A_{ij}/A_{ji}$. When the bank is sampled from the model being fitted, the implemented update holds the sampled bank fixed and follows the semi-gradient

$$G(z) := \nabla_{z'} \mathcal{L}_{\mathrm{BT}}(z'; p, \nu_r)\big|_{z'=z} = g(r; p, \nu_r), \qquad r = \mathrm{softmax}(z), \tag{18}$$

so its zeros satisfy (17) with weights evaluated at the model's own bank law.

The support, candidate tuple length and batch size are finite. Consequently the expectation of the normalized loss is a finite weighted sum of smooth pair losses. It can be differentiated term by term. For winner i and loser j,

$$\nabla_z \log(1 + e^{z_j - z_i}) = \frac{r_j}{r_i + r_j}(e_j - e_i).$$

Combining the two orientations of each pair proves (17). The score sums to zero across coordinates; the contrast constraint therefore introduces no nonzero constant score at a stationary point.

Let the graph have an edge for each pair that can co-occur among candidates. If $\{i, j\}$ is an edge, presenting both and drawing $Y = i$ has positive probability because $p_i > 0$ and candidates are independent of Y. A copy of i then wins at distance zero and every copy of j is a strict loser. Thus $A_{ij} > 0$. Reversing the realized outcome gives $A_{ji} > 0$. For an interior fitted law,

$$\nabla^2 \mathcal{L}_{\mathrm{BT}} = \sum_{i<j} (A_{ij} + A_{ji}) \frac{r_i r_j}{(r_i + r_j)^2} (e_i - e_j)(e_i - e_j)^\top.$$

On a connected graph this quadratic form is positive for every nonzero contrast vector: a zero value would require equal coordinates on every edge, then everywhere, and their sum is zero. The objective is strictly convex on the contrast space.

Existence follows from coercivity. A contrast vector with diverging norm has diverging coordinate range. A path of length at most $n - 1$ connecting a maximum and a minimum contains an edge difference at least the range divided by $n - 1$. Both directed weights on every graph edge are positive, and there are finitely many edges, so their minimum $a_{\min}$ is positive. Since $\log(1 + e^t) \geq \max(0, t)$, the objective is at least $a_{\min}$ times that edge difference. It diverges with the range. A continuous coercive function attains its minimum, and strict convexity makes the minimum unique. This proves Proposition A.2.

Equation (18) evaluates this conditional score with the current bank law held fixed. It omits derivatives of that law by definition. A theorem about the conditional Hessian therefore does not supply a convergence theorem for repeated model-dependent resampling.

A.4 Finite-Batch Pair Weights

For a deterministic full-support multiset, the realized winner support is Y_b and the batch loss is

$$L_B(r; Y_{1:B}) = \frac{\sum_{b=1}^{B} \sum_{j \neq Y_b} m_j \ell_{Y_b j}(r)}{\sum_{b=1}^{B} (K - m_{Y_b}) + \epsilon}, \qquad \ell_{ij}(r) = -\log \frac{r_i}{r_i + r_j}.$$

Every context has at least one valid pair, so the denominator is positive even at $\epsilon = 0$. Isolating the expected coefficient of ℓ_{ij} gives

$$\begin{aligned} A_{ij} &= \sum_{b=1}^{B} \mathbb{E}\left[\frac{\mathbf{1}\{Y_b = i\} m_j}{\sum_{c=1}^{B} (K - m_{Y_c}) + \epsilon} \right] \\ &= B p_i m_j \mathbb{E}\left[\frac{1}{K - m_i + V_{-1} + \epsilon} \right]. \end{aligned}$$

The second equality uses exchangeability, followed by conditioning on the distinguished outcome. The other $B - 1$ valid counts keep their original independent distribution. No expected-count approximation appears.

For an unordered pair, set $q_{ij} = A_{ij}/(A_{ij} + A_{ji})$. Its expected contribution is $(A_{ij} + A_{ji})\,\mathrm{CE}(q_{ij}, r_i/(r_i + r_j))$. Subtracting its entropy minimum gives

$$(A_{ij} + A_{ji}) D_{\mathrm{KL}}\left(\mathrm{Bern}(q_{ij}) \,\|\, \mathrm{Bern}(r_i/(r_i + r_j))\right) \geq 0.$$

Equality holds exactly when the fitted pair probability is q_{ij}. The weights satisfy

$$\frac{q_{ij}}{1 - q_{ij}} = \frac{p_i h_i^{(B)}/m_i}{p_j h_j^{(B)}/m_j}.$$

The normalized law in (12) therefore matches every pair. Any minimizer must match all pairs, since all weights are positive, and their ratios plus normalization determine that same law. This proves the optimum statement.

A.5 Recovery of the True Law and Batch-Size Limits

Positive true masses imply $r^{*,B} = p$ exactly when the factors $h_i^{(B)}/m_i$ are equal. For $m_i < m_j$, the denominator difference is

$$(m_j - m_i)(K + V_{-1} + \epsilon - m_i - m_j).$$

For $n \geq 3$, at least one further multiplicity is positive, so this expression is strictly positive for every value of the common other-context count V_{-1}. Reciprocals reverse the inequality and expectation preserves it:

$$\mathbb{E}\frac{1}{m_i(K - m_i + V_{-1} + \epsilon)} > \mathbb{E}\frac{1}{m_j(K - m_j + V_{-1} + \epsilon)}.$$

Thus unequal multiplicities cannot produce equal factors. Equal multiplicities make all valid counts identical and do produce equal factors.

For $n = 2$, the final factor becomes $V_{-1} + \epsilon$. It is positive if $\epsilon > 0$. It is also positive if $B > 1$, since each other context has a positive valid count. If $B = 1$ and $\epsilon = 0$, however, both multiplicity denominators are $m_1 m_2$; the optimum then equals p for unequal counts as well. This proves every case in Theorem A.1.

For $B = 1$, substituting $V_{-1} = 0$ gives (14). For the large-batch limit, define $\bar{v} = \sum_i p_i v_i$ and $v_{\min} = \min_i v_i > 0$. By the law of large numbers,

$$\frac{B}{v_i + V_{-1} + \epsilon} \longrightarrow \frac{1}{\bar{v}} \quad \text{almost surely.}$$

The left side is bounded by $1/v_{\min}$ for all B, since its denominator contains B valid counts, each at least $v_{\min}$. Dominated convergence gives $B h_i^{(B)} \to 1/\bar{v}$. Substitution in (12) proves (15).

Finally, a given law p^0 is the unique optimum exactly when it equals (12), equivalently when p_i^0/p_i is proportional to $h_i^{(B)}/m_i$ for all i.

A.6 The Identifying Feedback Family and Control Consequence

An incorrect transition law can change the selected action. The following construction gives the probability and reward conditions.

An identifying feedback family. Take $\mathcal{Y} = \{-1, 0, 1\}$ with absolute distance and true law $p(a) = (a, 1-2a, a)$, where $0 < a < 1/2$. Independently queried adjacent pairs produce comparison probabilities

$$q_{-0} = a, \qquad q_{0+} = 1 - a. \tag{19}$$

More generally, these observations identify a law through $p_- = q_{-0}$, $p_0 = q_{0+} - q_{-0}$ and $p_+ = 1 - q_{0+}$. For any strictly positive fixed allocation to both edges, the temperature-one BT objective has the unique population optimum

$$r^\star(a) = \frac{1}{1+a}(a, 1-a, a). \tag{20}$$

Each edge's cross-entropy is minimized by matching its observed win probability; the resulting ratios and normalization give (20). The likelihood of the actual channel instead has unique optimum $p(a)$. If the model starts from a source law equal to $r^\star(a)$, BT training keeps it despite identifying feedback; the control task of Section C.5 uses such a source law for its warm starts. For $a < 1/3$, both laws rank the middle outcome above either endpoint.

A parameterized control task. Use states $s_{-1}, s_0, s_{+1}, s_\dagger$. At s_0, the risky action gives reward $R_{\text{risk}} > 0$ and moves to s_{-1}, s_0, s_{+1} according to the modeled law. The safe action gives $R_{\text{safe}} > 0$ and enters $s_\dagger$. The other states are absorbing with zero subsequent rewards. Let $\gamma \in (0, 1)$ be the discount, t the true probability of returning to s_0, and ρ its modeled probability. The stationary risky and safe values under continuation probability s are

$$V_{\text{risk}}(s) = \frac{R_{\text{risk}}}{1 - \gamma s}, \qquad V_{\text{safe}} = R_{\text{safe}}. \tag{21}$$

Choose rewards such that the threshold

$$\kappa = \frac{1 - R_{\text{risk}}/R_{\text{safe}}}{\gamma} \quad \text{satisfies} \quad 0 < \kappa < 1. \tag{22}$$

The Bellman-optimal action is risky when the modeled continuation probability exceeds κ, and safe when it is below κ.

Proposition A.3 (Decision error from an overestimated continuation law)**.** *If $t < \kappa < \rho$, the true optimal action is safe while the model selects risky. The regret of deploying the model's stationary policy under the true law is*

$$\Delta(p, r) = V_p^*(s_0) - V_p^{\pi_r}(s_0) = R_{\text{safe}} - \frac{R_{\text{risk}}}{1 - \gamma t} > 0. \tag{23}$$

For the adjacent-feedback family, $t = 1 - 2a$ and $\rho = (1-a)/(1+a)$. Their discrepancy is

$$\rho - t = \frac{2a^2}{1+a} > 0. \tag{24}$$

Hence, for any admissible a, γ and positive R_{risk}, the open reward interval

$$\frac{R_{\text{risk}}}{1 - \gamma t} < R_{\text{safe}} < \frac{R_{\text{risk}}}{1 - \gamma \rho} \tag{25}$$

produces a strict wrong-action decision.

The interval is nonempty because risky value increases with continuation probability. The derivation below establishes the Bellman condition and identification claims.

Derivation. On $\mathcal{Y} = \{-1, 0, 1\}$, the left candidate in pair $(-1, 0)$ wins exactly when $Y = -1$. The left candidate in pair $(0, 1)$ wins when $Y = -1$ or $Y = 0$. Neither adjacent pair ties on this support. Consequently, $q_{-0} = p_-$ and $q_{0+} = p_- + p_0$. Normalization gives the inverse stated above, establishing identification of the full law.

Fix strictly positive query weights w_{-0}, w_{0+} summing to one. The BT population loss is $\sum_e w_e \, \mathrm{CE}(q_e, b_e(r))$. Subtracting each edge's entropy minimum leaves a nonnegative Bernoulli KL divergence. Thus any law matching both observed probabilities attains the global minimum, and every minimizer must match them. For the symmetric true law $p(a)$, the matching ratios are

$$\frac{r_-}{r_0} = \frac{a}{1-a}, \qquad \frac{r_0}{r_+} = \frac{1-a}{a}.$$

They imply $r_- = r_+$ and, after normalization, the law in (20). The channel likelihood instead matches both win probabilities, $r_- = q_{-0}$ and $r_- + r_0 = q_{0+}$, so its unique optimum is p. A source law equal to $r^\star(a)$ is therefore kept by BT and corrected by the channel likelihood. The center exceeds either endpoint in both laws when $a < 1/3$.

In the control task, the risky action gives reward R_{risk} whenever the process visits s_0. Summing the discounted geometric series gives $V_{\mathrm{risk}}(s) = R_{\mathrm{risk}}/(1-\gamma s)$; the safe action gives R_{safe} once. The Bellman equation is

$$V^*(s_0) = \max\{R_{\mathrm{safe}}, R_{\mathrm{risk}} + \gamma s V^*(s_0)\}.$$

When $s < \kappa$, the safe value satisfies this equation because $R_{\mathrm{risk}} + \gamma s R_{\mathrm{safe}} < R_{\mathrm{safe}}$. When $s > \kappa$, the risky value satisfies it and is strictly larger than the safe value. Discounted contraction makes either solution unique in its strict regime. Equality at $s = \kappa$ makes both actions optimal. Thus the threshold orders Bellman-optimal actions.

Under $t < \kappa < \rho$, the model's stationary optimal policy chooses risky at every revisit to s_0, while the true optimum chooses safe. Evaluating the former under the true continuation law gives the regret in (23). For the identifying family,

$$\frac{1-a}{1+a} - (1-2a) = \frac{2a^2}{1+a} > 0.$$

The positive reward interval in (25) is therefore nonempty. Its lower bound makes the safe action truly preferable; its upper bound makes the same action appear inferior under the fitted model. The interval also implies $0 < \kappa < 1$.

For fixed κ, the inequalities $1 - 2a < \kappa < (1-a)/(1+a)$ are equivalent to

$$\frac{1-\kappa}{2} < a < \frac{1-\kappa}{1+\kappa},$$

intersected with $0 < a < 1/2$. Restricting further to $a < 1/3$ preserves the ordering of outcomes. These are parameter conditions for the control effect, rather than instantiated experimental values.

A.7 Ordinal Reversals near Every Interior Truth

Use $\mathcal{Y} = \{0, 1, 2\}$ and squared distance. Bank $C_L = \{0, 1\}$ reveals whether $Y = 0$; bank $C_R = \{1, 2\}$ reveals whether $Y = 2$. Let $\Pr(C_L) = \lambda \in (0, 1)$ and draw the bank independently of Y. Every query is valid. The two singleton probabilities identify p_0, p_2, and normalization gives p_1. With $\mathrm{kl}(a\|b)$ denoting Bernoulli KL,

$$\mathcal{L}_{\mathrm{ch}}(q) - \mathcal{L}_{\mathrm{ch}}(p) = \lambda \, \mathrm{kl}(p_0\|q_0) + (1-\lambda)\, \mathrm{kl}(p_2\|q_2).$$

This risk has the unique optimum $q = p$. Its ordering of imperfect laws need not agree with predictive log loss. The distinction is a specialization of proper-scoring-rule geometry (Gneiting & Raftery, 2007).

Proof of Proposition 4.3. Use coordinates (q_0, q_2), with $q_1 = 1 - q_0 - q_2$. Both risk gradients vanish at p. Their Hessians there are

$$M_{\mathrm{pred}} = \nabla^2 \mathcal{L}_{\mathrm{pred}}(p) = \begin{pmatrix} 1/p_0 + 1/p_1 & 1/p_1 \\ 1/p_1 & 1/p_2 + 1/p_1 \end{pmatrix},$$

$$M_{\mathrm{ch}} = \nabla^2 \mathcal{L}_{\mathrm{ch}}(p) = \begin{pmatrix} \lambda/[p_0(1-p_0)] & 0 \\ 0 & (1-\lambda)/[p_2(1-p_2)] \end{pmatrix}.$$

Both are positive definite. They cannot be proportional because M_{pred} has positive off-diagonal entries and M_{ch} is diagonal. A symmetric 2×2 matrix with equal eigenvalues is scalar, while $M_{\text{pred}}^{-1/2} M_{\text{ch}} M_{\text{pred}}^{-1/2} = \mu I$ would imply $M_{\text{ch}} = \mu M_{\text{pred}}$. Thus this positive definite matrix has distinct eigenvalues $0 < \mu_- < \mu_+$. Let ϕ_-, ϕ_+ be corresponding generalized eigendirections, normalized by $\phi_-^{\mathsf{T}} M_{\text{pred}} \phi_- = \phi_+^{\mathsf{T}} M_{\text{pred}} \phi_+ = 1$. Choose ξ with $1 < \xi^2 < \mu_+/\mu_-$ and set $r = p + t\xi\phi_-$, $s = p + t\phi_+$ in these coordinates. For sufficiently small positive t, both laws remain interior and

$$\mathcal{L}_{\text{pred}}(r) - \mathcal{L}_{\text{pred}}(s) = \tfrac{1}{2} t^2 (\xi^2 - 1) + o(t^2) > 0,$$
$$\mathcal{L}_{\text{ch}}(r) - \mathcal{L}_{\text{ch}}(s) = \tfrac{1}{2} t^2 (\xi^2 \mu_- - \mu_+) + o(t^2) < 0.$$

Taking t arbitrarily small gives the claim. Strict inequalities persist under sufficiently small perturbations of the pair and of the interior p, λ. □

The result holds for every interior truth and positive bank mix in this geometry, establishing open sets of reversed pairs.

Exact finite reversal. An exact non-infinitesimal instance uses equal bank weights, $p = (1/3, 1/3, 1/3)$, $r = (5/12, 1/6, 5/12)$ and $s = (11/24, 1/3, 5/24)$, giving

$$\mathcal{L}_{\text{pred}}(r) - \mathcal{L}_{\text{pred}}(s) = \tfrac{1}{3} \log \tfrac{11}{10} > 0, \quad \mathcal{L}_{\text{ch}}(r) - \mathcal{L}_{\text{ch}}(s) = \tfrac{1}{6} \log \tfrac{3355495}{3841600} < 0.$$

A.8 Partial-Support Derivatives and Normalization

Fix each realized query's winner i_b, retained losing codes and multiplicities. Give the contexts independent logits z_b and set $D = \sum_b \sum_{j \text{ retained}} m_{bj}$. Differentiating

$$L = \frac{\sum_b \sum_{j \text{ retained}} m_{bj} \log(1 + e^{z_{bj} - z_{bi_b}})}{D + \epsilon}$$

gives, for each retained loser j,

$$\frac{\partial L}{\partial z_{bj}} = \frac{m_{bj}}{D + \epsilon} \frac{r_{bj}}{r_{bi_b} + r_{bj}}. \tag{26}$$

The winner derivative is the negative sum of its losing derivatives; absent or excluded coordinates have zero direct derivative. The all-invalid $\epsilon = 0$ convention gives zero loss and gradient. Samples, labels, masks and counts are held fixed during differentiation.

For two count assignments, marked (1) and (2), at identical logits and fixed winners and masks, the downward force on a retained losing coordinate is larger under (1) exactly when

$$\frac{m_{bj}^{(1)}}{D_1 + \epsilon} > \frac{m_{bj}^{(2)}}{D_2 + \epsilon}.$$

At fixed denominator it is linear in the count. With one query and one distinct loser, $D = m_{bj}$ and the count cancels when $\epsilon = 0$. Thus an unconditional strict count-to-force claim would be false. Ratios of two retained losing forces within a query cancel the shared denominator, but the total contribution of that query still depends on it.

The assigned-count contrast of Section 5.3 changes relative loser weighting and induced query weighting together. Its post-fit response is an observed effect of that combined intervention. The full-support stationary theorem additionally requires every outcome to be present, and shared neural parameters couple updates across queries. Neither the stationary law nor the sign of a multi-update neural change follows from this local identity.

B Proofs for Model-Sampled Candidates

B.1 Sampling Law, Ties and the Exact Finite-Batch Field

The support is $\mathcal{Y} = \{-1, 0, 1\}$ with absolute distance. Index its points by $\xi_- = -1$, $\xi_0 = 0$ and $\xi_+ = 1$. For each of $B \geq 1$ independent contexts, draw $K \geq 2$ candidate indices IID from an interior

model law $r = \mathrm{softmax}(z)$ and an independent outcome index $Y \sim p$. Distances are evaluated between the corresponding points. Rescaling the support multiplies all distances by the same positive factor and preserves every winner and tie mask. The logits are finite and the temperature is one. Winners and pairs follow Definition 3.1 and Eq. (4): on a valid query the unique nearest candidate wins and is compared with every other candidate slot, excluding its own copies. The summed retained losses are divided by the batch's total valid-pair count plus $\epsilon \geq 0$. At $\epsilon = 0$, all-invalid batches have loss and gradient zero; this is also their continuous extension from positive offsets. The semi-gradient holds candidate sampling, selection, masks and denominator fixed.

Write a candidate multiset as $m = (l, m_0, h)$ and let

$$\mathcal{M}_K = \{(l, m_0, h) \in \mathbb{Z}^3_{\geq 0} : l + m_0 + h = K\}, \qquad \pi_r(m) = \frac{K!}{l! m_0! h!} r_-^l r_0^{m_0} r_+^h.$$

For a context with at least one valid comparison, denote its winning support point by $w(m, y)$ and its strictly more distant candidate supports by

$$D_s(m, y) = \{j : m_j > 0,\ |\xi_j - \xi_y| > |\xi_{w(m,y)} - \xi_y|\}.$$

Its valid-pair count and unnormalized semi-gradient are

$$\begin{aligned} V(m, y) &= \sum_{j \in D_s(m,y)} m_j, \\ T(m, y; r) &= \sum_{j \in D_s(m,y)} m_j \frac{r_j}{r_{w(m,y)} + r_j} (e_j - e_{w(m,y)}). \end{aligned} \tag{27}$$

When there is no valid comparison, set $V = T = 0$ without requiring a support-level winner. Equation (27) follows from

$$\nabla_z \log(1 + \exp(z_j - z_w)) = \frac{r_j}{r_w + r_j}(e_j - e_w).$$

The same derivative applies when scores are log probabilities because their shared log-softmax normalizer cancels.

Candidate order cannot change T on this support. When the realized outcome is either endpoint, distinct support points have distinct distances, so any winning tie consists of copies of one support point. For $Y = 0$, a sampled center is the unique winning support point if present. If the center is absent, the query is invalid under Definition 3.1: either one distinct candidate remains or both endpoints tie at distance 1. RENEW's first-index rule would mask all its comparisons as well, so the tie convention changes no pair, and the counts determine V and T completely.

Let (V_b, T_b) be the independent context contributions. The realized batch semi-gradient is

$$g_B = \frac{\sum_{b=1}^B T_b}{\epsilon + \sum_{b=1}^B V_b}.$$

Let V_{-1} be the sum of the valid counts in $B - 1$ independent other contexts. For positive integers t, define

$$h_B(t; r, p) = \mathbb{E}\left[\frac{1}{t + V_{-1} + \epsilon}\right].$$

Exchangeability followed by conditioning on one context gives

$$G_B(r; p) = B \sum_{m \in \mathcal{M}_K} \pi_r(m) \sum_{\substack{y \in \{-,0,+\} \\ V(m,y) > 0}} p_y h_B(V(m, y); r, p) T(m, y; r). \tag{28}$$

The other-context count distribution is specified exactly by

$$\Pr(V = t) = \sum_{m \in \mathcal{M}_K} \pi_r(m) \sum_y p_y \mathbf{1}\{V(m, y) = t\}, \tag{29}$$

and its $(B - 1)$-fold convolution. For $B = 1$, $V_{-1} = 0$. All sums and supports are finite, so the conditioning and interchange of expectation with summation are valid. Equation (28) retains the expected reciprocal of the actual denominator. An interior law is stationary exactly when its right side is zero. The model probabilities enter the bank-law weights π_r, but those weights are held fixed by the semi-gradient definition.

B.2 THE SYMMETRIC FINITE-K CONDITION

Take $B = 1$, $r = (x, c, x)$ with $c = 1 - 2x$ and $0 < x < 1/2$, and $p = (a, 1 - 2a, a)$ with $0 < a < 1/2$. Denote the sampled counts by (L, M, H). There are three kinds of nonzero center-coordinate contribution.

When $Y = 0$ and $M > 0$, the center wins against $K - M$ endpoint copies and contributes

$$-\frac{x}{1-x}\frac{K-M}{K-M+\epsilon}.$$

When the realized outcome is the left endpoint and $L > 0$, the left endpoint wins. Its M center losers contribute

$$\frac{c}{1-x}\frac{M}{K-L+\epsilon}.$$

Right-endpoint losers have zero center coordinate but remain in the denominator. When the realized outcome is the left endpoint, $L = 0$ and $M > 0$, the center is the nearest available support point and contributes

$$-\frac{x}{1-x}\frac{H}{H+\epsilon}.$$

For a realized right endpoint, reflection exchanges L and H while preserving their joint distribution, so its expected center contribution equals that for a realized left endpoint. Configurations without a valid loser contribute zero.

Define

$$\begin{aligned} A_s &= \mathbb{E}\left[\frac{M\mathbf{1}\{L>0\}}{K-L+\epsilon}\right], \\ Q_s &= \mathbb{E}\left[\frac{(K-M)\mathbf{1}\{M>0\}}{K-M+\epsilon}\right], \\ C_s &= \mathbb{E}\left[\frac{H\mathbf{1}\{L=0,M>0\}}{H+\epsilon}\right]. \end{aligned} \tag{30}$$

At zero offset, zero-valid-count terms use the stipulated zero contribution. Averaging the preceding cases gives

$$G_0 = \frac{2acA_s - (1-2a)xQ_s - 2axC_s}{1-x}. \tag{31}$$

Reflection symmetry implies $G_- = G_+$. Each pair gradient sums to zero, implying $G_- = G_+ = -G_0/2$. Thus $G_0 = 0$ is stationarity in the full three-logit model at the symmetric law.

For the finite sum

$$U_{K,\epsilon}(q) = \sum_{j=1}^{K-1}\binom{K}{j}q^j(1-q)^{K-j}\frac{j}{j+\epsilon},$$

multinomial conditioning evaluates all three helpers. First, $K - L \sim \mathrm{Bin}(K, 1-x)$ and $\mathbb{E}[M \mid L] = (K-L)c/(1-x)$. The event $L > 0$ excludes $K - L = K$, while $K - L = 0$ contributes zero, giving

$$A_s = \frac{c}{1-x}U_{K,\epsilon}(1-x).$$

Second, $K - M \sim \mathrm{Bin}(K, 2x)$, and $M > 0$ excludes its upper endpoint, giving

$$Q_s = U_{K,\epsilon}(2x).$$

Third, $\Pr(L = 0) = (1-x)^K$ and $H \mid (L = 0) \sim \mathrm{Bin}(K, x/(1-x))$. The condition $M > 0$ excludes $H = K$, giving

$$C_s = (1-x)^K U_{K,\epsilon}\left(\frac{x}{1-x}\right).$$

These expressions retain both the missing-support events and their differing valid counts.

The event $(L, M, H) = (1, K-1, 0)$ has positive probability and makes the integrand of A_s positive, so $A_s > 0$. On $\{L = 0, M > 0\}$ the integrands of Q_s and C_s coincide; elsewhere the former is

nonnegative and the latter is zero. Hence $Q_s \geq C_s$, and $cA_s + x(Q_s - C_s) > 0$. Rearranging (31) yields

$$G_0 = \frac{2\{cA_s + x(Q_s - C_s)\}}{1-x}\left(a - \frac{xQ_s}{2\{cA_s + x(Q_s - C_s)\}}\right).$$

This proves the general-K equation (33) of Section B.4 and its positive prefactor. The true law is stationary exactly when $a = a_{K,\epsilon}(a)$. If the curve's value lies outside $(0, 1/2)$, it does not specify an admissible true law in this family.

B.3 Two Sampled Candidates at Every Finite Batch Size

Proof of Theorem 4.2. For $K = 2$, the helper expressions reduce to

$$A_s = \frac{2xc}{1+\epsilon}, \qquad Q_s = \frac{4xc}{1+\epsilon}, \qquad C_s = \frac{2xc}{1+\epsilon}.$$

Substitution into (31) gives

$$G_{1,0}(x; a) = \frac{4xc}{(1-x)(1+\epsilon)}\{a(1-x) - x\}.$$

The prefactor is positive for $0 < x < 1/2$. The expression in braces has exactly one zero, $x = a/(1+a)$, which lies in $(0, 1/2)$ for $0 < a < 1/2$.

For a general finite batch, every valid context has precisely one retained pair. An adjacent unordered candidate pair occurs with total probability $4xc$ and never has a distance tie. The unordered endpoint pair occurs with probability $2x^2$ and is valid precisely when the realized outcome is an endpoint, an event of probability $2a$. Therefore the context valid count is Bernoulli with probability

$$\pi_{\mathrm{v}} = 4xc + 4ax^2.$$

The other $B-1$ counts sum to $V_{-1} \sim \mathrm{Bin}(B-1, \pi_{\mathrm{v}})$. Every nonzero context term in (28) has valid count one, so the same conditional factor applies to all of them:

$$G_{B,0}(x; a) = \frac{4Bxc}{1-x}\mathbb{E}\left[\frac{1}{1+V_{-1}+\epsilon}\right]\{a(1-x) - x\}. \tag{32}$$

The expectation is strictly positive. Its dependence on p and r changes no zero of the remaining factor. The unique symmetric stationary point is therefore $r^*(a) = (a, 1-a, a)/(1+a)$ for every finite B. At truth, $x = a$ makes the last factor equal $-a^2 < 0$. The true law is consequently not stationary. A given symmetric source law is stationary if and only if it equals this stationary point. □

B.4 General-K Stationarity Law and a Numerical Check

The general-K stationarity equation. For general K and a single-context batch (Section 4.1), write $U_{K,\epsilon}(q) = \sum_{j=1}^{K-1}\binom{K}{j}q^j(1-q)^{K-j}\frac{j}{j+\epsilon}$, $A_s = \frac{c}{1-x}U_{K,\epsilon}(1-x)$, $Q_s = U_{K,\epsilon}(2x)$ and $C_s = (1-x)^K U_{K,\epsilon}\left(\frac{x}{1-x}\right)$. The law r is stationary exactly when

$$a = a_{K,\epsilon}(x) := \frac{xQ_s}{2\{cA_s + x(Q_s - C_s)\}}. \tag{33}$$

Evaluating (33) at $\epsilon = 0$ on the grid $a \in [0.01, 0.49]$ shows that the endpoint mass is under-estimated for every a when $K \leq 5$, whereas for $K \geq 6$ it is over-estimated for small a and under-estimated for large a. For $K \geq 6$, the truth is therefore stationary only at an isolated endpoint mass.

Numerical check. We run the exact implemented update of Section 4.1 numerically: support $\{-1, 0, 1\}$ with absolute distance, truth $p = (a, 1-2a, a)$, logits $z \in \mathbb{R}^3$ with $\sum_i z_i = 0$, K candidates drawn IID from $\mathrm{softmax}(z)$ in each of B contexts, nearest-candidate winners with tied candidates excluded, the loss (4) with $\epsilon = 0$, and the semi-gradient (18). Plain SGD runs for 20,000 steps (step size 0.05, $B = 16$) or 40,000 steps (step size 0.02, $B = 1$); we report the time-average of the endpoint mass over the second half, for three seeds.

K	a	analytic $x^\star$ (33)	SGD, $B = 1$	SGD, $B = 16$
2	0.1 / 0.2 / 0.3	0.091 / 0.167 / 0.231	–	0.089 / 0.166 / 0.229
3	0.1 / 0.2 / 0.3	0.092 / 0.172 / 0.245	0.086 / 0.171 / 0.244	0.122 / 0.202 / 0.265
8	0.1 / 0.2 / 0.3	0.112 / 0.217 / 0.301	0.110 / 0.216 / 0.298	0.186 / 0.257 / 0.311

For $K = 2$ the numerical endpoints match Theorem 4.2 at $B = 16$, as the theorem predicts for every batch size. For $K \geq 3$, single-context runs match (33) up to step-size noise (sample standard deviations over seeds 0.002–0.005), whereas sharing one valid-pair denominator across 16 contexts moves the stationary point substantially, in the direction of larger endpoint mass.

C Experimental Methods

C.1 IRIS Models, Data and Prediction Protocol

Public sources and model. We use the released IRIS Breakout checkpoint (Micheli et al., 2023) and Farama Minari's recorded Breakout expert dataset (Younis et al., 2024).

The published tokenizer uses a 512-entry codebook and produces a 4×4 code grid from a 64×64 RGB frame. The original causal transformer has ten layers, four attention heads, width 256, a 17-token observation/action block and a maximum of 20 blocks; embedding, residual and attention dropout are 0.1. We train all transformer layers, positional embeddings, observation/action embeddings and the original observation head: 8,314,624 parameters. The tokenizer's 4,027,523 parameters and the world model's 132,869 reward/end-head parameters remain frozen. The actor is unused, and the auxiliary LPIPS module is omitted while loading all tokenizer prediction weights.

Recorded observations and split. The dataset contains ten episodes and 23,075 recorded transitions, with 210×160 RGB observations and four discrete actions. Its Atari configuration uses frame skip four and no sticky actions. Episodes 0–6 form the adaptation split, episode 7 supplies validation, and episodes 8–9 supply test data. Using four within-episode history frames gives 17,675 training, 1,239 validation and 4,131 test windows; the two test episodes contribute 1,880 and 2,251 windows respectively. Terminal next observations are retained, and windows never cross episode boundaries.

Each frame is resized once with PIL bilinear interpolation to 64×64, converted to channels-first values in $[0, 1]$, and encoded through the frozen tokenizer after its standard $[-1, 1]$ preprocessing. Each input contains four 16-code frames with their actions, ending with the recorded current action, followed by the first 15 true codes of the target frame. The resulting length-83 input predicts all sixteen target codes with the public observation-head alignment. The first target code is predicted from the current action position; later scores use the same true target prefix across conditions.

Frozen feedback and fitted objectives. For every conditional target-token context, eight candidate codes are drawn IID with replacement from the original published model. The banks are generated once, with seeds 24100, 24200 and 24300 for training, validation and test, and then shared across all fitting seeds and conditions. The oracle labels the uniquely nearest distinct candidate to the target code under squared Euclidean distance between frozen codebook embeddings. Distances are computed in float64. A query is excluded when distinct nearest codes tie or when all candidates are identical; duplicate copies of the uniquely winning code remain valid.

For a candidate multiset C and observed winner w, let $S_{C,w}$ contain vocabulary outcomes for which w is the unique nearest distinct candidate, and let A_C be the union of all such winner cells. Given logits s, the channel loss on a valid query is

$$\ell_C(s; C, w) = \log \sum_{y \in A_C} e^{s_y} - \log \sum_{y \in S_{C,w}} e^{s_y}.$$

The sums cover the full 512-code vocabulary and count each outcome once. BT compares the winner with every losing candidate slot using temperature-one logit differences, retaining loser multiplicities. Its minibatch loss averages valid pairs, whereas channel likelihood averages valid winner queries; BT uses count offset 10^{-8}, while channel uses the valid-query mean. The supervised baseline receives every true target code, including codes whose comparison query is excluded.

The test bank has 66,096 token queries, of which 42,515 are valid, yielding 189,263 winner-loser pairs. Token NLL averages all target tokens; channel loss averages valid winner queries. The respective denominators are retained throughout the reported metrics.

Fitting and evaluation. Seeds 24101–24103 each start from the same published weights and complete three epochs with batch size 16. Training uses AdamW at learning rate 10^{-4}, weight decay 0.01 with the public IRIS parameter groups, and gradient norm clipping at 10. Within a seed, conditions share minibatch order and initialization. Each trajectory completes 3,315 optimizer updates, with no skipped all-invalid batches.

The primary checkpoint is the end of epoch three. All epoch checkpoints are retained. Minimum validation token NLL selects a secondary checkpoint, with the earliest epoch breaking exact ties; it selects epoch zero for BT, epoch one for channel, and epoch three for supervised training in every seed. Own-objective validation selects epoch three for BT, epoch two for channel, and epoch three for supervised training. These selections use validation data, and the primary tables retain the complete fixed-budget outcome.

The complete study used an NVIDIA A100-SXM4-40GB, PyTorch 2.6.0 with CUDA 12.6, torchvision 0.21.0, NumPy 1.26.4, Pillow 12.3.0 and h5py 3.16.0. Model fitting and scoring use float32 with TF32 disabled; frozen codebook distances use float64.

C.2 Multiplicity and Normalization Interventions

The intervention uses the released IRIS Breakout model, original prepared data and original frozen candidate banks. Five arms are fit at seeds 24101–24103: Sampled BT, balanced or de-duplicated BT, complementary views A and B, and the channel likelihood. Each resets to the same published checkpoint and completes three epochs of the original AdamW schedule, with paired minibatch order and seeded training stochasticity within a seed. The tokenizer and reward/end heads remain frozen; the transformer, embeddings and observation head receive 3,315 updates per fit.

Fixed information and manipulated counts. Sampled BT retains the original eight slots. For U distinct candidates, sort their code IDs and assign four copies each in the balanced view. View A gives consecutive pairs multiplicities $(1, 7)$ and view B gives $(7, 1)$; an unpaired last code receives four copies. Every transformed view has $4U$ slots and the same distinct support. Assignment uses no target or winner label. The original winner and distinct-code tie mask are retained, and winning copies are excluded from pairs.

New BT arms use a zero count offset and the actual minibatch valid-pair mean. Balanced four-copy and de-duplicated one-copy objectives are identical, so they are one arm. Channel likelihood is invariant to these repetitions and is fit once per seed. There are fifteen fits. The nine-fit study in Appendix C.1 uses the original 10^{-8} offset.

Identified contrast. Changing counts changes both relative loser weights inside a query and the query's share of the batch loss through its valid-pair count. The intervention measures their combined algorithmic effect while retaining observed winner information. It does not apportion the effect between these components or establish the full-support stationary law for a shared neural model.

Prespecified endpoints. The common epoch-three budget is primary. All epochs and per-query arrays are retained; validation-based selections are secondary, and test values select nothing. For population P, the fraction of positive Sampled BT excess removed by balancing is

$$F_P = \frac{\mathrm{NLL}_{\mathrm{sampled},P} - \mathrm{NLL}_{\mathrm{balanced},P}}{\mathrm{NLL}_{\mathrm{sampled},P} - \mathrm{NLL}_{\mathrm{original},P}}.$$

A nonpositive denominator makes this fraction undefined. Both all-token and covered-token fractions were specified before execution, with a one-half dominance threshold. The covered fraction describes the deterioration more directly: uncovered tokens improve under both BT arms, so their gain reduces the all-token denominator. Both fractions and all raw losses remain reported.

C.3 DELTA-IRIS MODEL, DATA AND SELECTION PROTOCOLS

Released model and inputs. We use the released Crafter 5M Delta-IRIS checkpoint (Micheli et al., 2024) with the public implementation. Its contextual/action-dependent tokenizer emits four codes from a 1,024-entry, 64-dimensional codebook. The world model uses a three-layer, eight-head, width-512 transformer and its frame encoder. The tokenizer, policy and reward/end heads remain frozen; 11,291,304 world-model parameters train.

Crafter (Hafner, 2022) provides 64×64 RGB observations and 17 actions. Experiments use Python 3.12, PyTorch 2.6.0/CUDA 12.6, NumPy 1.26.4, Crafter 1.8.3 and Gym 0.23.1. We evaluate adaptation of the released checkpoint rather than reproduction of its original training run.

Training and held-out cohorts. Twelve first episodes from constructor seeds 38100–38111 form eight training and four validation worlds, split by list position. The frozen released policy acts with temperature one, epsilon zero, initial five-zero-frame burn-in and policy seed 38190. The returned terminal next observation is retained.

The fixed-budget protocol is specified before generating four new test worlds at constructor seeds 38300–38303, with policy seed 38390 and bank seed 38391. All are disjoint from the training and validation world identities. They are held out from this adaptation study; public pretraining histories are unknown. Three new fitting seeds, 38201–38203, each fit all five original arms for 500 updates from the released weights. All original training/validation episodes, tokens, indices, banks and codebook distances are shared across arms. Test episodes have lengths 437/852/371/822 and provide 2,406 complete windows; the returns of these collection episodes only describe the data.

Alignment and losses. Each stride-one window contains 21 observed frames and 20 transitions inside one episode. The last transition supplies the four scored delta tokens; the final dummy block is unscored. Training and evaluation use the same positions and common true prefixes. The original sequence construction aligns predictions with these tokens. Only observation-token losses enter comparison fitting; reward/end losses are excluded.

Each test context receives one frozen bank of eight temperature-one replacement samples from the original model. The geometric oracle uses squared Euclidean distance between frozen codewords; exact distinct-winner ties and all-identical banks are invalid. BT uses zero-offset batch valid-pair means, including the original candidate-only complementary 1/7 versus 7/1 views and equal four-copy view. Channel uses the channel likelihood (5) with its valid-query mean. AdamW uses the original parameter groups, learning rate 10^{-4}, weight decay 0.01, gradient clipping at 10 and batch size 32. All arms share within-seed uniform window permutations. Each fit sees 15,925 windows in 500 batches, with no invalid-batch skips; checkpoint data at updates 100/250/500 are retained. Validation selects nothing; test is evaluated only at 500.

Prospective selection and test separation. The original released model, eight training and four validation worlds, cached tokens, banks, metric, losses, optimizer groups and batch order construction remain unchanged. New fitting seeds 38401–38403 each train all five arms for 500 updates, saving states and evaluating validation every 50 updates. Each fit chooses the earliest exact minimizer of its own implemented held-out feedback objective over these ten fitted checkpoints. The objective averages the implemented batch ratios across the fixed sequential 32-window validation batches, retaining all-invalid zeros and equally weighting the last short batch. Neither target NLL nor test scores enter selection. Update zero is the original model and is excluded from this selection rule.

All fifteen checkpoint choices are fixed before generating the new test cohort. Four prospectively listed constructor seeds 38500–38503, policy seed 38590 and bank seed 38591 create new worlds and query banks from the original model. Data collection, five-zero-frame initial burn-in, last-transition four-token scoring and true prefixes match the earlier protocol. Episode lengths are 480/963/769/1006, yielding 461/944/750/987 windows. These worlds are held out from this study, with no claim about public pretraining. Both selected and 500-update states are scored on this one cohort. None selects 500, so there are thirty distinct test evaluations.

Planning with adapted checkpoints. For every checkpoint of the selection study, an agent plans with that checkpoint as its world model and acts in the four new test worlds. At every step, it samples

64 action sequences of length 10 from the released policy. It scores each sequence by the reward that the checkpoint predicts with the frozen reward head and executes the first action of the best sequence. Each checkpoint and fitting seed runs two episodes in every test world with fixed environment seeds. Return is the mean episode return divided by that of the original model under the same planner, worlds and seeds. Bands and error bars show the standard deviation over the three fitting seeds.

C.4 Matched Parameter-Block Intervention

Comparison likelihood constrains probability at the resolution of an observed winner cell. Target likelihood also depends on allocation inside that cell. We test how this distinction interacts with the trainable parameters in a prospective 2×2 factorial: conditional comparison likelihood versus target cross entropy, and full observation-model adaptation versus final-affine adaptation. Both objectives use the same valid positions. Seeds 24101–24103 share the original IRIS Breakout checkpoint, data and candidate banks; each fit receives 3,315 updates. The head-progress criterion requires a relative calibration improvement of at least 5%.

Dropout is disabled in all arms. Features enter the last affine map after the earlier observation-head linear transformation and ReLU. The full block also updates that earlier map, the transformer, embeddings and positional parameters. This matches valid-position supervision across the factorial, unlike the earlier dropout-enabled, all-token target baseline.

The full block has 8,314,624 trainable parameters and the final-affine block has 131,584. Both use the IRIS AdamW recipe above and valid-position target or channel losses. Calibration-only progress matching chooses paired endpoints without consulting test losses.

C.5 Identifying Finite Control

The finite control instantiates Appendix A.6. Warm starts are unsmoothed empirical fits to source-world samples; both warm and cold initializations are evaluated. Each objective selects by its own validation loss from the fixed training grid. BT and the channel likelihood share adjacent-edge labels, support and metric; supervised fitting observes the outcome. BT-model labels are drawn separately from the BT model, while ordinal BT repeatedly presents the unique ordering facts.

Protocol item	Fixed value
Outcomes	$(-1, 0, +1)$
Source law	$(1/6,\ 2/3,\ 1/6)$
True law	$(1/5,\ 3/5,\ 1/5)$
Distance edges	$(-1, 0)$ and $(0, +1)$
Left-candidate cells	$\{-1\}$ and $\{-1, 0\}$
Temperature	1.0
Discount and rewards	$\gamma = 0.9$, risky step reward 1, safe reward $7/3$
Action threshold	0.6349206349
Risky-policy exact regret	0.159420
Seeds	23101, 23102, 23103
Data per seed	8192 source, 4096 train, 4096 validation outcomes
Per-edge split	2048 independent observations per adjacent edge
Policy evaluation	4096 rollouts per policy, horizon 200
Optimizer and checkpoints	Adam, learning rate 0.02, batch size 256; updates 0, 64, 256, 1024, 4096
Precision	JAX float32 training; NumPy float64 data generation and exact evaluation

Table 3: Configuration of the control task. Non-start states are absorbing.

The implemented fixed-graph losses average over the selected examples. The five objectives are selected by their own validation losses among the saved checkpoints. BT and the channel likelihood share the same adjacent-edge distance observations and the known support and metric. Supervised fitting observes full outcome indices. BT-model labels come from a separate labeler whose answers follow the BT model, with adjacent probabilities $(1/4, 3/4)$ and 4096 additional labels per split. Ordinal BT repeatedly presents the two ordering facts. Closed-form solutions serve as diagnostics rather than additional trained models.

C.6 Relation to RENEW

RENEW samples candidate trajectories from the model being trained, keeps duplicate candidate slots and picks the winner by the smallest summed L1 error. Its pair loss excludes slots tied with the winner and averages valid pairs with an offset. Its queries target contexts with high epistemic uncertainty. Our analysis isolates a categorical transition law under a fixed bank law. Our IRIS and Delta-IRIS oracle uses squared codebook distance and, like Definition 3.1, rejects a query when distinct candidates tie for nearest. Both tie rules give the same pairs in every construction of Section A. Minari observations lack the intermediate frames used by the frame max-pooling of IRIS, so the original Atari models start off their training distribution.

D Supplementary Results

D.1 Checkpoint Selection and Covered Targets across World Models

Table 4: **Selecting Sampled BT by feedback overshoots the prediction optimum.** The feedback pick never comes before the NLL pick. Sampled BT picks later wherever prediction is best early in training and pays for it in test NLL, whereas the channel likelihood overshoots less often and at a smaller cost. Both picks may be the original model.

	Objective	Breakout	Pong	Seaquest	Crafter
Overshoot (of 3 seeds)	Sampled BT	3/3	3/3	0/3	3/3
	Channel	3/3	2/3	0/3	0/3
Test NLL cost (nats)	Sampled BT	1.45	0.19	0.00	0.90
	Channel	0.10	0.11	0.00	0.00

D.2 Multiplicity and Normalization across Games

The assigned-count interventions preserve candidate identities and winner information while changing pair and query weights. Breakout, Pong and Seaquest meet the covered-excess removal criterion and retain the predicted sign in every episode–seed panel. Whole-stream improvements in Pong and Seaquest coexist with covered-target costs; the whole-stream positive-excess fractions are therefore undefined.

Episode	Seed	F_{covered}	F_{all}	Sign agreement	Signed change
all	24101	0.694876	0.916537	0.705171	0.562881
8	24101	0.687410	0.908724	0.710254	0.614434
9	24101	0.701277	0.923157	0.700928	0.519844
all	24102	0.675965	0.917498	0.700826	0.571396
8	24102	0.676169	0.916272	0.706434	0.614521
9	24102	0.675777	0.918643	0.696144	0.535395
all	24103	0.690595	0.897056	0.706672	0.577660
8	24103	0.670693	0.873310	0.711123	0.609156
9	24103	0.708035	0.917838	0.702957	0.551367

Table 5: Complete primary contrasts on Breakout. Fractions are unitless and signed log-probability changes are in nats. Every episode–seed covered fraction exceeds the prespecified one-half threshold. Pooled rows summarize these same episodes.

Seed / condition	All	Covered	Uncovered	First token
Original	4.677897	0.441039	10.297220	0.045006
24101 Sampled BT	1.966351	0.988135	3.263754	0.049112
24101 Balanced	1.738378	0.579095	3.275928	0.052108
24101 View A	1.974682	0.812338	3.516292	0.036102
24101 View B	1.881481	0.718734	3.423625	0.074259
24101 Channel	1.564081	0.553233	2.904764	0.041918
24102 Sampled BT	1.937834	0.962776	3.231049	0.039782
24102 Balanced	1.715647	0.533336	3.283740	0.030732
24102 View A	1.910238	0.745632	3.454848	0.053003
24102 View B	1.804009	0.644794	3.341471	0.040646
24102 Channel	1.452764	0.527261	2.680254	0.035477
24103 Sampled BT	1.943046	0.907992	3.315834	0.087929
24103 Balanced	1.743284	0.529623	3.352956	0.063809
24103 View A	1.924675	0.705992	3.541007	0.078391
24103 View B	1.917013	0.645531	3.603373	0.087353
24103 Channel	1.530420	0.511469	2.881851	0.086743

Table 6: Pong token NLL at the common epoch 3. The original checkpoint is evaluated once. All five arms and all three seeds are retained.

Seed / condition	All	Covered	Uncovered	First token
Original	3.516263	1.513252	5.310612	3.802239
24101 Sampled BT	2.976213	2.691006	3.231708	3.500189
24101 Balanced	2.407832	1.950258	2.817738	3.001852
24101 View A	2.632032	2.181377	3.035740	3.055605
24101 View B	2.539139	2.080581	2.949927	3.249180
24101 Channel	1.948011	1.456634	2.388200	2.446873
24102 Sampled BT	3.011413	2.701249	3.289265	3.147781
24102 Balanced	2.404389	1.940896	2.819597	2.610119
24102 View A	2.624985	2.158315	3.043040	2.831755
24102 View B	2.526320	2.051128	2.952010	2.760429
24102 Channel	1.997132	1.536389	2.409877	2.319056
24103 Sampled BT	2.986781	2.699272	3.244339	3.063293
24103 Balanced	2.423751	1.988589	2.813580	2.602587
24103 View A	2.666599	2.221337	3.065475	2.863256
24103 View B	2.589080	2.152686	2.980012	2.790172
24103 Channel	1.957945	1.465397	2.399182	2.289324

Table 7: Seaquest token NLL at the common epoch 3. The original checkpoint is evaluated once. All five arms and all three seeds are retained.

D.3 Selected and Fixed-Budget Delta-IRIS Predictions

The fixed-budget confirmation and the selection study use different held-out cohorts. The selected comparison meets the covered-excess removal threshold, whereas both fixed-budget comparisons fail it. All multiplicity-direction panels pass. These outcomes remain separate: a favorable relative fraction does not restore prediction to the original model.

In the selection study, all BT arms select update 150 and channel selects update 50 from the prespecified fitted-checkpoint grid. The original model is excluded from that selector, although it has the best channel-validation score. Figure 6 shows the validation trajectories; the tables below retain the test endpoints and their distinct denominators.

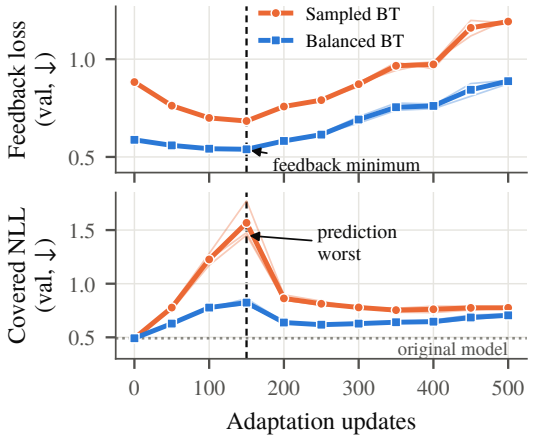

Figure 6: **The feedback minimum is the covered-NLL maximum.** Validation curves of the two BT objectives for Delta-IRIS on Crafter (thin: seeds; thick: mean). The update with the lowest feedback loss (dashed, update 150) has the highest covered-target NLL in every BT fit.

Seed	Covered gap	Covered excess	Fraction	Pooled sign fraction
38201	0.123434	0.373001	0.330922	0.740044
38202	0.080388	0.338204	0.237690	0.727323
38203	0.096713	0.344942	0.280375	0.733960

Table 8: All covered primary fractions are below 0.5. Pooled sign fractions are descriptive; every episode–seed direction panel passes the prespecified majority criterion.

Seed / endpoint	Sampled	Balanced	Gap	Excess	Fraction
38401 Selected	1.515633	0.825506	0.690127	1.029819	0.670144
38402 Selected	1.493336	0.820869	0.672467	1.007522	0.667446
38403 Selected	1.821076	0.887635	0.933441	1.335262	0.699069
38401 500	0.792827	0.708819	0.084008	0.307014	0.273630
38402 500	0.797600	0.710885	0.086715	0.311786	0.278123
38403 500	0.784548	0.718860	0.065689	0.298735	0.219890

Table 9: Every covered fraction, with numerator and denominator. Original covered NLL is 0.485814. The selected primary passes $F > 0.5$ for all three seeds; the fixed-500 secondary fails all three. All/covered/uncovered counts are 12,568/10,200/2,368; matched valid counts are 7,290/5,119/2,171.

Seed / method	All	Covered	Uncovered
Original	1.456932	0.562511	4.357885
38201 Sampled	2.190149	0.935512	6.259421
38201 Balanced	2.166701	0.812078	6.560265
38201 View A	2.202165	0.807138	6.726776
38201 View B	2.141395	0.842452	6.354371
38201 Channel	2.253724	0.840257	6.838142
38202 Sampled	2.140635	0.900716	6.162174
38202 Balanced	2.184966	0.820328	6.611015
38202 View A	2.195933	0.813728	6.678955
38202 View B	2.183235	0.847394	6.515882
38202 Channel	2.345038	0.854993	7.177829
38203 Sampled	2.178011	0.907453	6.298923
38203 Balanced	2.161420	0.810740	6.542196
38203 View A	2.193656	0.822888	6.639586
38203 View B	2.148130	0.825915	6.436583
38203 Channel	2.261377	0.824353	6.922200

Table 10: Delta-IRIS fixed-budget test fits at 500 updates. Denominators are 9,624 all tokens, 7,356 covered and 2,268 uncovered; matched valid counts are 6,040/3,937/2,103.

Seed / method	All	Covered	Uncovered
Original	1.195885	0.485814	4.254471
38401 Sampled	2.008366	1.515633	4.130781
38401 Balanced	1.423687	0.825506	4.000313
38401 View A	1.482686	0.834102	4.276421
38401 View B	1.503035	0.940271	3.927105
38401 Channel	1.234305	0.560566	4.136392
38402 Sampled	1.996941	1.493336	4.166183
38402 Balanced	1.421412	0.820869	4.008208
38402 View A	1.471907	0.816904	4.293288
38402 View B	1.498827	0.935852	3.923805
38402 Channel	1.246658	0.581227	4.112956
38403 Sampled	2.282747	1.821076	4.271365
38403 Balanced	1.470127	0.887635	3.979170
38403 View A	1.482362	0.839812	4.250107
38403 View B	1.575953	1.035165	3.905362
38403 Channel	1.243249	0.568136	4.151253

Table 11: Selected endpoints on the same new cohort. Token scores use all tokens in each population; decomposition terms retain their separate common valid mask. The original model's row is identical in both tables.

Seed / method	All	Covered	Uncovered
Original	1.195885	0.485814	4.254471
38401 Sampled	1.749636	0.792827	5.871024
38401 Balanced	1.752015	0.708819	6.245511
38401 View A	1.751251	0.699127	6.283204
38401 View B	1.746969	0.736949	6.097561
38401 Channel	1.856237	0.746874	6.634741
38402 Sampled	1.782098	0.797600	6.022755
38402 Balanced	1.773698	0.710885	6.351693
38402 View A	1.824165	0.721808	6.572495
38402 View B	1.731939	0.743855	5.988043
38402 Channel	1.838867	0.708533	6.707702
38403 Sampled	1.724783	0.784548	5.774780
38403 Balanced	1.780883	0.718860	6.355479
38403 View A	1.844368	0.754239	6.540028
38403 View B	1.755676	0.744642	6.110635
38403 Channel	1.851984	0.715425	6.747637

Table 12: Fixed 500 endpoints on the same new cohort. Token scores use all tokens in each population; decomposition terms retain their separate common valid mask. The original model's row is identical in both tables.

D.4 Finite-Control Decisions and Uncertainty

Across both initializations and every seed, BT selects the risky action while channel likelihood selects the optimal safe action. The same safe decision is obtained with supervised fitting or BT-model labels; ordinal feedback selects risky.

Table 13: Finite-control decisions across all three seeds and both warm/cold initializations. Regret is evaluated under the known true transition law.

Objective	Selected action	Exact regret
BT	Risky	0.159420
Channel likelihood	Safe	0
Supervised	Safe	0
BT-model labels	Safe	0
Ordinal BT	Risky	0.159420