FROM COMPACT SUMMARIES TO USABLE MEMORY: A CALIBRATION LAW FOR FINITE-STATE COMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

When a policy reads its previous action, choosing an action also writes the code it will read next. With K freely writable, exactly echoed action codes and no other persistent state, this is a K-state memory register. What does the accuracy of separately optimized K-label history summaries guarantee about the best online machine at the same state budget? We test each summary using the actual suffix, for independent finite observations and a binary terminal target. For every finite $K \geq 2$, allowing finite input alphabets to vary with K, the worst optimal online error under cut error at most ε is $\Theta(\min\{N\varepsilon, 1\})$, uniformly in $N \geq 3$ and $0 \leq \varepsilon \leq 1/4$. Thus a class-wide final-error guarantee $0 < \delta \leq 1/4$ requires cut accuracy on the scale δ/N. A same-state construction and delayed-query lower tasks establish the matching law.

1 INTRODUCTION

Some reinforcement-learning policies receive their previous action with the current observation (Rudin et al., 2022). At time t, such a policy reads a_{t-1} and selects a_t. If that choice is echoed exactly, selecting a_t writes the value it will read at time $t+1$. A cue seen once can be kept by selecting and repeating its action code (Ni et al., 2023). With K freely writable codes and no other persistent state, the read–write rule $a_t = \pi_t(o_t, a_{t-1})$ is a K-state finite-state machine (FSM). Finite-state controllers and action-based memory writes are established (Meuleau et al., 1999; Peshkin et al., 1999); here the previous action fixes the memory budget at exactly K states.

This read–write view suggests a role for a compact history representation: put its K-valued summary in the next action code. We can evaluate a summary offline by rereading the full prefix at each cut and testing whether its label predicts the final answer once future observations arrive. The policy must instead update its code from the old code and the new observation. If every cut admits a good summary, how accurately can the best same-K policy answer at the end, and how does the required cut accuracy change with the horizon?

One bit, two things to remember. Suppose the feedback register has two codes. Consider independent three-bit blocks (A_i, B_i, C_i), read one bit at a time. Count one toggle when $A_i = 1$ and another when both B_i and C_i are one. The final answer is whether the total count is odd, its *parity*. After B_i, exact updating needs both the parity so far and whether $B_i = 1$ until C_i arrives: four possibilities for a two-code register. When simultaneous ones are rare, ignoring the unfinished event costs little at one cut, but this conflict recurs in every block.

At a cut, an offline encoder can reread the past and send completed parity, including A_i if observed. A decoder tests this bit using the actual remaining observations (the *suffix*); it can ignore the one unfinished B_iC_i event at small risk. At the next cut, the encoder rereads the prefix and recovers that event. The FSM sees each observation only once. This is the smallest instance of a conflict at any finite memory budget: for K states, a later query can distinguish more than K combinations of the current answer and a pending feature. The online state must merge some of them, while each offline cut summary can recompute the answer from its full prefix. Section 3 turns this picture into a uniform bound.

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.

With 100 blocks of independent Bernoulli(0.1) bits, every cut admits a one-bit summary with at most 1% error, yet the best two-state machine errs about 43% at the end. Figure 1 shows why rereading the prefix and updating one stored code differ.

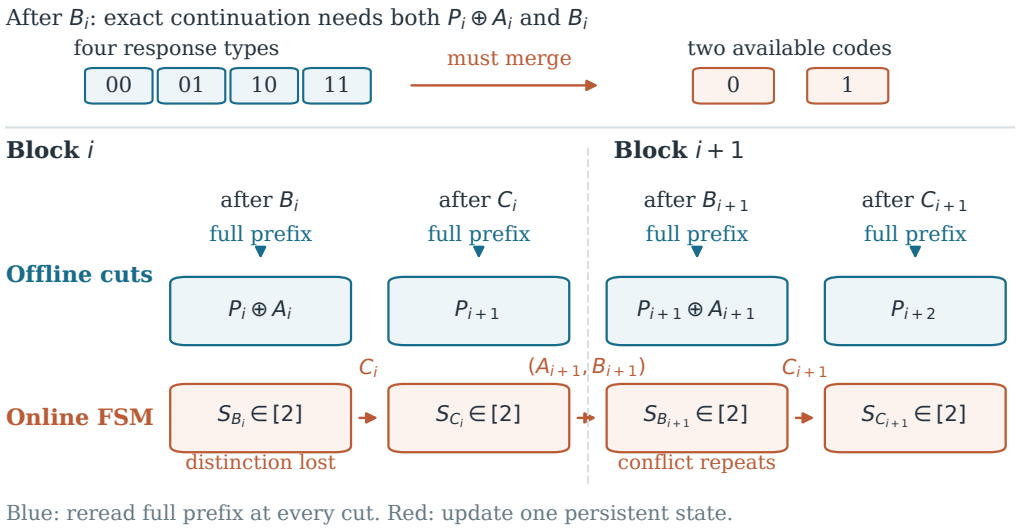

Figure 1: Four suffix-response types compete for two online states after B_i: completed parity and the pending bit can each be 0 or 1. Offline summaries reread the full prefix at each cut; the online path updates one persistent code across two blocks. P_i is parity before block i.

Two uses of the same memory. The example compares rereading the past with updating a stored value. At time t, the observed history is the *prefix* $X_{\le t} = (X_1, \dots, X_t)$. A *prefix encoder* reads that history and writes one of K summary values, or labels. A K-state FSM instead updates its old state S_{t-1} using the new observation X_t:

$$S_t = u_t(S_{t-1}, X_t). \tag{1}$$

Time is free, so updates may depend on t. All input-dependent information retained across steps must fit in these K values.

How accurate must the summary be? For independent finite observations and a binary final answer (the *terminal target*), choose the best summary and decoder separately at each cut. Their probability of a wrong answer is the *optimal cut error*. The summary is a message from the past to a receiver holding the future; an FSM must maintain one state process across all cuts. Recursive state representations are well established (Littman et al., 2001; Subramanian et al., 2022). We ask what this particular summary test guarantees for the best machine with the same memory budget.

Figure 1 isolates the distinction. For a sequence of N observations (the *horizon*), the *calibration profile* $\Gamma_{N,K}(\varepsilon)$ takes the best FSM error for each task, then the worst value across tasks with optimal error at most ε at every cut. We establish, for every finite $K \ge 2$,

$$\Gamma_{N,K}(\varepsilon) = \Theta(\min\{N\varepsilon, 1\}), \qquad N \ge 3, \quad 0 \le \varepsilon \le \tfrac{1}{4}, \tag{2}$$

with constants independent of K, horizon and tolerance. Thus even vanishing cut error can coexist with persistent execution error as the stream grows. At zero cut error execution is exact; the gap concerns *approximate* summaries. To guarantee final error $0 < \delta \le 1/4$ throughout this task class, the cut-error tolerance must have order δ/N.

What the law certifies. Previous-action feedback fixes one concrete K-code execution budget. Approximate information-state theory connects predictive state quality to decision quality using

reward and next-state prediction conditions (Subramanian et al., 2022). Our cut score asks a complementary question: what does average terminal prediction by separately optimized K-label summaries guarantee for one executable K-state updater? The profile answers this question in both directions, including tasks on which even the best updater pays the matching cost.

For exogenous observations and a terminal correctness reward, the same prediction error is an undiscounted return gap. We return to the legal memory-write conditions for previous-action feedback in Section 6.

Where the calibration adds information. The communication tools behind the proof are established: posterior recompression converts suitable cut messages into an online state (Woodruff & Yang, 2019; Du et al., 2023), and same-width exclusive-OR (XOR) composition propagates local conflicts (Goldreich, 2010). Earlier conversion results use different premises, including symmetry or amplified worst-case message accuracy. Here the cut premise is *average* terminal error under the original product law, optimized separately at each cut; the conclusion uses the *same* K states and bounds the best executable machine. The matching tasks show exactly when this average cut accuracy is sufficient as the horizon grows. Section 7 compares the resource and error conditions in detail.

2 Measuring summary accuracy and execution error

We now formalize the two procedures in the example: summarizing a history and updating a state. Both use the same memory size. Writes are freely allowed here; physical action restrictions return in Section 6. Let observation X_t take values in a finite alphabet $\mathcal{X}_t$. The sequence $X = (X_1, \dots, X_N)$ follows a product distribution $\mu = \bigotimes_{t=1}^{N} \mu_t$, so the observations are independent, with $N \geq 2$. Write $X_{>t} = (X_{t+1}, \dots, X_N)$ for the suffix, $\mathcal{X}_{a:b} = \prod_{i=a}^{b} \mathcal{X}_i$, and use labels $[K] = \{0, \dots, K-1\}$ for an integer $K \geq 1$. Let $f : \prod_t \mathcal{X}_t \to \{0, 1\}$ be a deterministic terminal target. At a cut $1 \leq t < N$, a prefix encoder $e_t : \mathcal{X}_{1:t} \to [K]$ inspects the entire prefix. A decoder $d_t : [K] \times \mathcal{X}_{t+1:N} \to \{0, 1\}$ predicts f using its label and the entire suffix. The decoder tests whether the summary retains enough of the past to answer correctly once the remaining observations are known. It is used only for evaluation; the controller observes the suffix later, one observation at a time. With $\mathbb{P}$ denoting probability, define

$$E_{\mathrm{cut},t}(K) = \min_{e_t, d_t} \mathbb{P}\{d_t(e_t(X_{\leq t}), X_{>t}) \neq f(X)\}, \qquad E_{\mathrm{cut}}(K) = \max_{t<N} E_{\mathrm{cut},t}(K). \tag{3}$$

Each cut is optimized separately, so the encoders for different t need not admit compatible recursive updates. Thus $E_{\mathrm{cut}}(K)$ measures the quality of standalone prefix representations, not yet the quality of one executable state process.

An executing controller starts from a fixed state and updates $S_t \sim T_t(\cdot \mid S_{t-1}, X_t)$, where T_t is a transition distribution and $S_t \in [K]$. It then outputs $b(S_N)$ using a binary readout $b : [K] \to \{0, 1\}$. The *optimal execution error* is $E_{\mathrm{seq}}(K) = \min_{(T_t), b} \mathbb{P}\{b(S_N) \neq f(X)\}$, where probability includes inputs and fresh runtime draws. The sole persistent input-dependent object is the K-valued state. The time index ("phase"), fixed program and within-step computation are free; retained history, belief vectors, caches and input-bearing random state count toward the memory budget. Fresh independent random draws are permitted. This is a state-cardinality constraint, not a bound on program length or computation time.

Finite domains ensure the optima are attained. Fixing the independent random draws converts a randomized controller into a deterministic time-dependent one, so randomization cannot improve the optimal error under μ. Simulating an optimal deterministic machine up to a cut, sending its state, and then simulating its suffix proves $E_{\mathrm{cut}}(K) \leq E_{\mathrm{seq}}(K)$.

Let $\mathcal{T}_N$ be the class of all such length-N product-source Boolean tasks, allowing their finite alphabets and distributions to vary with the task. For a task $\mathcal{T}$, write $E_{\mathrm{cut},\mathcal{T}}(K)$ and $E_{\mathrm{seq},\mathcal{T}}(K)$ for the two optimal errors above. Define

$$\Gamma_{N,K}(\varepsilon) = \sup_{\mathcal{T} \in \mathcal{T}_N :\, E_{\mathrm{cut},\mathcal{T}}(K) \leq \varepsilon} E_{\mathrm{seq},\mathcal{T}}(K). \tag{4}$$

For each task, first choose its best FSM; then take the worst error across admissible tasks. Each task may use a different FSM. The comparison fixes the number of states while allowing the finite observation alphabet to vary. The next section quantifies how far the two optimal errors can differ.

3 THE CALIBRATION LAW: LONGER TASKS NEED MORE ACCURATE SUMMARIES

If every cut admits a K-valued summary with error at most ε, how much larger can the best recursive error be? The following law answers for every finite state budget.

Theorem 1 (Finite-memory calibration profile)**.** *For every integer $K \geq 2$, $N \geq 3$ and $0 \leq \varepsilon \leq 1/4$, with $m = \lfloor N/3 \rfloor$,*

$$\frac{1-(1-\varepsilon/2)^m}{2} \leq \Gamma_{N,K}(\varepsilon) \leq \min\left\{\frac{1}{2},\ 2(N-1)\varepsilon\right\}. \tag{5}$$

Consequently, with constants uniform in K, N and ε,

$$\frac{1}{22}\min\{N\varepsilon, 1\} \leq \Gamma_{N,K}(\varepsilon) \leq 2\min\{N\varepsilon, 1\}. \tag{6}$$

Section 4 proves the same-capacity upper bound and introduces the lower-bound obstruction; Appendix A proves the general K lower bound. The sharper two-state example is worked out in Appendix B.

The uniform constants use the finite-alphabet task class in (4). Its lower-bound block reads a q-valued vector as one observation, with $K \leq q < 2K$; thus its input alphabet grows with K. When every observation is instead one independently sampled bit, a separate construction charges $L = 2\lceil \log_2 K \rceil + 2$ observations per block and gives, for $N \geq L$,

$$\Gamma^{\mathrm{bin}}_{N,K}(\varepsilon) \geq \frac{1-(1-\varepsilon/2)^{\lfloor N/L \rfloor}}{2}. \tag{7}$$

Here Γ^{bin} restricts (4) to independent binary observations. Proposition 11 proves this bound. For fixed K it still gives a linear horizon cost; when K varies with N, these bounds do not close the binary-input dependence on K.

The bound has two regimes. When $N\varepsilon \ll 1$, the worst optimal execution error is of order $N\varepsilon$. When $N\varepsilon$ is constant or larger, a small per-cut error by itself no longer guarantees a small execution error. Thus the meaningful quantity for a guarantee uniform over horizons is $N\varepsilon$, rather than ε alone. The next corollaries turn this into a criterion for vanishing error and a tolerance for a chosen final-error target.

Corollary 2 (Uniform vanishing-error criterion)**.** *For any sequence $(K_j, N_j, \varepsilon_j)$ in the theorem's domain,*

$$\Gamma_{N_j,K_j}(\varepsilon_j) \longrightarrow 0 \quad \Longleftrightarrow \quad N_j\varepsilon_j \longrightarrow 0. \tag{8}$$

Corollary 3 (Converting an accuracy budget)**.** *Fix $K \geq 2$, $N \geq 3$ and a desired error $0 < \delta \leq 1/4$. The condition $\varepsilon \leq \delta/\lceil 2(N-1) \rceil$ is sufficient for $\Gamma_{N,K}(\varepsilon) \leq \delta$. Within $0 \leq \varepsilon \leq 1/4$, that guarantee necessarily requires*

$$\varepsilon \leq 2\big[1-(1-2\delta)^{1/\lfloor N/3 \rfloor}\big] \leq \frac{40\delta}{N}. \tag{9}$$

The sufficient and necessary tolerances have the same order δ/N, with absolute constants over the stated range.

Appendix B.4 derives both corollaries. Figure 2 shows the two uses of the law: bounding final error at a given tolerance and seeing how three vanishing tolerances behave as the horizon grows.

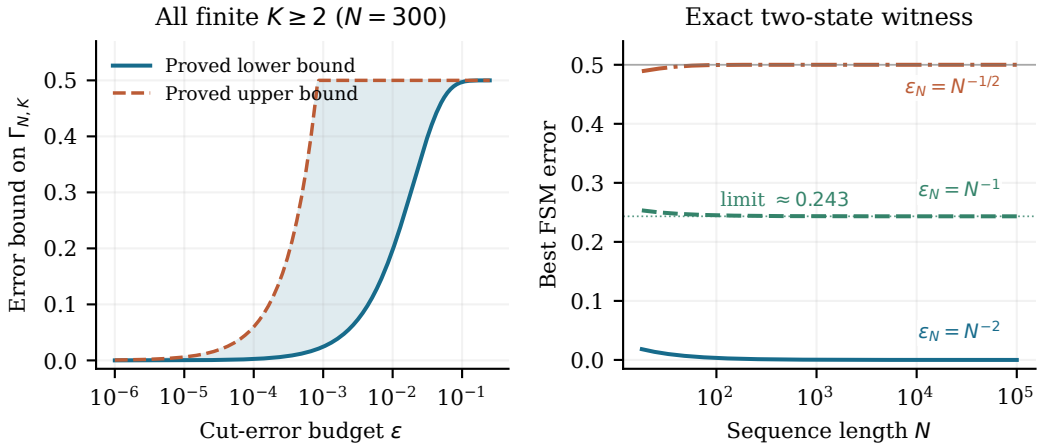

Figure 2: Analytical calibration and an exact witness, not measured errors. Left: proved bounds on the worst best-FSM error $\Gamma_{300,K}(\varepsilon)$ for every finite $K \geq 2$; the exact profile lies in the shaded region. Right: the exact optimal two-state error $[1 - (1 - 2\varepsilon_N)^{\lfloor N/3 \rfloor}]/2$ for the three-bit task introduced above, at whole-block horizons $N = 3m$, with cut error at most $\varepsilon_N = N^{-2}, N^{-1}, N^{-1/2}$. All three cut errors vanish, while execution error tends respectively to 0, $(1 - e^{-2/3})/2$, and $1/2$.

The exact two-state family is particularly sharp: with $N = 300$ and $\varepsilon = 0.01$, every cut has a one-bit summary with error at most 1%, while even the best two-state machine errs with probability $[1 - 0.98^{100}]/2 \approx 43.4\%$.

4 Why the calibration law holds at the same state budget

We must establish two facts: accurate summaries permit a useful updating machine, and some tasks force the horizon cost even for the best machine. The construction answers the first question; the running parity example introduces the second.

4.1 From separate summaries to one updating machine

The summaries at successive cuts need not have compatible updates. We must connect them without retaining the full history or adding states. Supply deterministic encoders e_t and suffix-assisted decoders d_t at all times, including e_N and the final readout $d_N : [K] \to \{0, 1\}$. Let e_0 be constant and initialize S_0 to its value. Take $X_{>N}$ to be empty. Write r_t for each supplied pair's error under the original input law, not the compiled process:

$$r_t = \mathbb{P}\{d_t(e_t(X_{\leq t}), X_{>t}) \neq f(X)\}. \tag{10}$$

To build an update rule, imagine a past compatible with the old summary, append the new observation, and summarize again. Draw that imagined past from the original source conditioned on its summary being s; this conditional distribution is the *posterior*. Averaging over these draws gives fixed transition tables, so no past is stored at runtime. This *posterior compiler* has transitions

$$Q_t(s' \mid s, x) = \mathbb{P}\{e_t(Px) = s' \mid e_{t-1}(P) = s\}, \tag{11}$$

where P has the original prefix distribution and Px appends x to P. Rows whose conditioning event has probability zero are arbitrary. At runtime the controller draws from these fixed tables and retains only its K-valued state. For a length-t prefix P, state s and suffix w, put $p_t(s, w) = \mathbb{P}\{f(Pw) = 1 \mid e_t(P) = s\}$ and $\Delta_t = 2\mathbb{E}[p_t(1 - p_t)]$, where $\mathbb{E}$ averages over the original joint distribution of the summary label and suffix. Here Δ_t is the disagreement probability between two independent prefixes compatible with the same summary state and suffix; it is the error paid when the proof swaps one compatible prefix for another. For transition tables $Q = (Q_t)_t$ and terminal decoder d_N, the execution error probability, also called *risk*, is

$$R_Q(d_N) = \mathbb{P}\{d_N(S_N) \neq f(X)\}. \tag{12}$$

Proposition 4 (Compilation with supplied decoders). *The controller* (11)*, with output* $d_N(S_N)$*, uses* K *persistent states and satisfies*

$$\mathcal{L}(S_t) = \mathcal{L}(e_t(X_{\le t})), \qquad R_Q(d_N) \le r_N + \sum_{t<N} \Delta_t \le r_N + 2\sum_{t<N} r_t. \tag{13}$$

Here $\mathcal{L}$ *denotes the distribution of a random variable. A deterministic time-dependent* K*-state controller with no larger error exists, without necessarily preserving those marginals.*

Proof. Start with a synthetic empty prefix R_0. At step t, append the actual input, forming a synthetic prefix $W_t = R_{t-1}X_t$, and set $S_t = e_t(W_t)$. If $t < N$, replace W_t by a fresh prefix R_t drawn from the original distribution conditional on $e_t(R_t) = S_t$. This proof construction induces exactly (11).

Inductively W_t has the original prefix distribution and is independent of the untouched actual suffix. Independence of observations ensures that posterior replacement preserves this joint product law for the next step. Conditional on the common summary state and suffix, the target bits before and after replacement are independent Bernoulli variables with parameter p_t, so their disagreement is Δ_t. These successive replacements connect the actual input sequence to W_N. A union bound gives an error of at most $\sum_{t<N} \Delta_t$; the final readout adds r_N because W_N has distribution μ. Comparing both target bits with the same supplied decoder output gives $\Delta_t \le 2r_t$. Fixing the independent per-time random draws gives deterministic transition functions with no larger risk. □

State distributions match the standalone encoders at each time, but joint distributions with the actual prefix or across times need not match. To bound the best FSM, choose optimal encoders and decoders at each $t < N$, then set $e_N = f$ with identity readout on labels $0, 1$. Thus $r_N = 0$ at $K \ge 2$. This is an offline choice, not runtime access to the history. Then

$$E_{\mathrm{cut}}(K) \le E_{\mathrm{seq}}(K) \le 2\sum_{t<N} E_{\mathrm{cut},t}(K) \le 2(N-1)E_{\mathrm{cut}}(K). \tag{14}$$

Together with a majority predictor this proves the upper half of Theorem 1. The sum in (13) has a constant multiplier; converting the sum to a maximum introduces the horizon factor. We next ask whether a better controller could always avoid this factor.

4.2 Why the best machine can still pay the horizon cost

In the one-bit example, a cut summary needs to ignore at most one unfinished event. An online machine faces a new unfinished event in every block. For independent $\mathrm{Bernoulli}(r)$ bits (A_i, B_i, C_i), set

$$H = \bigoplus_{i=1}^{m} \big(A_i \oplus (B_i \wedge C_i)\big), \qquad m = \lfloor N/3 \rfloor. \tag{15}$$

Independent padding fills any unused steps.

Lemma 5 (The one-bit illustration). *For* $0 < r \le 1/2$*, this task has* $E_{\mathrm{cut}}(2) \le r^2$ *and*

$$E_{\mathrm{seq}}(2) = \frac{1-(1-2r^2)^m}{2}. \tag{16}$$

Appendix B proves the exact formula: the best one-block two-state error is r^2, and the same-width XOR argument multiplies the best achievable correlation across independent blocks.

For general K, fix $d \ge 1$, $q = 2^d$, and $0 < p \le 1/2$. One block reads three independent observations in order: A is a fair bit, B is uniform on $\mathbb{F}_2^d$, and C is $\perp$ with probability $1-p$ or a uniform nonzero vector with probability p. The block answer is $F = A$ if $C = \perp$ and $F = A \oplus \langle B, C\rangle$ otherwise. The $2q$ equally likely pairs (A, B) leave different response functions of C. Their minimum pairwise distance is $\alpha = pq/[2(q-1)]$. Even if a state's decoder uses an arbitrary Boolean response, a triangle-inequality packing argument gives

$$\frac{\alpha}{2}\left(1 - \frac{K}{2q}\right)_+ \le e_K(q,p) \le \frac{p}{2}, \tag{17}$$

where $e_K(q,p)$ is the best one-block K-state error. The upper bound simply retains A and ignores the rare query.

Proposition 6 (A gap on the same repeated task)**.** *For m independent copies of this block with their answers XORed, every $K \geq 2$ satisfies*

$$E_{\mathrm{cut}}(K) = e_K(q,p), \qquad E_{\mathrm{seq}}(K) \geq \frac{1-(1-2e_K(q,p))^m}{2}. \tag{18}$$

If $2 \leq K < 2q$ and $m \geq 2$, the execution error is strictly larger than the optimal error at every cut. The same task is computed exactly with $2q$ states.

At each cut the offline encoder rereads the completed parity and faces at most one pending query. The best online machine must carry its state across all blocks. Appendix A proves the cut equality even when the encoder uses all earlier history, and applies same-width XOR composition without assuming independent errors by the machine. Taking $q = 2^{\lceil \log_2 K \rceil}$ and $p = 2\varepsilon$ gives $e_K(q,p) \geq p/8$ in (17), proving the lower half of Theorem 1. For this witness, $2q < 4K$ states suffice for zero error (exactly $2K$ when K is a power of two). This recovery is a property of the witness, not a bound for every admissible task.

5 Two supporting distinctions

The profile concerns the best machine for each task. For a *supplied* posterior compiler, Proposition 12 in Appendix C gives a held-out, decomposable error certificate. Its confidence penalty scales with $N/\sqrt{M}$, where M is the number of independent validation episodes, and can be less sample-efficient than direct validation of a fixed executable machine. The appendix quantifies this penalty and also handles transition tables fitted on the same validation data.

Nor is the posterior compiler always the best machine. The three-observation example in Appendix D has cut and optimal execution error hr, but its cut-optimal supplied summaries compile to error $2hr(1-r)$. This is excess risk of this particular compiler, not an additional lower bound on all K-state machines.

6 The previous-action interpretation

The finite-state result also applies to a simple control interface when its memory register can be written through feedback actions.

When prediction error is return loss. Consider an exogenous terminal-decision task in which the agent observes $X_1, \ldots, X_N$, then takes one binary action a and receives

$$R = \mathbf{1}\{a = f(X)\}. \tag{19}$$

A K-state predictor and a K-state controller are then the same object: use the predictor's final readout as the action, or read any terminal action rule as a predictor. Let J_K^* be the optimal expected return among K-state controllers. A full-history policy has return one, and

$$1 - J_K^* = E_{\mathrm{seq}}(K). \tag{20}$$

Thus the calibration profile is exactly a worst-case optimal return-gap profile for this exogenous control class.

When the FSM can be written through actions. Like external-memory policies (Peshkin et al., 1999; Toro Icarte et al., 2020), this implementation needs writable codes. Assign each state a distinct action code $c(s)$ with exact feedback. Suppose the initial action is $c(S_0)$, every required write is legal, and intermediate actions affect neither observations nor reward. Then $a_t = c(u_t(c^{-1}(a_{t-1}), X_t))$ implements the FSM, followed by the separate terminal decision $b(c^{-1}(a_N))$. Stochastic tables use the same encoding with fresh draws. This realizes the same-state construction.

The write condition matters: if every legal command returns the same cue, the feedback cannot retain a binary state. Under restricted writes, the available policies form a subset of these FSMs, so same-task lower bounds still apply; implementing the upper construction requires writable codes.

7 Related work

From action feedback to a memory budget. Previous actions are supplied to policies in control (Rudin et al., 2022), and have been identified as a way to retain a cue in a memory benchmark (Ni et al., 2023). Finite-state controllers optimize actions and memory updates (Meuleau et al., 1999; Amato et al., 2010); external-memory RL studies explicit write actions and action–memory product spaces (Peshkin et al., 1999; Toro Icarte et al., 2020). These interfaces motivate our exact K-state execution budget under the writable-code conditions of Section 6. Kirtland et al. (n.d.) frame memory as trajectory abstraction. History abstractions (Majeed & Hutter, 2021) and approximate information states (Subramanian et al., 2022) already connect representation quality to planning and control. In particular, Subramanian et al. (2022, Theorem 9) bound finite-horizon planning loss from approximate reward and next-information-state prediction conditions across histories and actions. Recursive-update AIS conditions and synthesized finite information-state machines provide executable memory under additional state-update or history-wide consistency conditions (Subramanian et al., 2022; Kalat et al., 2024). We study the guarantee obtained from a different representation score: the average terminal error of prefix messages optimized independently at each cut. Theorem 1 determines its exact-K online execution cost. Finite-window policies have discounted control bounds under filter-stability assumptions (Kara & Yüksel, 2022); their memory grows with the window length.

Learned recurrent memory. Recurrent RL learns history encoders (Ni et al., 2022). Prefix-scannable models compose compatible summaries; their general online scan retains logarithmically many chunk states (Yau et al., 2026). Vector-valued memory differs from a register with exactly K possible values.

Communication and small-width computation. At zero error, the fixed-order OBDD criterion recalled by Brody et al. (2011, Fact 7) characterizes width K by at most K distinct suffix-response classes at every cut. Their Appendix B illustrates how partial domains complicate this compatibility question. The deterministic streaming conversion of Woodruff & Yang (2019, Theorem 3), stated for symmetric functions, reconstructs compatible prefixes; Du et al. (2023, Theorem 5.7) use posterior resampling under product distributions to convert worst-case two-player protocols after repeating the protocol to amplify accuracy, which increases its message length. These results establish a natural route from cut messages to streaming memory. Applying posterior recompression to the original-law average cut score gives our exact-state bound $E_{\mathrm{seq}}(K) \le 2\sum_{t<N} E_{\mathrm{cut},t}(K)$ in Proposition 4. For the converse, Goldreich (2010, Lemma A.6) gives same-width XOR correlation multiplication, and Brody et al. (2011) give parity/AND width obstructions. Our tunable delayed-query blocks satisfy the cut premise at every split and bound the best K-state machine, completing the calibration profile of Theorem 1.

Error notions. Our use of calibration relates optimal *absolute* errors at the same memory budget; surrogate-risk calibration studies another error conversion (Bartlett et al., 2006). The supplementary certificate and compiler-specific excess risk concern supplied objects.

8 Discussion

The previous-action register captures the obstacle: it must choose what to carry forward before it knows which detail the next observation will query. A cut encoder can reread completed events and overlook one rare unfinished query. A finite-state machine has to carry the pending information while retaining the answer accumulated so far, and the conflict can recur at every block. This is why a small error at each cut can add up even after choosing the best machine.

For independent finite observations and a binary terminal target, our bounds put a precise budget on this conversion: across the task class, best-machine error has order $\min\{N\varepsilon, 1\}$ for every finite $K \ge 2$. The compiler shows that cut error of order $1/N$ suffices for a uniform small-error guarantee without enlarging the state; the delayed-query family shows why that horizon dependence is needed. The law quantifies what cutwise summary accuracy guarantees about the best online machine at the same state budget.

REPRODUCIBILITY STATEMENT

Appendix A proves the uniform law; the other appendices contain the two-state calculation and certificate proofs. The supplementary ZIP provides the two figure scripts, their analytical arrays and PDF exports, and an exact 30-case one-block delayed-query check with stored optimal codebooks and state-transition tables. Its README gives standalone CPU commands for regenerating the figures and checking those finite cases.

AI USE DISCLOSURE

AI tools assisted with mathematical development, source comparison, numerical checks, figures, code preparation and writing.

REFERENCES

Christopher Amato, Daniel S. Bernstein, and Shlomo Zilberstein. Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *Autonomous Agents and Multi-Agent Systems*, 21(3):293–320, 2010. doi: 10.1007/s10458-009-9103-z. URL https://link.springer.com/article/10.1007/s10458-009-9103-z.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. doi: 10.1198/016214505000000907. URL https://www.stat.berkeley.edu/~bartlett/papers/bjm-ccrb-05.pdf.

Joshua Brody, Kevin Matulef, and Chenggang Wu. Lower bounds for testing computability by small width OBDDs. In *Theory and Applications of Models of Computation (TAMC)*, pp. 320–331, 2011. doi: 10.1007/978-3-642-20877-5_32. URL https://web.mit.edu/matulef/www/papers/lbsforobdds-full.pdf. Full author version, including appendices.

Elbert Du, Michael Mitzenmacher, David P. Woodruff, and Guang Yang. Separating k-player from t-player one-way communication, with applications to data streams. *Theory of Computing*, 19 (10):1–44, 2023. doi: 10.4086/toc.2023.v019a010. URL https://theoryofcomputing.org/articles/v019a010/.

Oded Goldreich. On testing computability by small width OBDDs. Technical Report TR10-061, revision 2, Electronic Colloquium on Computational Complexity, 2010. URL https://eccc.weizmann.ac.il/report/2010/061/revision/2/download/.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL https://doi.org/10.1080/01621459.1963.10500830.

Shadi Tasdighi Kalat, Sriram Sankaranarayanan, and Ashutosh Trivedi. Anticipating oblivious opponents in stochastic games, 2024. URL https://arxiv.org/abs/2409.11671. arXiv preprint arXiv:2409.11671.

Ali Devran Kara and Serdar Yüksel. Near optimality of finite memory feedback policies in partially observed Markov decision processes. *Journal of Machine Learning Research*, 23(11):1–46, 2022. URL https://jmlr.org/papers/v23/20-1152.html.

Aaron Kirtland, Alexander Ivanov, Cameron Allen, Michael L. Littman, and George Konidaris. Memory as state abstraction over trajectories, n.d. URL https://camallen.net/files/traj_abstr_preprint.pdf. Undated author preprint; accessed September 22, 2026.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1555–1561, 2001. URL https://papers.nips.cc/paper_files/paper/2001/file/1e4d36177d71bbb3558e43af9577d70e-Paper.pdf.

Sultan J. Majeed and Marcus Hutter. Reducing planning complexity of general reinforcement learning with non-Markovian abstractions, 2021. URL https://arxiv.org/abs/2112.13386. arXiv preprint arXiv:2112.13386.

Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie Pack Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 427–436, 1999. URL https://arxiv.org/abs/1301.6721.

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free RL can be a strong baseline for many POMDPs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16691–16723, 2022. URL https://proceedings.mlr.press/v162/ni22a.html.

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in RL? decoupling memory from credit assignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9dc5accb1e4f4a9798eae145f2e4869b-Paper-Conference.pdf.

Leonid Peshkin, Nicolas Meuleau, and Leslie Pack Kaelbling. Learning policies with external memory. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 307–314, 1999. URL https://arxiv.org/abs/cs/0103003.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 91–100, 2022. URL https://proceedings.mlr.press/v164/rudin22a.html.

Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal of Machine Learning Research*, 23(12):1–83, 2022. URL https://www.jmlr.org/papers/v23/20-1165.html.

Rodrigo Toro Icarte, Richard Valenzano, Toryn Q. Klassen, Phillip Christoffersen, Amir-massoud Farahmand, and Sheila A. McIlraith. The act of remembering: A study in partially observable reinforcement learning, 2020. URL https://arxiv.org/abs/2010.01753v1. arXiv preprint arXiv:2010.01753, version 1.

David P. Woodruff and Guang Yang. Separating k-player from t-player one-way communication, with applications to data streams. In *46th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 132 of *Leibniz International Proceedings in Informatics*, pp. 97:1–97:14, 2019. doi: 10.4230/LIPIcs.ICALP.2019.97. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2019.97.

Morris Yau, Sharut Gupta, Valerie Engelmayer, Kazuki Irie, Jacob Andreas, and Stefanie Jegelka. Sequential-parallel duality in prefix-scannable models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4fef2f009c960581664363987fbe34d5-Paper-Conference.pdf.

A General finite memory budgets

The two-state example explains the obstruction with a single pending bit. For arbitrary K, the pending information becomes a vector whose feature is revealed only by a later query. The following packing and block-composition argument proves the uniform lower bound in Theorem 1. Throughout, $q = 2^d$ for an integer $d \geq 1$ and $0 < p \leq 1/2$.

A.1 One block: separated answers require different states

For Boolean functions h, r of a suffix with law ν, write $d_\nu(h, r) = \Pr_{Z\sim\nu}\{h(Z) \neq r(Z)\}$. This weighted Hamming distance satisfies the triangle inequality, even if the law has zero-mass points.

Lemma 7 (Packing arbitrary response functions)**.** *Suppose a prefix type I is uniform on $\{1, \ldots, M\}$ and independent of a suffix $Z \sim \nu$. Its target response is $h_I(Z)$. If $d_\nu(h_i, h_j) \geq \Delta$ for all $i \neq j$, any encoder with at most K labels and any Boolean decoder have error at least*

$$\frac{\Delta}{2}\left(1 - \frac{K}{M}\right)_+. \tag{21}$$

The decoder functions need not belong to the set $\{h_i\}$.

Proof. A label selects a Boolean response $r_s(Z)$. Each of these at most K responses can have distance strictly less than $\Delta/2$ from at most one h_i, by the triangle inequality. Therefore at least $(M - K)_+$ prefix types have distance at least $\Delta/2$ from every selected response, regardless of which label the encoder assigns. Averaging their equal probabilities proves the claim. The bound also covers randomized choices by fixing their input-independent randomness, or by observing that a mixture of assignments cannot improve on the nearest available response for a given type. □

Choose an integer $d \geq 1$, let $q = 2^d$, and take $0 < p \leq 1/2$. The three observations in one block, in order, are mutually independent:

$$\begin{aligned} &A \sim \text{Bernoulli}(1/2), \qquad && B \sim \text{Unif}(\mathbb{F}_2^d), \\ &\Pr(C = \perp) = 1 - p, \qquad && \Pr(C = z) = \frac{p}{q-1} \quad (z \in \mathbb{F}_2^d \setminus \{0\}). \end{aligned} \tag{22}$$

The sentinel $\perp$ is a distinct symbol. Define

$$v(b, c) = \begin{cases} 0 & c = \perp, \\ \langle b, c\rangle & c \neq \perp, \end{cases} \qquad F(A, B, C) = A \oplus v(B, C), \tag{23}$$

where the inner product is over $\mathbb{F}_2$. Reading B precedes knowing which linear feature will be queried. The answer bit A must also be retained. An atomic vector is one observation here.

Lemma 8 (The one-block bottleneck)**.** *Let $e_K(q, p)$ denote the best K-state execution error for the block in* (22)–(23)*, with $K \geq 2$. It equals the optimal cut error after (A, B) and satisfies*

$$\frac{pq}{4(q-1)}\left(1 - \frac{K}{2q}\right)_+ \leq e_K(q, p) \leq \frac{p}{2}. \tag{24}$$

In particular, if $q \geq K$, then $e_K(q, p) \geq p/8$. For this fixed block, zero error requires and is achievable with $2q$ states.

Proof. After $(A, B) = (a, b)$ the response is $h_{a,b}(c) = a \oplus v(b, c)$. All $2q$ types have equal probability. If $b \neq b'$, their difference is a nonzero linear functional, which equals one on exactly $q/2$ nonzero vectors. Putting $\alpha = pq/[2(q-1)]$, the pairwise distances are

Pair of distinct types	Distance under the law of C
$a = a', b \neq b'$	α
$a \neq a', b = b'$	1
$a \neq a', b \neq b'$	$1 - \alpha$

Since $q \geq 2$ and $p \leq 1/2$, one has $p/2 \leq \alpha \leq 1/2$. Lemma 7, with $M = 2q$, yields the lower bound. Ignoring the query and returning A gives error $p/2$: every nonzero query is balanced over uniform B. Balance over queries for every fixed B is not needed.

Any optimal cut encoder $e(a, b)$ and decoder $r_s(c)$ can be executed in three steps: store A in labels 0/1; upon observing B, write $e(A, B)$; upon observing C, write $r_s(C)$ in labels 0/1 and apply the identity readout. All steps use at most K labels. Thus the cut optimum is also the execution optimum. For $q \geq K$, the lower bound is at least $p/8$.

For zero error, all $2q$ positive-mass response types must have distinct states because their pairwise distances are positive. Conversely, storing (A, B) uses $2q$ states and evaluates the final query exactly. Randomization cannot improve this by fixing the independent random choices. □

A.2 Independent blocks: the lower bound survives global optimization

For a sign target h of a finite input stream, let $c_K(h)$ be the largest correlation $\mathbb{E}[h\widehat{h}]$ over deterministic K-state machines with a known fixed start and arbitrary sign readout. Relabeling states makes this quantity independent of the chosen initial label. Output complementation implies that the maximum absolute correlation is also $c_K(h)$, and $c_K((-1)^f) = 1 - 2E_{\mathrm{seq},f}(K)$.

Lemma 9 (Same-width XOR composition)**.** *For independent finite streams Y, Z and sign targets $h_1(Y), h_2(Z)$,*

$$c_K(h_1 h_2) \leq c_K(h_1) c_K(h_2). \tag{25}$$

Proof. Any deterministic machine has boundary state $S(Y)$ and final sign output $g(S(Y), Z)$. For $a_s = \mathbb{E}[h_2(Z) g(s, Z)]$, fixing the suffix start state and allowing either readout sign gives $|a_s| \leq c_K(h_2)$. Independence yields

$$\begin{aligned}
\mathbb{E}[h_1(Y) h_2(Z) g(S(Y), Z)] &= \mathbb{E}[h_1(Y) a_{S(Y)}] \\
&\leq c_K(h_2) \mathbb{E}\,|\mathbb{E}[h_1(Y) \mid S(Y)]| \\
&\leq c_K(h_2) c_K(h_1).
\end{aligned}$$

The last inequality uses the best sign readout of the existing prefix state. It costs no new state. Taking the maximum proves the assertion. Time-dependent suffix tables can be reindexed to start at local time one. Randomized full machines are covered by fixing the independent random choices. □

This is the same-width argument of Goldreich (2010, Lemma A.6). It allows arbitrary boundary states across blocks.

Take $m \geq 1$ independent copies of (22), and let $f_m = \bigoplus_{i=1}^{m} F(A_i, B_i, C_i)$. For $0 \leq u \leq 1/2$, write

$$G_m(u) = \frac{1 - (1 - 2u)^m}{2}. \tag{26}$$

Proof of Proposition 6. Let P_i be the exact XOR of the completed blocks before block i. It is available to an offline prefix encoder. At a block boundary send the completed parity, and decode all future blocks from the suffix; the error is zero. After A_i, send $P_i \oplus A_i$; the suffix contains B_i, C_i and all later blocks, so decoding is again exact. After B_i, the same one-bit encoding can ignore the unfinished query and decode later blocks exactly, incurring error $p/2$. This already establishes admissibility at every cut.

For the equality in (18), put $H_i = P_i \oplus A_i$. The fresh fair mask makes (H_i, B_i) uniform on $2q$ types and independent of C_i and all future blocks. Encode (H_i, B_i) with an optimal one-block cut codebook; decode it and XOR the suffix-known answers of later blocks. This attains $e_K(q, p)$ at the cut after B_i.

Conversely, an arbitrary prefix encoder induces, conditional on (H_i, B_i), a possibly randomized label through its access to additional past history. That extra history is independent of C_i and the future blocks. Fix any positive-probability realization of the future blocks and XOR its known target out of the decoder's output. The resulting rule, together with the conditional label distribution, is a valid randomized codebook for the one-block response problem. Its risk is at least $e_K(q, p)$, since a

randomized assignment cannot improve the minimum for fixed response functions. Averaging over future blocks proves the reverse cut inequality.

Lemma 9 gives correlation at most $(1 - 2e_K(q, p))^m$, proving the execution bound. If $K < 2q$, the one-block error lies in $(0, 1/2)$, by Lemma 8. Hence $G_m(e_K) > e_K$ for $m \geq 2$. Finally, a $2q$-state machine stores $(P_i \oplus A_i, B_i)$ at the middle cut, then updates the parity after C_i. The clock specifies which state encoding is in use; no extra phase register is stored. □

Lemma 10 (Independent padding)**.** *Appending independent observations that do not enter the target leaves optimal execution error unchanged. If $K \geq 2$, the new cuts after the genuine input can encode the exact target without error.*

Proof. Ignoring padding proves one direction. For the reverse, fix the padding and all runtime random draws. The portion after the genuine stream becomes a deterministic binary readout of its final state. Independence preserves the original task law under this conditioning. Each resulting machine has error at least the optimum for the unpadded task; averaging proves the claim. At new cuts an offline encoder sends the completed Boolean target in two labels. At earlier cuts, a decoder can ignore the padding, so their achievable errors do not increase. □

Proof of Theorem 1. For $\varepsilon > 0$, set $p = 2\varepsilon \leq 1/2$ and $q = 2^{\lceil \log_2 K \rceil}$. Then $K \leq q < 2K$ and Lemma 8 gives $e_K(q, p) \geq p/8 = \varepsilon/4$. With $m = \lfloor N/3 \rfloor$, Proposition 6 and Lemma 10 construct an admissible length-N task satisfying

$$E_{\mathrm{seq}}(K) \geq G_m(\varepsilon/4) = \frac{1 - (1 - \varepsilon/2)^m}{2}. \tag{27}$$

Every observation has a finite alphabet, all observations are independent, and the target is Boolean. Thus this is a member of the supremum in (4). Proposition 4 and the constant predictor prove the upper envelope. At $\varepsilon = 0$ that upper bound gives $\Gamma_{N,K}(0) = 0$, and the lower envelope is zero too.

For the order constants, put $u = \varepsilon/2$. Bernoulli's inequality gives $(1 - u)^{-m} \geq 1 + mu$ for $0 \leq u < 1$. Consequently the lower envelope is at least

$$\frac{m\varepsilon}{4 + 2m\varepsilon}. \tag{28}$$

Writing $N = 3m + r$ with $0 \leq r \leq 2$ and $m \geq 1$ shows $N \leq 5m$. With $x = N\varepsilon$, the preceding quantity is at least $x/(20 + 2x) \geq \min\{x, 1\}/22$. The upper envelope is at most $2\min\{x, 1\}$, proving (6). □

A.3 A SEPARATE CONSTRUCTION WITH INDEPENDENT BINARY OBSERVATIONS

Let $\Gamma^{\mathrm{bin}}_{N,K}$ denote (4) restricted to binary observations with independent, potentially biased coordinates.

Proposition 11 (Binary lower bound with the block length charged)**.** *For $K \geq 2$, put $d = \lceil \log_2 K \rceil$ and $L = 2d + 2$. For $N \geq L$ and $0 \leq \varepsilon \leq 1/4$,*

$$\Gamma^{\mathrm{bin}}_{N,K}(\varepsilon) \geq \frac{1 - (1 - \varepsilon/2)^{\lfloor N/L \rfloor}}{2} \geq \frac{1}{6} \min\{\lfloor N/L \rfloor \varepsilon, 1\}. \tag{29}$$

For $N < L$, only the trivial zero lower bound is asserted by this construction.

Proof. For $\varepsilon > 0$, put $p = 2\varepsilon$, $q = 2^d$, and observe in order

$$A, B_1, \ldots, B_d, R, Z_1, \ldots, Z_d. \tag{30}$$

All variables are independent; A, all B_j and all Z_j are fair bits, while $R \sim \mathrm{Bernoulli}(p)$. The block target is $F_{\mathrm{bin}} = A \oplus R\langle B, Z\rangle$. Zero query vectors are allowed: excluding them would destroy independence of the query coordinates.

After all B_j, the $2q$ uniform response types have suffix functions $r_{a,b}(R, Z) = a \oplus R\langle b, Z\rangle$. Their pairwise distances are $p/2$ for equal masks and distinct vectors, 1 for opposite masks and equal vectors, and $1 - p/2$ otherwise. Lemma 7 therefore gives

$$E_{\mathrm{seq},F_{\mathrm{bin}}}(K) \geq \frac{p}{4}\left(1 - \frac{K}{2q}\right) \geq \frac{p}{8}. \tag{31}$$

Here allowing an arbitrary prefix encoder and a Boolean function of the entire suffix relaxes the streaming restrictions and gives a valid lower bound.

Repeat $m = \lfloor N/L \rfloor$ independent blocks and XOR their targets. At block boundaries, an offline encoder sends exact completed parity. At any interior cut, it can send completed parity XOR the current A and omit the entire unfinished gated inner product; the decoder computes subsequent blocks from the suffix. Its unconditional error is

$$\tau = \Pr\{R\langle B, Z\rangle = 1\} = \frac{p(q-1)}{2q} \le p/2. \tag{32}$$

Indeed, $B = 0$ contributes zero, while each nonzero B has a fair inner product with uniform Z. This same omission rule works inside the B bits, after R, and inside the Z bits; it does not need unavailable past coordinates. Immediately after A the suffix can alternatively evaluate the current query exactly. All cut encoders use at most two labels.

Apply Lemma 9 to (31), and append independent fair padding bits as in Lemma 10. The resulting length-N task is admissible and proves the first inequality in (29). With $x = m\varepsilon$, the same Bernoulli bound used earlier gives $x/(4+2x) \ge \min\{x, 1\}/6$. The endpoint $\varepsilon = 0$ follows from the nonnegative risk and the upper bound. □

For fixed K, L is a constant. When K varies, the block length records the $\log K$ observations needed to present a vector one bit at a time.

B THE TWO-STATE ILLUSTRATION

We give the one-block calculation and composition argument for Lemma 5, then derive the accuracy consequences of the profile.

B.1 EXACT ONE-BLOCK ERROR

Consider $F(A, B, C) = A \oplus (B \wedge C)$ under iid $\mathrm{Bernoulli}(r)$ inputs, $0 < r \le 1/2$. At the cut after A, B, the four possible prefixes (A, B) leave the Boolean functions $0, C, 1, 1-C$ of the remaining bit C, with probabilities $(1-r)^2, r(1-r), r(1-r), r^2$. Each controller state selects a rule that maps C to the final answer, called a *response function*. For a fixed pair of rules (a *codebook*), the best encoder assigns each prefix to the response of smaller conditional error. The six possible pairs give the following weighted errors.

Response codebook	Minimum error
$\{0, 1\}$	r^2
$\{0, C\}$	$r(1-r)$
$\{0, 1-C\}$	$2r^2(1-r)$
$\{1, C\}$	$r((1-r)^2 + r^2)$
$\{1, 1-C\}$	$(1-r)^2$
$\{C, 1-C\}$	$r(1-r)$

All are at least r^2 for $0 < r \le 1/2$. For the fourth entry, the difference from r^2 is $r(1-r)(1-2r) \ge 0$; the other comparisons are immediate. Duplicate response functions cannot improve the optimum. A two-state machine that retains A and ignores B, C has error r^2. Thus the optimal one-block cut and execution errors both equal r^2. Encoding the target and prediction as signs in $\{-1, +1\}$, their correlation is the expected product of those signs. Its maximum over two-state machines is $1 - 2r^2$.

B.2 XOR COMPOSITION AND ATTAINMENT

Lemma 9, applied to the independent masked blocks, bounds the optimal two-state correlation by $(1-2r^2)^m$. A two-state machine that tracks $\bigoplus_i A_i$ ignores all B_i, C_i and has error $[1 - (1 - 2r^2)^m]/2$, attaining the bound and proving the execution formula.

B.3 Every cut and independent padding

At a complete-block boundary, the prefix encoder sends completed target parity and decoding is exact. After A_i, it sends completed parity XOR A_i, and the suffix decoder can compute B_iC_i exactly. After B_i, it sends the same bit and ignores the straddling AND, with error r^2. Cuts within irrelevant padding can transmit the exact target.

Padding cannot lower optimal execution error. Condition on its independent inputs and fresh random draws. The resulting processing is a fixed map from the two states at time $3m$ to a binary readout, which cannot outperform the optimal unpadded machine. Average over the conditioning. The reverse inequality follows by ignoring padding, so optimal error is unchanged.

B.4 Vanishing error and accuracy budgets

The profile bounds also give the two accuracy consequences in Section 3. For Corollary 2:

Proof. Apply both uniform constants in (6). □

For Corollary 3:

Proof. Apply the upper bound in (5) and invert its lower bound. For $m = \lfloor N/3 \rfloor \geq N/5$, the right side of (9) is at most $-2\ln(1-2\delta)/m \leq 8\delta/m \leq 40\delta/N$. The sufficient tolerance is at least $\delta/(2N)$. □

C Transition perturbations and empirical certification

Fix the encoders and decoders independently of $M \geq 1$ iid validation episodes $X^{(i)}$ from the test distribution, and let $0 < \beta < 1$. For $\mathbf{1}$ denoting an indicator, define

$$Z_{it} = \mathbf{1}\{d_t(e_t(X^{(i)}_{\leq t}), X^{(i)}_{>t}) \neq f(X^{(i)})\}, \qquad \widehat{B} = \frac{1}{M}\sum_{i=1}^{M}\left(Z_{iN} + 2\sum_{t<N} Z_{it}\right). \tag{33}$$

Proposition 12 (Held-out decoder certificate)**.** *For the exact population tables Q, with probability at least $1-\beta$,*

$$R_Q(d_N) \leq \min\left\{1, \widehat{B} + (2N-1)\sqrt{\frac{\ln(1/\beta)}{2M}}\right\}. \tag{34}$$

Proof. The episode-level losses in (33) are iid in $[0, 2N-1]$, with expectation $r_N + 2\sum_{t<N} r_t$. Apply one-sided Hoeffding (Hoeffding, 1963) and Proposition 4. □

For example, at $N = 300$, $M = 10{,}000$ and $\beta = 0.05$, the confidence term alone is about 7.33, so the bound clipped to one is vacuous.

The only transition errors that matter are those weighted by state–input pairs encountered under the ideal compiled process. For a binary output, changing the state trajectory can change the error indicator by at most one; this connects the coupling below to the terminal-error bound.

Let $X_{<t} = (X_1, \ldots, X_{t-1})$ and let P_t be the joint distribution of $(e_{t-1}(X_{<t}), X_t, e_t(X_{\leq t}))$. Its marginal a_t is the distribution of the previous summary and current input, obtained by summing over the next summary. Write TV for total variation distance. For implemented stochastic tables $\widetilde{Q}_t$, define $\eta_t = \mathrm{TV}(P_t, a_t\widetilde{Q}_t)$. From the same M validation episodes, construct the empirical triple distribution $\widehat{P}_t$ and row marginal $\widehat{a}_t$, and set $\widehat{\eta}_t = \mathrm{TV}(\widehat{P}_t, \widehat{a}_t\widetilde{Q}_t)$. Choose any fixed bound D_t on the number of possible triples, such as $K^2|\mathcal{X}_t|$, and put

$$u_t = \sqrt{\frac{D_t \ln 2 + \ln(2N/\beta)}{2M}}. \tag{35}$$

Proposition 13 (Empirical table certificate)**.** *With encoders and decoders fixed independently of the validation episodes as in Proposition 12, with probability at least $1-\beta$, simultaneously for all implemented stochastic tables,*

$$R_{\widetilde{Q}}(d_N) \le \min\left\{1, \widehat{B} + (2N-1)\sqrt{\frac{\ln(2/\beta)}{2M}} + \sum_{t=1}^{N}(\widehat{\eta}_t + 2u_t)\right\}. \tag{36}$$

C.1 Coupling using the ideal state distribution

The ideal machine Q and the implemented machine $\widetilde{Q}$ use the same actual inputs. While their states agree, couple their next-state draws to maximize the probability of agreement. The probability of a first disagreement at time t is at most

$$\mathbb{E}\left[\mathrm{TV}\left(Q_t(\cdot \mid S_{t-1}, X_t), \widetilde{Q}_t(\cdot \mid S_{t-1}, X_t)\right)\right] = \eta_t. \tag{37}$$

Dropping the event of previous agreement can only increase this bound. The equality uses the ideal state distribution from Proposition 4 and independence of X_t from earlier inputs and randomness. It does not require the implemented controller's state to have this distribution. If the coupled states never differ, the common readout is identical. Hence

$$R_{\widetilde{Q}}(d_N) \le R_Q(d_N) + \sum_{t=1}^{N}\eta_t. \tag{38}$$

C.2 A bound uniform over implemented tables

For any row-stochastic $\widetilde{Q}_t$, the triangle inequality and TV contraction give

$$\begin{aligned}\eta_t &= \mathrm{TV}(P_t, a_t\widetilde{Q}_t)\\ &\le \mathrm{TV}(P_t, \widehat{P}_t) + \mathrm{TV}(\widehat{P}_t, \widehat{a}_t\widetilde{Q}_t) + \mathrm{TV}(\widehat{a}_t\widetilde{Q}_t, a_t\widetilde{Q}_t)\\ &\le \widehat{\eta}_t + 2\,\mathrm{TV}(P_t, \widehat{P}_t).\end{aligned} \tag{39}$$

This deterministic inequality holds for every stochastic table simultaneously. On an alphabet of size at most D_t, $\mathrm{TV}(P_t, \widehat{P}_t) = \max_A(P_t(A) - \widehat{P}_t(A))$. For each of at most 2^{D_t} subsets A, one-sided Hoeffding (Hoeffding, 1963) bounds the probability of deviation exceeding u by e^{-2Mu^2}. Thus

$$\mathbb{P}\{\mathrm{TV}(P_t, \widehat{P}_t) > u_t\} \le \frac{\beta}{2N}, \qquad u_t = \sqrt{\frac{D_t \ln 2 + \ln(2N/\beta)}{2M}}. \tag{40}$$

A union bound over times gives all table bounds with probability at least $1-\beta/2$. Applying Proposition 12 with failure probability $\beta/2$ controls the population weighted decoder loss on another event. Their intersection, (39), and (38) prove Proposition 13.

Neither event needs independence between cuts, nor independence between the decoder-loss statistics and the triple statistics. The encoders and decoders are fixed independently of these episodes. The tables may depend on the episodes because the event is uniform over all table choices. If the tables are empirical conditional frequencies, then $\widehat{P}_t = \widehat{a}_t\widetilde{Q}_t$, including empty rows of zero empirical weight. This proves $\widehat{\eta}_t = 0$ without a minimum row mass.

If the terminal decoder is known to be exact under the original input distribution, omit Z_{iN} and replace $2N-1$ by $2(N-1)$ in the certificates. If decoders or encoders are selected using these same episodes, the present Hoeffding argument does not apply to that selection; an independent evaluation split or a justified uniform bound is needed.

D THE THREE-OBSERVATION COUNTEREXAMPLE TO COMPILER OPTIMALITY

Let $A \sim \text{Bernoulli}(1/2)$, $B \sim \text{Bernoulli}(r)$ and $Y \sim \text{Bernoulli}(h)$ be independent and observed in that order, where $0 < r, h < 1/2$. Set $f = A$ when $Y = 0$ and $f = B$ when $Y = 1$.

Proposition 14 (Optimal cuts, suboptimal posterior execution)**.** *At $K = 2$, $E_{\text{cut}}(2) = E_{\text{seq}}(2) = hr$. The encoders $e_1 = A$, $e_2 = A$, $e_3 = f$, with exact decoders at times 1 and 3 and $d_2(a, 0) = a$, $d_2(a, 1) = 0$, achieve the optimal error at every cut. Their posterior compiler nevertheless has*

$$R_Q(d_3) = 2hr(1 - r), \qquad R_Q(d_3) - E_{\text{seq}}(2) = hr(1 - 2r) > 0. \tag{41}$$

At time 1, storing A permits exact decoding using the suffix (B, Y). At time 3, the prefix encoder $e_3 = f$ permits exact final decoding. At time 2, each of the two summary values corresponds to a Boolean response function of Y, written as the pair of outputs at $Y = 0$ and $Y = 1$. For each unordered pair of distinct response functions, the encoder assigns each (A, B) to whichever incurs smaller conditional prediction error. This gives the following complete table; duplicate response functions cannot improve any optimum.

Response functions	Optimal encoder error
$\{00, 01\}$	$(1 - h)/2$
$\{10, 11\}$	$(1 - h)/2$
$\{00, 10\}$	hr
$\{01, 11\}$	$h(1 - r)$
$\{00, 11\}$	$h/2$
$\{01, 10\}$	$h/2$

For example, $\{00, 10\}$ reproduces A at $Y = 0$ and always predicts zero at $Y = 1$, costing hr. For the constant codebook $\{00, 11\}$ the optimal choice agrees with A because $h < 1/2$, and errors occur when $Y = 1$ and $A \neq B$, costing $h/2$. The other cases follow by the same conditional comparison. Since $0 < r, h < 1/2$, the minimum entry is hr.

The controller that stores A, ignores B, and resets to zero for $Y = 1$ achieves this cut lower bound. Hence $E_{\text{cut}}(2) = E_{\text{seq}}(2) = hr$. For the supplied encoders in Proposition 14, $S_1 = S_2 = A$. At the last step, conditioning on $e_2 = A$ leaves the forgotten B independent with its original $\text{Bernoulli}(r)$ distribution. For $Y = 1$ the last posterior table therefore outputs a fresh sample B', independent of the actual B. For $Y = 0$ it outputs A exactly. This gives $R_Q(d_3) = h\mathbb{P}(B \neq B') = 2hr(1 - r)$ and the stated positive excess.

All individual cut errors are minimal, so these supplied objects also minimize any positive weighted sum of those errors and their maximum. Appending independent irrelevant observations preserves the optimum: the second-cut lower bound is unchanged and the same machine can carry its final answer through the additional steps. Setting $r = 1/4$ and $h = N^{-1/2}$ for $N > 4$ gives an individual-task sequence with $E_{\text{seq}} = E_{\text{cut}} = 1/(4\sqrt{N}) \to 0$ while $NE_{\text{cut}} = \sqrt{N}/4 \to \infty$. Thus the class-wide calibration criterion need not be necessary on each individual task.