PROVABLE TRANSFORMER CONTEXT COMPRESSION FOR IN-CONTEXT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers can perform in-context reinforcement learning without parameter updates. However, existing theoretical constructions often require context sizes that grow with the number of sampled transitions, leading to increasing memory costs in large-scale problems. This raises a fundamental scalability question: how much context is intrinsically necessary to carry out such learning? We answer this question for fixed-policy TD evaluation. We show that, with a shared linear decoder, the minimum context dimension required to exactly reproduce the first L TD iterates is the Krylov dimension induced by the sampled transitions. When the available context dimension m falls below this threshold, exact reproduction is no longer possible; in this regime, we characterize the optimal approximation error and show that it decays as $\Theta(\gamma^m \max\{1, 1/(m(1-\gamma))\})$ where γ is the discounting factor. Moving from representation complexity to actual memory cost, we characterize the optimal retained memory for exact continuous representations as $\Theta(nk_L)$[1] and show experimentally that on a sampled task whose exact rank is full but whose numerical rank is small, a context accurate to 10^{-2} is 88.7% smaller than the sparse sufficient statistics of TD. Finally, we establish the learnability of the shared executor through pretraining by proving global convergence of an auxiliary objective over effective attention parameters under suitable task diversity. Experiments across diverse TD evaluation problems validate parameter recovery and the predicted context–accuracy tradeoff.

1 INTRODUCTION

Reinforcement learning (RL) studies how an agent learns to make decisions through interaction with an environment to maximize cumulative reward (Sutton & Barto, 2018). In conventional RL, this adaptation typically involves updating the parameters of a policy or value network from collected experience. More recently, with the growing ability of sequence models to use long interaction histories, in-context reinforcement learning (ICRL) has emerged as a different mode of adaptation: a pretrained agent can improve on a new task by conditioning on its interaction history while keeping its model parameters fixed (Moeini et al., 2025). In-context RL has demonstrated strong empirical success across a range of domains including navigation, continuous-control tasks, and open-ended embodied environments (Duan et al., 2016; Bauer et al., 2023; Lu et al., 2023; Grigsby et al., 2024).

The empirical success naturally raises the question of why ICRL can adapt to unseen tasks using context. Recent theoretical work interprets ICRL as the Transformer forward pass executing an implicit RL algorithm, and can be divided into *supervised pretraining* and *reinforcement pretraining*. Under supervised pretraining, theoretical results show that Transformers can implement a range of classical RL algorithms (Laskin et al., 2023; Lee et al., 2023; Lin et al., 2024; Liang & Lai, 2026; Chen et al., 2025). Under reinforcement pretraining, both linear and softmax Transformers can provably implement TD-style updates, with the corresponding computation emerging from policy-evaluation pretraining (Wang et al., 2025a;b; Xie et al., 2026b;a).

However, identifying the algorithm implemented in context does not reveal how much information must be retained to reproduce its update process. Existing in-context RL theory typically retains the full interaction history in the prompt. In several representative theoretical constructions, states

[1] k_L here denotes Krylov dimension induced by the sampled transitions

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.

or state–action pairs along a trajectory are represented using either one-hot embeddings in tabular settings (Lin et al., 2024; Wang et al., 2025b), or more general d-dimensional feature maps, such as $\phi : \mathcal{S} \to \mathbb{R}^d$ (Wang et al., 2025a; Liang & Lai, 2026; Xie et al., 2026b). Consequently, for a trajectory of N transitions $\{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^N$, the task-specific memory of these constructions scales as $\Theta(dN)$. Such linear growth with the interaction horizon can become a substantial bottleneck in long-horizon or high-dimensional RL problems, as storing and processing an ever-growing context incurs increasing inference-time memory and computational costs. Existing constructions therefore establish that retaining the full interaction history is sufficient, but leave open whether it is necessary. Taking TD learning for policy evaluation as a canonical example, this raises two natural questions:

(Q1) *Necessary context requirement:* What is the minimum context dimension, and the corresponding memory requirement, needed to exactly reproduce the TD iterates?

(Q2) *Approximation with limited context:* When the available context dimension falls below this threshold, what is the best achievable approximation error?

To answer these questions, we consider a trajectory $\mathcal{D}_N = \{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^N$ collected under a fixed policy, and n is the state number. Then we study the exact context required to preserve these iterates and the approximation error when the available context is limited. Our main contributions are as follows.

1. Context complexity. We characterize the intrinsic context dimension required for in-context TD learning. With a shared linear decoder, we show that the minimum context dimension required to exactly reproduce L empirical TD iterates is the Krylov dimension k_L (Saad, 2003). When the available context dimension m falls below this threshold, we further characterize the best achievable approximation and establish matching upper and lower error bounds of order $\Theta(\gamma^m \max\{1, 1/(m(1-\gamma))\})$.

2. Memory complexity. We distinguish context dimension, peak memory, and retained task memory. For representations with encoders continuous in empirical model, we characterize the minimum retained memory required to answer all state–depth TD queries exactly as $\Theta(nk_L)$ real registers[2]. On a sampled task of full exact rank but small numerical rank, a context accurate to 10^{-2} retains **88.7%** less memory than the sparse TD statistics. We further show that the optimal retained memory for answering all state–depth queries within error ϵ is $\Theta_\gamma\big(n \min\{n, L, \log(1/\epsilon)\} \log(1/\epsilon)\big)$ bits.

3. Learnability and experiment. Finally, we study whether the compressed TD computation can be learned through pretraining. Under suitable task diversity, we show that an auxiliary objective is globally strongly convex in the effective attention parameters and provide provable global convergence to the target TD update. Extensive experiments further validate parameter recovery, the predicted context–accuracy tradeoff, and substantial memory savings.

To the best of our knowledge, we are the first to characterize the context and memory requirements for preserving an in-context RL computation, as well as the approximation error below the exact-representation threshold. We require recovery across all TD depths, rather than only the terminal output, because our goal is to reproduce the algorithmic computation itself rather than merely its final result. Unlike classical compression methods, we realize the compression end-to-end within a fixed-parameter linear-attention network operating directly on raw transitions. Table 1 compares our work with existing ICRL constructions.

2 Related Work

Theoretical Mechanisms of In-Context Reinforcement Learning. Recent theory studies ICRL by viewing the Transformer forward pass as implementing an RL algorithm. Under supervised pretraining, Lin et al. (2024) construct Transformers that approximate classical online decision-making algorithms, while Liang & Lai (2026) extend this perspective to SARSA and actor–critic, and Chen et al. (2025) analyze strategies for non-stationary RL. Without explicitly supervising the internal update rule, Wang et al. (2025a) show that linear-attention Transformers can implement TD updates, and Wang et al. (2025b) characterize parameters realizing such computations as global minimizers of a pretraining objective. Xie et al. (2026b) extend the mechanism to standard softmax attention, whereas Xie et al. (2026a) interpret chain-of-thought generation as repeated TD computation.

[2]A real register denotes one storage location that can hold a real number exactly.

Table 1: Constructions that execute TD policy evaluation in the forward pass. N, n, d, L denote transitions, states, feature dimension, and TD depth; $k_L = \dim \mathcal{K}_L(\gamma\widehat{P}, \widehat{r})$. Theory counts task scalars; buffers are float32 array bytes$/1024$ over 100 encodings at $N = 2000$, $n = d = 24$, $L = 30$, $k_L = 12$, excluding parameters and forward scratch. Constructions for other in-context RL problems (Lin et al., 2024; Chen et al., 2025; Liang & Lai, 2026) retain different objects and are listed in Appendix G as not comparable.

Work	**Problem setting**	**Retained object**	**Retained memory**	**Buffer (KiB)**
Wang et al. (2025a)	TD(0), linear features, sampled trajectory	N transition tokens $(\phi_t, \gamma\phi_{t+1}, R_{t+1})$	$\Theta(Nd)$	383.00
Wang et al. (2025b)	tabular TD, known dynamics	n state tokens with rows of γP and r	$\Theta(n^2)$	4.79
Xie et al. (2026b)	softmax (kernel) TD, sampled trajectory	N state tokens (x_t, R_{t+1})	$\Theta(Nd)$	211.04
Xie et al. (2026a)	TD(0) with chain of thought, linear features	N transition tokens and L iterate tokens	$\Theta((N+L)d)$	579.15
Ours	tabular TD (Thm. 4.3) or linear features (Thm. 4.4), sampled trajectory	k_L Krylov tokens (B, Y, g)	$\mathbf{O(nk_L + n)}$	**2.49**

Reinforcement Learning and Traditional Compression. Reinforcement learning studies sequential decision making through interaction with an environment, with classical methods including temporal-difference learning, Q-learning, and policy-gradient methods (Sutton, 1988; Watkins & Dayan, 1992; Williams, 1992). For TD policy evaluation, classical compression replaces raw trajectories with sufficient statistics, as in LSTD and iLSTD (Bradtke & Barto, 1996; Boyan, 2002; Geramifard et al., 2006; Katharopoulos et al., 2020), and Krylov subspaces also arise in value approximation and model reduction(Parr et al., 2007; 2008; Petrik, 2007). Classical model-reduction methods provide a variety of low-dimensional representations for dynamical and Markov systems (Ho & Kalman, 1966; Freund, 2003; Antoulas, 2005; Kemeny & Snell, 1976; Givan et al., 2003). Unlike classical compression methods, we characterize the context needed for TD evaluation and realize the compression end-to-end within a fixed-parameter linear-attention network operating directly on raw transitions.

Context Compression. Context compression has been widely studied to reduce the computational and memory costs of models. Existing approaches either compress long contexts into compact learned representations (Rae et al., 2020; Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024; Zhang et al., 2025), or reduce the input through token/segment selection and retrieval compression (Li et al., 2023; Jiang et al., 2023; 2024; Pan et al., 2024; Xu et al., 2024). In contrast, we characterize the memory–accuracy limits of in-context TD compression and realize them directly from raw transitions with fixed-parameter attention.

3 PRELIMINARIES

Reinforcement Learning Setting. We consider an infinite-horizon discounted Markov decision process (MDP) with finite[3] state space $\mathcal{S} = [n]$, action space $\mathcal{A}$, transition kernel $P_{\mathrm{MDP}}(s' \mid s, a)$, reward function $r_{\mathrm{MDP}}(s, a)$, and discount factor $\gamma \in [0, 1)$(Puterman, 1994). For a fixed policy π, the MDP induces a Markov reward process (MRP) with transition matrix $P^\pi \in \mathbb{R}^{n\times n}$ and expected one-step reward vector $r^\pi \in \mathbb{R}^n$, where $P^\pi_{ij} := \sum_{a\in\mathcal{A}} \pi(a \mid i) P_{\mathrm{MDP}}(j \mid i, a)$ and $r^\pi_i := \sum_{a\in\mathcal{A}} \pi(a \mid i) r_{\mathrm{MDP}}(i, a)$. The policy-evaluation problem is to estimate the value function $v^\pi(i) := \mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t R_{t+1} \mid S_0 = i\right]$. Temporal-difference (TD) learning estimates this value function by bootstrapping from consecutive transitions (Sutton, 1988; Sutton & Barto, 2018). In general, a single TD(0) update takes the form $v_{t+1} = v_t + \alpha_t\big(R_{t+1} + \gamma v_t(S_{t+1}) - v_t(S_t)\big)\nabla_{v_t} v_t(S_t)$. In the remainder of the paper, we therefore work directly with the induced MRP and focus on the corre-

[3]The analysis can be extended beyond finite state spaces when states admit a finite-dimensional representation $\phi : \mathcal{S} \to \mathbb{R}^d$, in which case the corresponding constructions are carried out in the feature space.

sponding TD-learning dynamics for fixed-policy evaluation. Let $\mathcal{D}_N = \{(S_t, A_t, R_{t+1}, S_{t+1})\}_{t=0}^{N-1}$ contain N transitions collected under π. We define $N_i := \sum_{t=0}^{N-1} \mathbf{1}\{S_t = i\}$ and $N_{ij} := \sum_{t=0}^{N-1} \mathbf{1}\{S_t = i, S_{t+1} = j\}$, then assume **full state coverage**, i.e., $N_i > 0$ for all $i \in [n]$. Appendix A.1 discusses the rationale for this assumption. The empirical transition and reward estimates are $\widehat{P}_{ij} := \frac{N_{ij}}{N_i}$ and $\widehat{r}_i := \frac{1}{N_i} \sum_{t:S_t=i} R_{t+1}$. For a value vector $v \in \mathbb{R}^n$, define the empirical TD error $\widehat{\delta}_t(v) := R_{t+1} + \gamma v(S_{t+1}) - v(S_t)$ and the empirical state-visitation matrix $\widehat{D}_N := \mathrm{diag}\left(\frac{N_1}{N}, \ldots, \frac{N_n}{N}\right)$. Starting from $\widehat{v}_0 = 0$, let $\widehat{v}_\ell \in \mathbb{R}^n$ denote the empirical value estimate after ℓ iterations of visitation-normalized batch TD(0) (Yao & Liu, 2008):

$$\widehat{v}_{\ell+1} = \widehat{v}_\ell + \widehat{D}_N^{-1} \frac{1}{N} \sum_{t=0}^{N-1} \widehat{\delta}_t(\widehat{v}_\ell) e_{S_t} = \sum_{j=0}^{\ell} (\gamma \widehat{P})^j \widehat{r}. \tag{1}$$

Therefore, letting $\widehat{A} := \gamma \widehat{P}$, reproducing the first L TD iterates amounts to preserving the successive directions $\widehat{r}, \widehat{A}\widehat{r}, \ldots, \widehat{A}^{L-1}\widehat{r}$. This motivates the Krylov space (Saad, 2003)

$$\mathcal{K}_L(\widehat{A}, \widehat{r}) := \mathrm{span}\{\widehat{r}, \widehat{A}\widehat{r}, \ldots, \widehat{A}^{L-1}\widehat{r}\}, \qquad k_L := \dim \mathcal{K}_L(\widehat{A}, \widehat{r}).$$

4 Context and Memory Complexity of In-Context TD

We study the representation and memory required to reproduce the empirical TD dynamics in (1), and the best achievable accuracy when the available context falls below the exact requirement. Our construction is end-to-end: an *encoder* compresses the raw transitions $\mathcal{D}_N$ into a task-specific context within the forward pass, and an *executor* operates on this context to reproduce the queried TD values without parameter updates; linear attention is the canonical realization, and softmax attention is treated in Appendix B.

Context representation. Let $z_{\ell,i} \in \mathbb{R}^{d_{\mathrm{model}}}$ denote the ith active source token after ℓ TD updates, and $Z_\ell := [\, z_{\ell,1}, \ldots, z_{\ell,m} \,] \in \mathbb{R}^{d_{\mathrm{model}} \times m}$ denote the query-independent source context. For a state query q, let $z_{\ell,\star}(q)$ be a masked query token and define $\overline{Z}_\ell(q) := [\, Z_\ell, z_{\ell,\star}(q) \,]$. Note that the query token is not counted among the m active sources.

Definition 4.1 (m-dimensional context representation)**.** We call $\{Z_\ell\}_{\ell=0}^{L-1}$ an m-**dimensional context representation** of a trajectory $\{\widetilde{v}_\ell\}_{\ell=0}^{L}$ if there exist a shared linear decoder $D \in \mathbb{R}^{n \times m}$ and a fixed readout $\rho \in \mathbb{R}^{d_{\mathrm{model}}}$ with $s_\ell := Z_\ell^\top \rho \in \mathbb{R}^m$ and

$$\widetilde{v}_\ell = D \sum_{j=0}^{\ell-1} s_j, \qquad 0 \le \ell \le L.$$

The representation is *exact* if $\widetilde{v}_\ell = \widehat{v}_\ell$ for every $\ell \le L$. We write $\mathfrak{C}_m$ for the set of trajectories that admit an m-dimensional context representation.

The m active source tokens carry the reduced TD increments s_ℓ, and the shared decoder D maps them back to the state space; the value is $\widetilde{v}_\ell(q) = e_q^\top D \sum_{j<\ell} s_j$. The context dimension counts tokens, not scalars, so we also count the scalar storage a context occupies. For a stored object X, let $\mathrm{Loc}(X)$ be the set of distinct scalar locations occupied by X. Aliased locations are counted once and stored copies separately; both quantities count real registers and exclude shared parameters.

Definition 4.2 (Retained and peak task memory)**.** For one query q, the **retained task memory** is $M_{\mathrm{ret}} := \left|\mathrm{Loc}(\overline{Z}_\ell(q)) \cup \mathrm{Loc}(D)\right|$, the number of locations that must persist after encoding in order to execute the compressed TD computation. Let M_{enc} be the largest number of distinct locations occupied during encoding, and let $W(\ell)$ be the temporary workspace of execution layer ℓ. The **peak task memory** is $M_{\mathrm{peak}} := \max\left\{M_{\mathrm{enc}},\ \max_{0 \le \ell < L} \left|\, \mathrm{Loc}(\overline{Z}_\ell(q)) \cup \mathrm{Loc}(D) \cup \mathrm{Loc}(W(\ell)) \right|\right\}$.

4.1 End-to-End Construction: Encoder and Executor

The iterates (1) depend on the data only through $(\widehat{A}, \widehat{r})$, and their increments lie in $\mathcal{K}_L$, which suggests compressing in two stages, $N \to n \to k_L$; both are carried out by attention heads with fixed, task-independent parameters directly on the raw records, so compression is part of the forward pass.

Selector heads. During encoding let $U = [u_1, \dots, u_M] \in \mathbb{R}^{d_{\text{model}} \times M}$ collect the active tokens, each partitioned into named coordinate blocks with selectors E_a, so that $E_a u_i = u_i[a]$. An encoder head is a residual linear-attention update (Katharopoulos et al., 2020; von Oswald et al., 2023)

$$U^+ = U + PU\big[\mathcal{R} \odot (U^\top Q U)\big], \qquad P = \sigma E_o^\top E_v, \quad Q = E_k^\top E_q, \quad \sigma \in \{\pm 1\}, \tag{2}$$

which adds $\sigma \sum_i u_i[v]\,(u_i[k]^\top u_t[q])$ to block o of token t over the source tokens i allowed by the fixed routing mask $\mathcal{R} \in \{0,1\}^{M \times M}$: Q selects the key and query blocks that form the score, P selects the source value block and writes it with sign σ, into the output block, and $\mathcal{R}$ specifies which source–target pairs may interact. We also allow $\operatorname{recip}_0(x) = 1/x$ for $x > 0$, $\operatorname{recip}_0(0) = 0$, and the release of tokens at stage boundaries.

Stage 1: raw records to statistics ($N \to n$). Raw token u_t^{raw} carries $(e_{S_t}, e_{S_{t+1}}, R_{t+1}, 1)$, and state token u_i^{state} carries e_i with zero-initialized statistic blocks. Routing every raw token to every state token with the matching score $e_{S_t}^\top e_i = \mathbf{1}\{S_t = i\}$, three heads accumulate $N_{i,:}$, $\sum_{t:S_t=i} R_{t+1}$, and N_i in state token i; $\operatorname{recip}_0(N_i)$ and the input γ then form $\widehat{P}_{i,:}$, $\widehat{r}_i$, and $\widehat{A}_{i,:}$ by self-routed heads. The raw tokens are released, and the n state tokens carry $(\widehat{A}, \widehat{r})$ (details in Appendix A.3).

Stage 2: from n states to k_L Krylov directions ($n \to k_L$). The state-level context already reproduces TD exactly with the executor below (Appendix A.4), but it uses n tokens although the first L increments span only $\mathcal{K}_L$. Let $h = \min\{n, L\}$. Starting from $b_1 = \widehat{r}$ in a work token, fixed heads on h candidate tokens run h rounds of an unnormalized Arnoldi recurrence in primal–dual form (Saad, 2003),

$$d_j = b_j^\top b_j, \qquad \omega_j = \operatorname{recip}_0(d_j)\, b_j, \qquad \eta_j = d_j \operatorname{recip}_0(d_j), \qquad b_{j+1} = \widehat{A} b_j - \textstyle\sum_{i \le j} b_i\, \omega_i^\top \widehat{A} b_j, \tag{3}$$

The nonzero b_j are orthogonal and span $\mathcal{K}_L$, and the gate η_j is one exactly on them, so token release keeps exactly k_L candidates. With $B = [b_1, \dots, b_{k_L}]$ and $\Omega = [\omega_1, \dots, \omega_{k_L}]$, further heads compile

$$H = \Omega^\top \widehat{A} B, \qquad g = \Omega^\top \widehat{r}, \qquad Y = B + \Omega(H - I_{k_L})^\top, \qquad Z_0 = [\,B;\, Y;\, g^\top\,] \in \mathbb{R}^{(2n+1) \times k_L}. \tag{4}$$

Here B provides coordinates in the Krylov subspace, H and g are the reduced operator and reward, and Y stores H inside width-n blocks so that the executor can read it through inner products (Appendix A).

Executor. Write the context at depth ℓ as $Z_\ell = [B; Y; s_\ell^\top]$ with $s_0 = g$, and the masked query token as $z_{\ell,\star}(q) = (e_q, 0_n, c_\ell)$ with $c_0 = 0$. Execution reuses one linear-attention layer at every depth (Yang et al., 2024),

$$\overline{Z}_{\ell+1} = \overline{Z}_\ell + P_{\text{TD}} \overline{Z}_\ell M_{k_L} \big(\overline{Z}_\ell^\top Q_{\text{TD}} \overline{Z}_\ell\big), \qquad P_{\text{TD}} = \operatorname{diag}(0_{2n}, 1), \quad Q_{\text{TD}} = \begin{bmatrix} -I_n & I_n & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5}$$

with the mask $M_{k_L} = \operatorname{diag}(I_{k_L}, 0)$ letting only the k_L source tokens act as sources. The static blocks B, Y never change, and the last row evolves as $s_{\ell+1} = [I + (Y - B)^\top B] s_\ell = H s_\ell$ and $c_{\ell+1} = c_\ell - e_q^\top B s_\ell$, using $\Omega^\top B = I$; hence $-c_\ell(q) = e_q^\top B \sum_{j<\ell} H^j g = \widehat{v}_\ell(q)$ by Krylov moment matching, and $\{Z_\ell\}$ is an exact k_L-dimensional context representation with $D = B$ and $s_\ell = H^\ell g$.

Theorem 4.3. *Let the empirical TD system be covered and $L \ge 1$, we have: **(a) Context dimension.** The minimum dimension of an exact context representation is k_L. **(b) Construction.** For fixed n and L, a task-independent encoder maps every covered raw transitions in its forward pass, to the context (4) with exactly k_L tokens, and the executor (5) exactly outputs $\widehat{v}_\ell(q)$ for all q and ℓ. **(c) Retained memory.** The construction uses $O(nk_L)$ retained task memory and is optimal up to a factor of two among continuous exact encoders, even with arbitrary decoders. Hence the optimal retained memory is $\Theta(nk_L)$.*

Part (a) is the exact requirement: every increment $\widehat{A}^j \widehat{r} = \widetilde{v}_{j+1} - \widetilde{v}_j = D s_j$ lies in $\operatorname{col}(D)$, so $m \ge \operatorname{rank} D \ge k_L$, and (4) attains it. Part (b) is the constructive content: the k_L tokens, the decoder, and the reduced operator are produced by a fixed attention network in its forward pass, so compression is not a preprocessing step. What's more, we can get $M_{\text{ret}} = (2n+1)(k_L+1)$ and $M_{\text{peak}} = O(Nn + n^2 + nh)$ with $h = \min\{n, L\}$.

For Part (c), fix $1 \le k \le \min\{n-1, L-1\}$. On any open set[4] of covered systems with $k_L = k$ containing a model with $\mathbf{1} \notin \mathcal{K}_L$, every encoder continuous in $(\widehat{A}, \widehat{r})$ and exact for all state–depth queries must retain at least $nk + k$ real registers, even with an arbitrary nonlinear decoder (Appendix C); our context retains $(2n+1)k$ for the source tokens, within a factor two, plus one query token. The bound is class-level: on a restricted family with fewer degrees of freedom, such as the block-cycle tasks of Section 5 where $\mathbf{1} \in \mathcal{K}_L$, a family-specific encoder can retain far less; the theorem controls task-uniform encoders such as ours. If $k_L = n$ and $L \ge n+1$, at least n^2 registers are necessary, so the statistics are optimal up to $(n+1)/n$; for offline coding with arbitrary encoders and decoders, the optimal memory for all state–depth queries within error ϵ is $\Theta_\gamma(n \min\{n, L, \log(1/\epsilon)\} \log(1/\epsilon))$ bits (Appendix C.2). The same context is also executed exactly by one looped layer of standard two-head softmax attention (Appendix B), and a single-depth query needs $\Theta(\log L)$ layers (Appendix E).

Linear features and existing in-context TD constructions. Existing constructions run TD with linear features on N transition tokens and expose every iterate, one per layer (Wang et al., 2025a) or per generated token (Xie et al., 2026a), and a looped network can be read out at any loop (Yang et al., 2024); our contexts should replace those prompts at every depth without changing the executor. Let $\phi : \mathcal{S} \to \mathbb{R}^d$, $\phi_t = \phi(S_t)$, $\phi'_t = \phi(S_{t+1})$, and consider batch TD(0) in sum form with a public step size α, the update of Wang et al. (2025a) with $C_\ell = \alpha I$ and their $1/N$ absorbed into α:

$$\theta_{\ell+1} = \theta_\ell + \alpha\,(b_N - A_N\theta_\ell), \qquad A_N = \textstyle\sum_t \phi_t(\phi_t - \gamma\phi'_t)^\top, \quad b_N = \textstyle\sum_t \phi_t R_{t+1}, \quad \theta_0 = 0. \tag{6}$$

With $M := I_d - \alpha A_N$ and $s_0 := \alpha b_N$, the iterates are $\theta_\ell = \sum_{j<\ell} M^j s_0$, so their increments span $\mathcal{K}_L(M, s_0) \subseteq \mathbb{R}^d$; let $k_L^\phi := \dim \mathcal{K}_L(M, s_0)$, and let a context representation of (θ_ℓ) have a decoder $D \in \mathbb{R}^{d\times m}$ with answer $\phi(q)^\top D \sum_{j<\ell} s_j$.

Theorem 4.4 (Feature-based TD)**.** ***(a)*** *The minimum dimension of an exact context representation of $(\theta_\ell)_{\ell \le L}$ is k_L^ϕ.* ***(b)*** *For fixed d, L and public α, γ, a fixed-parameter encoder whose first stage uses no reciprocal (its second stage is that of Theorem 4.3) maps every batch of feature transitions in its forward pass to a context with exactly k_L^ϕ tokens, and the executor (5) outputs $\phi(q)^\top\theta_\ell$ for all q and $\ell \le L$; $M_{\mathrm{ret}} = (2d+1)(k_L^\phi + 1)$, $M_{\mathrm{peak}} = O(Nd + d^2 + dh)$.* ***(c)*** *Every encoder continuous in (M, s_0) on an open set of pairs with $k_L^\phi = k \le \min\{d-1, L-1\}$ and exact through an arbitrary decoder retains at least $dk + k$ registers, so the optimal retained memory is $\Theta(dk_L^\phi)$.*

Stage 1 forms A_N, b_N by two heads (key ϕ_t, query e_i) and needs no recip_0 because (6) is unnormalized, unlike the visitation-normalized tabular statistics; Stage 2, which does use recip_0 as in Section 4.1, and the executor run on (M, s_0), so the context has at most d tokens against N (Appendix A.6).

Corollary 4.5 (Drop-in replacement)**.** *With the parameters of Wang et al. (2025a) and $C_\ell = \alpha I$, their N-token prompt outputs $-\phi(q)^\top\theta_\ell$ at layer ℓ; replacing it by the d-token context $[I_d, \phi(q);\ M^\top, 0;\ s_0^\top, 0]$ or by the k_L^ϕ-token Krylov context of Theorem 4.4(b), the executor (5) outputs $\phi(q)^\top\theta_\ell$ at every layer $\ell \le L$, and the retained memory drops from $N(2d+1)$ to $(2d+1)(k_L^\phi + 1)$. On any task family with $\|M\|_2 \le \Lambda_M$, the two-head softmax executor of Appendix B with the public gain $d(2 + \sqrt{d}\,\Lambda_M)$, run on the d coordinate queries, outputs the coordinates of θ_ℓ, of which $\phi(q)^\top\theta_\ell$ is a fixed linear readout.*

4.2 Approximation with Limited Context

When $m < k_L$, exact reproduction through depth L is impossible, and we ask how accurately an m-dimensional context can approximate the TD trajectory. With $\|x\|_\mu^2 = \sum_i \mu_i x_i^2$, the normalized worst-case error of the depth-L problem of Definition 4.1 is $\mathfrak{E}_{m,L}(\gamma) := \sup_{(\widehat{P},\widehat{r},\mu)} \inf_{\widetilde{v}\in\mathfrak{C}_m} \max_{0\le\ell\le L} \|\widehat{v}_\ell - \widetilde{v}_\ell\|_\mu / \|\widehat{r}\|_\mu$, where the outer supremum is over covered empirical systems of any size with a strictly positive stationary distribution μ and $\widehat{r} \ne 0$, and $\mathfrak{C}_m$ is the class of Definition 4.1; $\mathfrak{E}_m(\gamma)$ is the depth-free version ($L = \infty$, $\sup_{\ell\ge 0}$), which dominates

[4]Here "open" means relatively open in the $k_L = k$ model class, i.e., open with respect to the subspace topology induced by $(\widehat{A}, \widehat{r})$.

$\mathfrak{E}_{m,L}(\gamma)$ for every L; the law below is stated for $\mathfrak{E}_m$, and the finite-depth case follows it. Elementary bounds leave a gap of order $(1-\gamma)^{-3/2}$ as $\gamma \to 1$, so we ask for the joint dependence on m and γ and for a context that attains it. Classical Krylov approximation (Freund, 2003; Liesen & Strakoš, 2013) does not: it spends the whole budget on the *first* m increments, so its worst-case error is exactly $\gamma^m/(1-\gamma)$ (Proposition D.1), which below the horizon barely decays with m.

Sample-and-hold context. We therefore let the context store sampled iterates at chosen depths and hold each one until the next. Let $J = \lfloor 1/(1-\gamma) \rfloor$ and split $m = k + j$ as $(m, 0)$ if $\gamma < \frac{1}{2}$, as $(0, m)$ if $\gamma \geq \frac{1}{2}$ and $m \leq J$, and as $(m-J, J)$ otherwise. The sampled depths are $\mathcal{T}_m(\gamma) = \{1, \dots, k\} \cup \{k + i_s : 1 \leq s \leq j\}$ with $i_s = \min\{i \geq 0 : 1 - \gamma^i \geq s/(j+1)\}$, so that consecutive samples split the discounted mass into equal parts. Token u stores the increment $x_u = \widehat{v}_{\tau_u} - \widehat{v}_{\tau_{u-1}}$ for $\tau_u \in \mathcal{T}_m(\gamma)$, $\tau_0 = 0$, and its depth stamp τ_u; a fixed encoder of five heads builds these tokens by latching an accumulated TD trajectory at the sampled depths (Appendix A.12). The executor is the looped layer (5) with the source recurrence replaced by a depth counter: at depth ℓ the gate $s_{\ell,u} = \mathbf{1}\{\ell+1 = \tau_u\}$ releases $e_q^\top x_u$ into the query accumulator, so the output is $\widetilde{v}_\ell = \widehat{v}_{\sigma(\ell)}$, yielding an m-dimensional context representation with $D = [x_1, \dots, x_m]$.

Theorem 4.6. *For every $m \geq 1$ and $\gamma \in (0,1)$, with $\Psi_m(\gamma) = \gamma^m \max\{1, \frac{1}{m(1-\gamma)}\}$,*

$$\tfrac{1}{64}\,\Psi_m(\gamma) \;\leq\; \mathfrak{E}_m(\gamma) \;\leq\; e^2\,\Psi_m(\gamma).$$

The upper bound is attained on every covered system by the sample-and-hold context of dimension m, consequently $\mathfrak{E}_m(\gamma) = \Theta\big(\gamma^m \max\{1, 1/(m(1-\gamma))\}\big)$.

What the lower bound rules out. Within the effective horizon, $m < 1/(1-\gamma)$, no m-dimensional context of any kind, whatever encoder produced it, has worst-case error below $\Psi_m/64 \asymp 1/(m(1-\gamma))$: the error decays at most linearly in m, and geometric decay is possible only beyond the horizon; tabular systems are feature systems with $\phi = e_s$, so the same holds for feature-based contexts. Krylov truncation exceeds the optimum by the factor $\min\{m, 1/(1-\gamma)\}/e^2$ (proofs in Appendix D; anchored hybrids and instance-dependent rates in Appendix D.1).

Finite depth. Definition 4.1 fixes a depth L, and the finite-depth error inherits the law: $\mathfrak{E}_{m,L} \leq \mathfrak{E}_m \leq e^2 \Psi_m$ for every L, and $\mathfrak{E}_{m,L} = 0$ exactly when $m \geq L$, since $k_L \leq L$. The two ring instances behind the lower bound use depths at most $m + 4/(1-\gamma)$, so $\mathfrak{E}_{m,L} \geq \Psi_m/64$ as soon as $L \geq m + 4/(1-\gamma)$: beyond the effective horizon, Theorem 4.6 holds verbatim at finite depth. For shorter depths L takes the place of the horizon $1/(1-\gamma)$: sample-and-hold with its sampled depths at the quantiles of the discounted mass through depth L has error at most $e^2 \gamma^m \max\{1, (1-\gamma^L)/(m(1-\gamma))\}$, and for $8m \leq L \leq 1/(1-\gamma)$ the optimal error is $\Theta(L/m)$ (Proposition D.9 in Appendix D.5).

4.3 Learnability of the TD Update

The construction above uses the fixed parameters $(P_{\mathrm{TD}}, Q_{\mathrm{TD}})$ to execute TD on the compressed context, and we now ask whether pretraining recovers them with the encoder kept fixed. First, $(P_{\mathrm{TD}}, Q_{\mathrm{TD}})$ is not unique: linear attention is invariant to the reciprocal rescaling $PZM(Z^\top Q Z) = (aP)ZM(Z^\top (a^{-1}Q)Z)$, $a \neq 0$, so the correct parameters form an equivalence class rather than a point. To remove this ambiguity we fix $P = P_{\mathrm{TD}}$ and parameterize the trainable query–key interaction as $Q_{\mathrm{eff}}(C, W) := \left[\begin{smallmatrix} C & W^\top & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{smallmatrix}\right]$ with $C, W \in \mathbb{R}^{n \times n}$, whose target is $C^\star = -I_n$ and $W^\star = I_n$, for which $Q_{\mathrm{eff}}(C^\star, W^\star) = Q_{\mathrm{TD}}$. Let τ index a pretraining task with empirical quantities $(\widehat{A}_\tau, \widehat{r}_\tau)$. On the state-level context of Stage 1, the executor with $Q_{\mathrm{eff}}(C, W)$ evolves the source scalars by the task-dependent operator $F_\tau(C, W) := I_n + C^\top + \widehat{A}_\tau W$, $s_{\tau,\ell+1} = F_\tau(C, W)\, s_{\tau,\ell}$, and decodes the increment $-C^\top s_{\tau,\ell}$; at the target parameters, $s_{\tau,\ell} = \widehat{A}_\tau^\ell \widehat{r}_\tau$ and the decoded increment is the empirical TD increment. Write $v^\theta_{\tau,\ell} = -C^\top \sum_{j<\ell} s_{\tau,j}$ for the value the executor produces at depth ℓ with $s_{\tau,0} = \widehat{r}_\tau$.

Which pretraining signal identifies the executor? The usual endpoint objective $\mathcal{J}_{\mathrm{end}}(C, W) = \frac{1}{2}\mathbb{E}_\tau \|v^\theta_{\tau,\infty} - \widehat{v}_{\tau,\infty}\|_2^2$ cannot identify the TD update: along $(C, W) = (-aI_n, aI_n)$, $0 < a \leq 1$, the executor runs TD with step size a, whose fixed point does not depend on a, so the minimizers of

$\mathcal{J}_{\mathrm{end}}$ form a continuum. Supervising the whole trajectory, $\mathcal{J}_{\mathrm{traj}}(C,W) = \frac{1}{2}\mathbb{E}_\tau \sum_{\ell \le L} \|v^\theta_{\tau,\ell} - \widehat{v}_{\tau,\ell}\|_2^2$, removes the degeneracy.

Proposition 4.7 (Local identifiability). *If $\mathbb{E}_\tau[\widehat{r}_\tau \widehat{r}_\tau^\top] \succ 0$, $\mathbb{E}_\tau[(\widehat{r}_\tau \widehat{r}_\tau^\top) \otimes (\widehat{A}_\tau^\top \widehat{A}_\tau)] \succ 0$, and $L \ge 2$, the Jacobian of $(C,W) \mapsto (v^\theta_{\tau,\ell})_{\ell \le L}$ at $(-I_n, I_n)$ is injective, so the target is a strict local minimizer of $\mathcal{J}_{\mathrm{traj}}$, whereas the Jacobian of $(C,W) \mapsto v^\theta_{\tau,\infty}$ there annihilates $(-I_n, I_n)$, so no endpoint objective identifies the step size.*

The proof (Appendix F) uses only the first two depths, which identify C and then W. For a global guarantee we replace $\widehat{r}_\tau$ in these two depths by a calibration input u with $\mathbb{E}[uu^\top] = I_n$, which excites every direction and makes the objective quadratic:

$$\mathcal{J}(C,W) := \tfrac{1}{2}\,\mathbb{E}_{\tau,u}\Big[\big\|-C^\top u - u\big\|_2^2 + \big\|F_\tau(C,W)\,u - \widehat{A}_\tau u\big\|_2^2\Big]. \tag{7}$$

The first term identifies the TD increment and the second the dynamics; no solved value function enters the loss, and what remains is that the tasks excite every operator direction.

Assumption 4.8 (Task richness). There are constants $0 < \kappa \le \beta'$ with $\kappa I_n \preceq \mathbb{E}_\tau[\widehat{A}_\tau^\top \widehat{A}_\tau] \preceq \beta' I_n$.

The lower bound yields strong convexity and the upper bound smoothness; the condition holds, e.g., whenever discounted permutations occur with positive probability. Let (C_t, W_t) be the parameters after t steps of full-batch gradient descent on $\mathcal{J}$ with step size η (Algorithm 1).

Theorem 4.9 (Global convergence in the effective parameter space). *Under Assumption 4.8, $\mathcal{J}$ is a globally strongly convex and smooth quadratic with the **unique minimizer** $(C^\star, W^\star) = (-I_n, I_n)$. With $\lambda_- = \frac{3-\sqrt{5}}{2}\min\{1,\kappa\}$, $\lambda_+ = \frac{3+\sqrt{5}}{2}\max\{1,\beta'\}$, and any $\eta \le 1/\lambda_+$,*

$$\|C_t - C^\star\|_F^2 + \|W_t - W^\star\|_F^2 \le (1-\eta\lambda_-)^{2t}\big(\|C_0 - C^\star\|_F^2 + \|W_0 - W^\star\|_F^2\big).$$

Thus the target is the unique global minimizer, reached from any initialization (Appendix F); the theorem identifies the executor, not the encoder, and what a jointly trained encoder learns is examined in Section 5.

5 Experiments

Effective parameter recovery. On 2,400 raw-batch tasks ($n = 24$, $L = 30$, five paired initializations, no checkpoint selection) we compare the three signals of Section 4.3. Full-batch gradient descent on the calibration objective (7) recovers $C = -I_{24}$, $W = I_{24}$ to normalized parameter error 1.0×10^{-15} in 3,000 steps, as Theorem 4.9 predicts, and Adam on (7) reaches 3.2×10^{-16}; Adam on $\mathcal{J}_{\mathrm{traj}}$ reaches 7.4×10^{-2} in 6,000 steps and 3.3×10^{-3} in 20,000, in line with Proposition 4.7; Adam on $\mathcal{J}^{(30)}_{\mathrm{end}}$, the depth-30 bootstrapped form of $\mathcal{J}_{\mathrm{end}}$ (Appendix H.6), stops at 0.51: its depth-30 output is within 0.04 of TD on held-out tasks, but its trajectory is off by up to 0.63, since $\mathcal{J}_{\mathrm{end}}$ constrains only the fixed point (Figure 4, Appendix H.6).

Context–accuracy tradeoff. Figure 1 varies the context dimension m for Krylov projection and truncation, sample-and-hold, and the Krylov-anchored hold (Appendix D.1). On 100 covered 12-block cyclic MRPs with $k_L = 12$ (panel a), Krylov projection becomes exact at the threshold of Theorem 4.3(a) (0.86 at $m = 11$, 6×10^{-16} at $m = 12$), while truncation and sample-and-hold improve gradually; on reversible MRPs with $k_L = 24$ (panel b) Krylov projection still drops from 1.1×10^{-1} at $m = 4$ to 2.2×10^{-7} at $m = 12$. On worst-case rings at $\gamma = 0.99$ (panel c), sample-and-hold and the anchored hold stay within $[0.45, 2.7] \times \Psi_m$, with sub-horizon slopes -0.89 and -0.93 against the predicted -1, whereas Krylov truncation has slope -0.14 and is up to $37\times$ worse, as Theorem 4.6 predicts.

Memory usage. In Table 2, the block cycles ($k_L = 12$ by construction) check the exact threshold: the context retains $(2n+1)(k_L+1)$ values independent of N, is not smaller than the statistics at $n = 24$, and is 94.6% smaller at $n = 480$; the bound $nk_L + k_L$ of Theorem 4.3(c) is a class-level reference, not an instance-level certificate for these tasks, whose trajectories are fixed by the block labels alone (peak memory: Appendix H.5). The feature rows run TD (6) through the encoder of

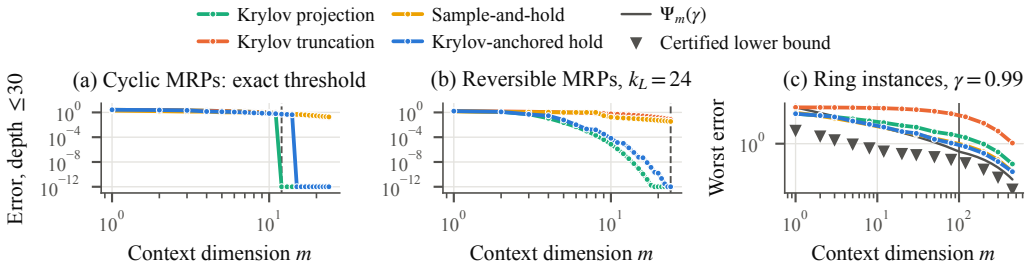


Figure 1: **Context-dimension–accuracy tradeoff.** (a) Cyclic MRPs with $k_L = 12$ and (b) reversible MRPs: mean error over 100 tasks per panel ($n = 24$, $\gamma = 0.9$, $L = 30$), interquartile ranges shaded, dashed lines at k_L. (c) Worst-case rings at $\gamma = 0.99$ against Ψ_m of Theorem 4.6 and a certified lower bound (vertical line: $m = 1/(1-\gamma)$).

Table 2: **Retained memory** (KiB, float32). Top: 12-block cycles with $k_L = 12$. Bottom: feature TD (6), Statistics $= (A_N, b_N)$, Ours = the exact context (Boyan) or the dimension-32 context with error $\leq 1.1 \times 10^{-2}$ (mountain car; Raw stored sparsely). Peak as in Definition 4.2.

Task	n or d	N	Raw	Statistics	Sparse TD	Ours	Peak
Block cycle	24	960	11.44	2.44	**0.75**	2.49	37.5
Block cycle	480	19200	228.8	903.8	102.2	**48.8**	10822.5
Boyan's chain	4	20000	703.1	**0.08**	–	0.18	1.1
Mountain car	452	320000	21250	799.8	–	**116.7**	9226

Theorem 4.4: on Boyan's chain ($d = 4$, $k_L^\phi = 4$) nothing is saved over the $d^2 + d$ LSTD statistics; on mountain car with tile-coded features ($d = 512$) the trajectory has full exact rank but at most 10 singular values above $10^{-2}\sqrt{L}$, and a context of dimension 32 is accurate to 1.1×10^{-2} or better with 108–117 KiB, 6–7× below the LSTD statistics and independent of N. On sampled tabular tasks of full exact rank, contexts accurate to 10^{-2} are 9× smaller than sparse TD on a dense kernel but not on a sparse gridworld (Appendix H.5).

Learned bottleneck encoders. Finally we train standard softmax Transformers that compress the TD iterates of the fixed first stage into m latent tokens by cross-attention and emit a context for the executors of Appendix A.7, so every model lies in $\mathfrak{C}_m$ (Appendix H.9; three seeds, 8,000 updates per point). The learned encoders beat sample-and-hold on held-out tasks for $m \leq 8$ (0.50–0.89 against 1.00–1.72) and the anchored hold for $m \leq 4$, but their error does not decrease with m: it stays at 0.61–0.63 (clocked, $m \geq 4$) and 0.50–0.54 (time-invariant, $m \geq 8$) up to $m = 24$, where the compiled contexts reach 0.10 and 0.03; standard bottleneck encoders do not discover the Krylov or clock structure that turns tokens into accuracy, which the compiled encoders supply.

6 Conclusion

We characterize the intrinsic context complexity of in-context TD evaluation through the Krylov dimension of the empirical TD dynamics. A fixed-parameter linear-attention network constructs the corresponding compressed context directly from raw transitions and reproduces the same TD trajectory, with retained memory within a factor of two of the optimum among continuous exact encoders. Below the exact-representation threshold, we establish a γ-uniform worst-case context–accuracy law attained by a simple sample-and-hold construction. We further show that trajectory-level supervision identifies the shared executor under suitable task diversity, whereas endpoint supervision does not identify the update rule. Together, these results connect representation, memory, approximation, and learnability in in-context TD, while joint encoder learning and policy control remain open directions.

AI USE STATEMENT

In this work, we used ChatGPT Astra and Claude Fable 5.1 as AI-assisted research tools. Their use included polishing the writing and presentation of the paper, simplifying and improving the exposition of several proofs, assisting with the implementation and execution of part of the experiments, and helping extend the analysis from one-hot state representations to general d-dimensional feature representations. All AI-assisted mathematical arguments, derivations, and proofs were independently reviewed and verified by the authors. The authors take full responsibility for the correctness of all theoretical claims, proofs, experimental results, and the final content of the paper.

ETHICS STATEMENT

This work does not involve human subjects, personal or sensitive data, or experiments that raise specific ethical concerns. The study is primarily theoretical and computational in nature, and all experiments are conducted using controlled settings without involving real-world participants or private information. We therefore do not identify any significant ethical issues associated with this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the theoretical constructions, experimental settings, and implementation procedures in the appendices. The appendices include the necessary architectural specifications, proof details, experimental configurations, task-generation procedures, and evaluation protocols required to reproduce the reported results. In particular, the complete code for all experiments is included in the supplementary material, together with the corresponding scripts and configurations needed to reproduce the main figures, tables, and quantitative results reported in the paper. Where applicable, random seeds, training budgets, hyperparameters, and model configurations are also provided to facilitate independent verification and reproduction of our experiments.

REFERENCES

Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics, 2005.

Jakob Bauer, Kate Baumli, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, Vibhavari Dasagi, Lucy Gonzalez, Karol Gregor, Edward Hughes, Sheleem Kashem, Maria Loks-Thompson, Hannah Openshaw, Jack Parker-Holder, Shreya Pathak, Nicolas Perez-Nieves, Nemanja Rakicevic, Tim Rocktäschel, Yannick Schroecker, Satinder Singh, Jakub Sygnowski, Karl Tuyls, Sarah York, Alexander Zacherl, and Lei M. Zhang. Human-timescale adaptation in an open-ended task space. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1887–1935. PMLR, 2023. URL https://proceedings.mlr.press/v202/bauer23a.html.

Justin A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49:233–246, 2002.

Steven J. Bradtke and Andrew G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.

Baiyuan Chen, Shinji Ito, and Masaaki Imaizumi. Optimal dynamic regret by transformers for non-stationary reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/25c592c5ae46736181075d3f3730709e-Abstract-Conference.html.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.232.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016. URL https://arxiv.org/abs/1611.02779.

Roland W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12: 267–319, 2003.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uREj4ZuGJE.

Alborz Geramifard, Michael Bowling, and Richard S. Sutton. Incremental least-squares temporal difference learning. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pp. 356–361, 2006.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1–2):163–223, 2003.

Jake Grigsby, Linxi Fan, and Yuke Zhu. AMAGO: Scalable in-context reinforcement learning for adaptive agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/7204434dcb9383a1454dc1e97e58ea9c-Abstract-Conference.html.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

B. L. Ho and Rudolf E. Kalman. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, 14:545–548, 1966.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 1658–1677. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5156–5165, 2020.

John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1976. ISBN 978-0-387-90192-3.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hy0a5MMPUv.

Jonathan N. Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8644b61a9bc87bf7844750a015feb600-Abstract-Conference.html.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–6353. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.391.

Haodong Liang and Lifeng Lai. Transformers provably implement in-context reinforcement learning with policy improvement, 2026. URL https://arxiv.org/abs/2605.05755.

Jörg Liesen and Zdeněk Strakoš. *Krylov Subspace Methods: Principles and Analysis*. Oxford University Press, Oxford, 2013. ISBN 9780199655410. doi: 10.1093/acprof:oso/9780199655410.001.0001.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=yN4Wv17ss3.

Chris Lu, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob N. Foerster, Satinder P. Singh, and Feryal Behbahani. Structured state space models for in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 47016–47031, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/92d3d2a9801211ca3693ccb2faa1316f-Abstract-Conference.html.

Amir Moeini, Jiuqi Wang, Jacob Beck, Ethan Blaser, Shimon Whiteson, Rohan Chandra, and Shangtong Zhang. A survey of in-context reinforcement learning, 2025. URL https://arxiv.org/abs/2502.07978.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3d77c6dcc7f143aa2154e7f4d5e22d68-Abstract-Conference.html.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.57.

Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael L. Littman. Analyzing feature generation for value-function approximation. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 737–744, 2007.

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L. Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 752–759, 2008.

Marek Petrik. An analysis of laplacian methods for value function approximation in mdps. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2574–2579, 2007.

Allan Pinkus. *n-Widths in Approximation Theory*. Springer, 1985.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, 1994.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylKikSYDH.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2 edition, 2003. doi: 10.1137/1.9780898718003.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988. doi: 10.1007/BF00115009.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018. ISBN 9780262039246. URL https://mitpress.mit.edu/9780262039246/reinforcement-learning/.

Lloyd N. Trefethen. *Approximation Theory and Approximation Practice*. Society for Industrial and Applied Mathematics, 2013.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2e3435554b430bd8fe92a60c509929a0-Abstract-Conference.html.

Jiuqi Wang, Rohan Chandra, and Shangtong Zhang. Towards provable emergence of in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://arxiv.org/abs/2509.18389.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992. doi: 10.1007/BF00992698.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

Zixuan Xie, Xinyu Liu, Rohan Chandra, and Shangtong Zhang. Convergence and emergence of in-context reinforcement learning with chain of thought, 2026a. URL https://arxiv.org/abs/2605.07123.

Zixuan Xie, Xinyu Liu, Claire Chen, Shuze Daniel Liu, Rohan Chandra, and Shangtong Zhang. Beyond linear attention: Softmax transformers implement in-context reinforcement learning, 2026b. URL https://arxiv.org/abs/2605.07333.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented lms with context compression and selective augmentation. In *International Conference on Learning Representations*, 2024.

Kevin Xu and Issei Sato. On expressive power of looped transformers: Theoretical analysis and enhancement via timestep encoding. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 69613–69646, 2025.

Liu Yang, Kangwook Lee, Robert D. Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *International Conference on Learning Representations*, 2024.

Hengshuai Yao and Zhi-Qiang Liu. Preconditioned temporal difference learning. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1208–1215. ACM, 2008. doi: 10.1145/1390156.1390308.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *International Conference on Learning Representations*, 2025.

CONTENTS OF THE APPENDICES

A Architecture, head library, and end-to-end exactness 16
B Two-head softmax realization of the executors 22
C Memory lower bounds: real registers and bits 23
D Limited context: proofs, the Krylov-anchored hold, and instance-dependent rates 26
E Execution depth 32
F Learnability of the shared executor 33
G Connections to classical statistical and reduced models 35
H Experimental details and additional results 35
I Scope and limitations 44

A Architecture, Head Library, and End-to-End Exactness

This appendix specifies the encoder of Section 4.1 at the level of individual heads and proves Theorem 4.3(a)–(b). All head parameters are fixed for public n, L, a context dimension m where relevant, and, for the sample-and-hold context, the discount γ. Raw records, statistics, Krylov directions, duals, reduced matrices, and sampled iterates are activations. Routes may depend on public quantities and on the current stage, never on task values; only the gates computed in the forward pass decide how many candidate tokens survive release.

A.1 Coverage and Empirical Scope

Full state coverage means $N_i > 0$ for every state; it does not require every transition $i \to j$ to be observed. It makes the empirical rows, reward means, and the visitation normalization in (1) well defined, and expanding the sample update gives the closed form $\widehat{v}_\ell = \sum_{j<\ell} \widehat{A}^j \widehat{r}$ without any independence assumption. Coverage neither implies that $\widehat{P}$ is irreducible nor controls the statistical error of $\widehat{v}_\ell$ relative to the population trajectory; the statistical error is measured separately in Appendix H.4. The exact results and the memory law of Appendix C require no irreducibility. The stationary-norm statements of Section 4.2 add their own positive-stationary-distribution or contraction condition. For data that do not cover every state, a completion or a restricted query domain must be specified before the target is defined; we do not impute missing rows.

A.2 Token Layout and Selector Heads

Every encoder token has the same block layout, so that one selector matrix E_a addresses block a in every token. The blocks are, in order: three width-n blocks id, succ, and slot for the current state, the successor state, and the candidate-position tag; three scalars one, γ, and rew for a constant, the input discount, and the observed reward; the statistics $\mathrm{counts} \in \mathbb{R}^n$, rsum, visits, and inv; the model blocks $\mathrm{Prow}, \mathrm{Arow} \in \mathbb{R}^n$, r, and a work vector $\mathrm{Rvec} \in \mathbb{R}^n$; the Krylov blocks $B, \mathrm{dual}, AB, \mathrm{Hcol}, \mathrm{Hminus}, Y \in \mathbb{R}^n$ with the scalars norm, α, g, and gate. The total width is $13n + 11$. A head is specified by its key block k, query block q, value block v, output block o, and sign σ, meaning $P = \sigma E_o^\top E_v$ and $Q = E_k^\top E_q$ in (2); equivalently $W_K = E_k$, $W_Q = E_q$, $W_V = E_v$, $W_O = \sigma E_o^\top$, so every nonzero projection entry is ± 1. All heads of one layer read the activations before any head writes, which is what makes simultaneous projection subtraction possible; if a block is overwritten, a fixed linear reset clears it first.

A.3 Stage 1: Raw Records to Statistics

Let $E_S = [e_{S_0}, \ldots, e_{S_{N-1}}]$, $E_{S^+} = [e_{S_1}, \ldots, e_{S_N}]$, and $R_+^\top = [R_1, \ldots, R_N]$. Displaying the transpose of the active-token matrix, with the raw tokens in the first block row and the state tokens in the second,

$$\big(U_0^{(1)}\big)^\top = \begin{bmatrix} E_S^\top & E_{S^+}^\top & R_+ & \mathbf{1}_N & 0 \\ I_n & 0 & 0 & \mathbf{1}_n & \gamma \mathbf{1}_n \end{bmatrix} \begin{matrix} \\ \scriptstyle \mathrm{id} \end{matrix} \begin{matrix} \\ \scriptstyle \mathrm{succ} \end{matrix} \begin{matrix} \\ \scriptstyle \mathrm{rew} \end{matrix} \begin{matrix} \\ \scriptstyle \mathrm{one} \end{matrix} \begin{matrix} \\ \scriptstyle \gamma \end{matrix}, \tag{8}$$

where the statistic blocks of the state tokens are initialized to zero. To aggregate raw records into state statistics, every raw token is routed to every state token with $Q_{\mathrm{match}}[\mathrm{id}, \mathrm{id}] = I_n$, so that the score between raw token t and state token i is $e_{S_t}^\top e_i = \mathbf{1}\{S_t = i\}$. Keeping this score, three heads with value–output maps $P_{\mathrm{count}}[\mathrm{counts}, \mathrm{succ}] = I_n$, $P_{\mathrm{reward}}[\mathrm{rsum}, \mathrm{rew}] = 1$, and $P_{\mathrm{visit}}[\mathrm{visits}, \mathrm{one}] = 1$ give

$$u_i^{\mathrm{state}}[\mathrm{counts}] = N_{i,:}, \qquad u_i^{\mathrm{state}}[\mathrm{rsum}] = \sum_{t:S_t=i} R_{t+1}, \qquad u_i^{\mathrm{state}}[\mathrm{visits}] = N_i.$$

Then $u_i^{\mathrm{state}}[\mathrm{inv}] = \mathrm{recip}_0(N_i)$. With state self-routing and $Q_{\mathrm{norm}}[\mathrm{one}, \mathrm{inv}] = 1$, the heads $P_{\mathrm{P}}[\mathrm{Prow}, \mathrm{counts}] = I_n$ and $P_{\mathrm{r}}[r, \mathrm{rsum}] = 1$ produce $u_i^{\mathrm{state}}[\mathrm{Prow}] = N_{i,:}/N_i = \widehat{P}_{i,:}$ and $u_i^{\mathrm{state}}[r] = \widehat{r}_i$, and a further self-routed head with $Q_\gamma[\mathrm{one}, \gamma] = 1$ and $P_{\mathrm{A}}[\mathrm{Arow}, \mathrm{Prow}] = I_n$ gives $u_i^{\mathrm{state}}[\mathrm{Arow}] = \gamma \widehat{P}_{i,:} = \widehat{A}_{i,:}$. The raw tokens are then released. Because the three aggregation heads are sums over raw tokens with a fixed query per state token, they can be evaluated one chunk

of C records at a time, adding each chunk's contribution before the next chunk is loaded, as in the recurrent form of linear attention (Katharopoulos et al., 2020); the result is identical and the live raw storage is $O(Cn)$ rather than $O(Nn)$. Our implementation uses this chunked evaluation.

A.4 The State-Level Context

Before compressing further, note that the n state tokens already form an exact n-dimensional context representation. Take $B = I_n$, $Y = \widehat{A}^\top$, and $s_0 = \widehat{r}$, i.e.,

$$\overline{Z}_0(q) = \begin{bmatrix} I_n & e_q \\ \widehat{A}^\top & 0 \\ \widehat{r}^\top & 0 \end{bmatrix} \in \mathbb{R}^{(2n+1)\times(n+1)}, \tag{9}$$

so that the ith source token contains its state identity e_i, its discounted empirical transition row, and its empirical reward. With the executor (5) and mask $M_n = \mathrm{diag}(I_n, 0)$, Lemma A.2 gives $s_{\ell+1} = [I + (\widehat{A}^\top - I)^\top I]s_\ell = \widehat{A}s_\ell$, hence $s_\ell = \widehat{A}^\ell\widehat{r}$ and $-c_\ell = e_q^\top \sum_{j<\ell} \widehat{A}^j\widehat{r} = \widehat{v}_\ell(q)$. The first $2n$ rows never change, so the sampled RL information and the query remain intact throughout the forward pass, and the empirical value is exactly recoverable after compressing N records into n state tokens. This is the context on which the learnability analysis of Section 4.3 operates.

A.5 Stage 2: Discovering and Compiling Krylov Directions

Let $h = \min\{n, L\}$. Stage 2 retains the n state tokens, introduces one work token, and allocates h candidate tokens with fixed position tags $J_h = [e_1, \ldots, e_h]$ in the block slot:

$$\big(U_0^{(2)}\big)^\top = \begin{array}{c} \\ \mathrm{id} \\ \mathrm{one} \\ \mathrm{Arow} \\ r \\ \mathrm{slot} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{id} & \mathrm{one} & \mathrm{Arow} & r & \mathrm{slot} \end{array} \\ \begin{bmatrix} I_n & \mathbf{1}_n & \widehat{A} & \widehat{r} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & \mathbf{1}_h & 0 & 0 & J_h^\top \end{bmatrix} \end{array}, \tag{10}$$

where the block rows are the state, work, and candidate tokens. Each candidate token j uses B for the current Krylov direction b_j, norm for $d_j = b_j^\top b_j$, α for $\mathrm{recip}_0(d_j)$, dual for ω_j, AB for $\widehat{A}b_j$, Hcol for a column of the reduced operator, Hminus for that column minus its tag, Y for the compiled block, g for the initial reduced coefficient, and gate for the active-direction indicator; these blocks start at zero.

Initialization and one Arnoldi round. A state-to-work head with $Q_{\mathrm{gather}}[r, \mathrm{one}] = 1$ and $P_{\mathrm{gather}}[\mathrm{Rvec}, \mathrm{id}] = I_n$ forms $u^{\mathrm{work}}[\mathrm{Rvec}] = \sum_i e_i\widehat{r}_i = \widehat{r}$, which a work-to-candidate copy ($Q[\mathrm{one}, \mathrm{one}] = 1$, $P[B, \mathrm{Rvec}] = I_n$) writes into the first candidate as $b_1 = \widehat{r}$. At round j, candidate self-attention forms the squared norm ($Q[B, B] = I_n$, $P[\mathrm{norm}, \mathrm{one}] = 1$), then $\alpha_j = \mathrm{recip}_0(d_j)$, the dual $\omega_j = \alpha_j b_j$ ($Q[\mathrm{one}, \alpha] = 1$, $P[\mathrm{dual}, B] = I_n$), and the gate $\eta_j = d_j\alpha_j$ ($Q[\mathrm{norm}, \alpha] = 1$, $P[\mathrm{gate}, \mathrm{one}] = 1$). The operator application and the projection use

$$Q_{\mathrm{op}}[\mathrm{Arow}, B] = I_n, \quad P_{\mathrm{op}}[AB, \mathrm{id}] = I_n, \qquad Q_{\mathrm{proj}}[\mathrm{dual}, B] = I_n, \quad P_{\mathrm{proj}}[B, B] = -I_n,$$

routed from the states to candidate j and from candidates $1, \ldots, j$ to candidate $j+1$ respectively, after a copy of AB from candidate j into the block B of candidate $j+1$. Thus $u_j^{\mathrm{cand}}[AB] = \sum_i e_i(\widehat{A}_{i,:}b_j) = \widehat{A}b_j$ and $b_{j+1} = \widehat{A}b_j - \sum_{i\le j} b_i\,\omega_i^\top\widehat{A}b_j$, which is (3). Since $\eta_j = \mathbf{1}\{b_j \neq 0\}$ and breakdown is permanent (Lemma A.3), $(\eta_1, \ldots, \eta_h) = (1^{k_L}, 0^{h-k_L})$, and token release keeps exactly the first k_L candidates.

Compilation. With $B = [b_1, \ldots, b_{k_L}]$ and $\Omega = [\omega_1, \ldots, \omega_{k_L}]$, heads over all candidate pairs form the reduced operator column by column with $Q_H[\mathrm{dual}, AB] = I_n$ and $P_H[\mathrm{Hcol}, \mathrm{slot}] = I_n$, so that candidate t receives $\sum_i e_i(\omega_i^\top\widehat{A}b_t)$, the tth column of $H = \Omega^\top\widehat{A}B$; a copy and a subtraction of the tag form $\mathrm{Hminus} = \mathrm{Hcol} - \mathrm{slot}$; a copy of B into Y followed by $Q_Y[\mathrm{Hminus}, \mathrm{slot}] = I_n$, $P_Y[Y, \mathrm{dual}] = I_n$ adds $\sum_i \omega_i(H_{ti} - \delta_{ti})$ to candidate t, giving $Y = B + \Omega(H - I)^\top$; and a work-to-candidate head with $Q_g[\mathrm{Rvec}, \mathrm{dual}] = I_n$, $P_g[g, \mathrm{one}] = 1$ gives $g_t = \omega_t^\top\widehat{r}$. The retained context is (4). In total the exact schedule uses $6h + 12$ head calls, two reciprocal overwrites, and explicit release. Seven further heads implement the terminal correction used by Krylov truncation

in Appendix D.6: they gather g into the work vector, advance it through m reduced multiplications by H (key slot, query Rvec, value Hcol into AB, then copied back into Rvec), read the leading coefficient, and subtract $H^m g$ from the last column of H through a gate and a scaled inverse.

Balancing. The implementation used in the experiments additionally rescales each active candidate to unit ℓ_1 norm before it is used, with $\|b_j\|_1$ formed by a ReLU absolute value and a summation head, and applies the projection subtraction twice. Balancing multiplies b_j and divides ω_j by the same positive scalar, so $B\Omega^\top$, H, g, and the executed trajectory are unchanged (Lemma D.10), while the dual entries stay bounded (Appendix H.7).

A.6 Linear Features: Proofs of Theorem 4.4 and Corollary 4.5

Stage 1 for features. Raw token t carries $(\phi_t,\ \phi_t - \gamma\phi'_t,\ R_{t+1})$, where the difference block is a fixed linear map of the input (ϕ_t, ϕ'_t) and the public γ; coordinate token $i \in [d]$ carries e_i with zero-initialized blocks $\mathrm{Acol} \in \mathbb{R}^d$ and b. Two heads routed from every raw token to every coordinate token, with key ϕ and query id, so that the score between raw token t and coordinate token i is $\phi_i(S_t)$, and with values $\phi_t - \gamma\phi'_t$ and R_{t+1}, give

$$u_i[\mathrm{Acol}] = \textstyle\sum_t \phi_i(S_t)(\phi_t - \gamma\phi'_t) = (A_N)_{i,:}, \qquad u_i[b] = \textstyle\sum_t \phi_i(S_t) R_{t+1} = (b_N)_i.$$

A self-routed head with key one, query the public α, value Acol, output Arow, and $\sigma = -1$, followed by adding id, forms $\mathrm{Arow} = e_i - \alpha(A_N)_{i,:} = M_{i,:}$, and the same head on b forms $s_0 = \alpha b_N$. No reciprocal is used in this stage: in the tabular case the visitation normalization $1/N_i$ is what requires recip_0, whereas (6) is un-normalized. As in Appendix A.3, the sums can be streamed one chunk at a time. Stage 2 (Appendix A.5) runs unchanged on (M, s_0) in place of $(\widehat{A}, \widehat{r})$, including its reciprocals $\mathrm{recip}_0(d_j)$ in ω_j and η_j; the reciprocal-free claim of Theorem 4.4(b) concerns Stage 1 only. Nothing in Lemmas A.3 and A.4 uses stochasticity of the operator.

Proof of Theorem 4.4. (a) The increments $\theta_{j+1} - \theta_j = M^j s_0 = D s_j$ lie in $\mathrm{col}(D)$ for $j < L$, so $m \geq \operatorname{rank} D \geq k_L^\phi$; the Krylov context of (b) attains it. *(b)* By the heads above, the coordinate tokens carry (M, s_0) exactly. Lemmas A.3 and A.4 applied to (M, s_0) give the context $[B; Y; g^\top]$ with exactly k_L^ϕ tokens, and the argument of Theorem 4.3(b) gives $s_\ell = H^\ell g$ and $-c_\ell(q) = \phi(q)^\top B \sum_{j<\ell} H^j g = \phi(q)^\top \sum_{j<\ell} M^j s_0 = \phi(q)^\top \theta_\ell$, where the query token carries $\phi(q)$ in its first block. The memory counts are those of Appendix A.14 with d for n; Stage 1 keeps a chunk of raw tokens of width $2d+1$ and d coordinate tokens of width $O(d)$. *(c)* Fix a base pair (M_0, s_0) with $\dim \mathcal{K}_L(M_0, s_0) = k$ in the open set, put $\Delta_0 = [s_0, M_0 s_0, \ldots, M_0^{k-1} s_0]$ and $M_0^k s_0 = \Delta_0 \alpha_0$. For $\theta = (\Delta, \alpha) \in \mathbb{R}^{d\times k} \times \mathbb{R}^k$ near θ_0 define, with the companion matrix $C(\alpha)$ of Lemma C.3, $M(\theta) = M_0 + (\Delta C(\alpha) - M_0\Delta)\Delta^+$ and $s(\theta) = \Delta e_1$. Near θ_0, Δ has full column rank, so Δ^+ is continuous, $M(\theta)\Delta = \Delta C(\alpha)$, and hence $M(\theta)^j s(\theta) = \Delta C(\alpha)^j e_1$: the Krylov dimension is k and the pair stays in the open set for θ close to θ_0. The trajectory determines Δ (the first k increments) and then α (the $(k+1)$st increment, using $L \geq k+1$), so Enc composed with $\theta \mapsto (M(\theta), s(\theta))$ is a continuous injection of an open subset of $\mathbb{R}^{dk+k}$, and invariance of domain gives $M \geq dk + k$ exactly as in the proof of Theorem C.1(1). The context of (b) retains $(2d+1)k$ registers, within the factor $(2d+1)/(d+1) < 2$. □

Proof of Corollary 4.5. Theorem 1 of Wang et al. (2025a) states that with their parameters $(P_\ell^{\mathrm{TD}}, Q_\ell^{\mathrm{TD}})$ and $C_\ell = \alpha I$, the bottom-right entry of layer ℓ equals $-\phi(q)^\top w_\ell$ with $w_0 = 0$ and $w_{\ell+1} = w_\ell + \alpha \sum_j (R_{j+1} + \gamma w_\ell^\top \phi_{j+1} - w_\ell^\top \phi_j)\phi_j$, their $1/N$ being absorbed into α; expanding, $w_{\ell+1} = w_\ell + \alpha(b_N - A_N w_\ell)$, which is (6), so $w_\ell = \theta_\ell$. For the d-token context, Lemma A.2 with $X = I_d$, $Y = M^\top$, $s_0 = \alpha b_N$ gives $s_{\ell+1} = [I + (M^\top - I)^\top] s_\ell = M s_\ell$ and $-c_\ell = \phi(q)^\top \sum_{j<\ell} M^j s_0 = \phi(q)^\top \theta_\ell$. For the Krylov context this is Theorem 4.4(b). For the softmax statement, take the reduced system $(X, H, g) = (I_d, M, s_0)$ or $(B, \Omega^\top M B, \Omega^\top s_0)$ and, in place of the n state query types $(\star, q)$ of Appendix B, the d coordinate query types $(\star, i)$, $i \in [d]$, each carried by one query token. Lemma B.1 with d in place of n gives $u_{\ell,\star}(i) = e_i^\top X \sum_{j<\ell} H^j g = \theta_{\ell,i}$, and $\phi(q)^\top \theta_\ell = \sum_i \phi_i(q)\, u_{\ell,\star}(i)$ is a fixed linear readout of the d query scalars. The readout is applied after the layer because the softmax layer is not linear in the query tag: a feature query cannot be formed inside the layer as a combination of the one-hot tags, which is why the corollary states the

softmax claim for coordinate queries. Admissibility (11) needs a family-uniform gain: on a family with $\|M\|_2 \le \Lambda_M$, the d-token system has $\|X\|_{\max} = 1$ and $\|H - I\|_{\max} \le \|M - I\|_2 \le 1 + \Lambda_M$; for the ℓ_1-balanced Krylov context, $\|B\|_{\max} \le 1$ and $|H_{ij}| = |\omega_i^\top M b_j| \le \|M b_j\|_2 / \|b_i\|_2 \le \sqrt{d}\,\Lambda_M$, since $\|b_j\|_2 \le \|b_j\|_1 = 1$ and $\|b_i\|_2 \ge \|b_i\|_1/\sqrt{d}$, so $\|H - I\|_{\max} \le 1 + \sqrt{d}\,\Lambda_M$. Both families are therefore admissible for $\Lambda = d(2 + \sqrt{d}\,\Lambda_M)$, which is public when Λ_M is; $\Lambda_M = 1 + \alpha\|A_N\|_2$ always works, and it is a constant of the family whenever the features and the step size are bounded. The softmax context retains $(m + 1 + d)(3(d + m + 1) + 1)$ reals, with d query tokens in place of one. The count $N(2d + 1)$ is the size of the prompt $Z_0 \in \mathbb{R}^{(2d+1)\times(N+1)}$ without its query column. □

A.7 Executors

Definition A.1 (Time-invariant and depth-counter executors)**.** An m-dimensional context has static source blocks $X, Y \in \mathbb{R}^{n\times m}$, source scalars $s_0 \in \mathbb{R}^m$, and a query token $(e_q, 0_n, c_0 = 0)$, so that $\overline{Z}_0(q) = [X, e_q;\, Y, 0_n;\, s_0^\top, 0]$. The **time-invariant** executor is the looped layer (5) with mask $M_m = \mathrm{diag}(I_m, 0)$; its output is $-c_\ell$, and it gives $s_{\ell+1} = [I + (Y - X)^\top X]s_\ell$ and $c_{\ell+1} = c_\ell - e_q^\top X s_\ell$. The **depth-counter** executor replaces the source recurrence by fixed public time stamps τ_u and a counter $t_\ell = \ell$: the gate $s_{\ell,u} = 1 - \delta^2\,\mathrm{recip}_0(\delta^2)$ with $\delta = t_\ell + 1 - \tau_u$ equals $\mathbf{1}\{\ell + 1 = \tau_u\}$, and the same query head gives $-c_\ell = \sum_{u:\tau_u\le\ell} e_q^\top x_u$.

In both cases $-c_\ell(q) = e_q^\top X \sum_{j<\ell} s_j$, so every output is an m-dimensional context representation in the sense of Definition 4.1 with $D = X$. The time-invariant executor realizes the recurrences $s_{\ell+1} = K s_\ell$, which is how looped attention without timestep information behaves (Yang et al., 2024); the depth-counter executor adds a timestep signal in the sense of Xu & Sato (2025). The two can share a looped block and write into one query scalar.

Lemma A.2 (Executor identities)**.** *For any query-independent X, Y, s_0, the time-invariant executor satisfies $s_{\ell+1} = [I + (Y - X)^\top X]s_\ell$ and $c_{\ell+1} - c_\ell = -e_q^\top X s_\ell$. For the depth-counter executor, $c_{\ell+1} - c_\ell = -\sum_u \mathbf{1}\{\ell + 1 = \tau_u\} e_q^\top x_u$.*

Proof. The only nonzero entry of P_{TD} is its last diagonal entry, so static rows never change. The score between source tokens i and t is $x_i^\top(y_t - x_t)$ and the source–query score is $-x_i^\top e_q$. Transposing the last-row update gives both identities. The masked query never acts as a source, so the source scalars do not depend on q. For the depth-counter executor, $\delta^2\,\mathrm{recip}_0(\delta^2) = \mathbf{1}\{\delta \ne 0\}$ because $\delta^2 \ge 0$, and the query head is identical. □

A.8 Exactness and Minimality

Lemma A.3 (Primal–dual orthogonality and saturation)**.** *In exact arithmetic, the nonzero vectors produced by (3) are mutually orthogonal, their duals satisfy $\omega_i^\top b_j = \delta_{ij}$, and a zero candidate is followed only by zeros. After $h = \min\{n, L\}$ rounds the nonzero columns span $\mathcal{K}_L$, and their number is k_L.*

Proof. If $b_j \ne 0$, then $d_j > 0$ and $\omega_j = b_j/d_j$. If the preceding columns are nonzero and orthogonal, then for $p \le j$, $b_p^\top b_{j+1} = b_p^\top \widehat{A} b_j - \sum_{i\le j} b_p^\top b_i\, \omega_i^\top \widehat{A} b_j = 0$. A zero b_j gives $d_j = 0$, $\omega_j = 0$, $\widehat{A} b_j = 0$, and hence $b_{j+1} = 0$. Before breakdown, induction gives $b_j = \widehat{A}^{j-1}\widehat{r} + u_j$ with $u_j \in \mathcal{K}_{j-1}$, so the first j columns span $\mathcal{K}_j$. If $b_{k+1} = 0$ after k nonzero columns, the rearranged recurrence $\widehat{A} b_i = b_{i+1} + \sum_{p\le i} b_p \omega_p^\top \widehat{A} b_i$ shows that the first k columns span an invariant subspace containing $\widehat{r}$. If no breakdown occurs, the h columns span $\mathcal{K}_h = \mathcal{K}_L$ when $L \le n$, and $\mathbb{R}^n$ otherwise. The case $\widehat{r} = 0$ gives the zero sequence. □

Lemma A.4 (Forward compilation)**.** *The heads of Appendix A.5 form (4), and every inactive candidate carries a zero payload.*

Proof. With candidate tags p_i, the reduced-column head produces $h_j = \sum_i p_i(\omega_i^\top \widehat{A} b_j)$, whose coordinates form column j of the padded H. The scalar head gives $g_j = \omega_j^\top \widehat{r}$. The compilation

head writes $y_j = b_j + \sum_i \omega_i (h_i - p_i)^\top p_j = b_j + \sum_i \omega_i (H_{ji} - \delta_{ji})$, which on active columns is $B + \Omega(H - I)^\top$. For $j > k_L$ all of $b_j, \omega_j, \widehat{A} b_j$ vanish, so the corresponding row and column of H and the entries g_j, y_j are zero. The gate $d_j \operatorname{recip}_0(d_j)$ is one exactly on nonzero columns, and by Lemma A.3 the gate pattern is a prefix. □

Proof of Theorem 4.3(a) and (b). (a) Let $\{Z_\ell\}$ be an exact m-dimensional context representation with decoder D. Each increment $\widehat{A}^j \widehat{r} = \widehat{v}_{j+1} - \widehat{v}_j = D s_j$ lies in $\operatorname{col}(D)$ for $j < L$, so $\mathcal{K}_L \subseteq \operatorname{col}(D)$ and $m \geq \operatorname{rank} D \geq k_L$. The context (4) attains $m = k_L$ by part (b).

(b) Stage 1 produces $(N_{i,:}, \sum_t R_{t+1} \mathbf{1}\{S_t = i\}, N_i)$ in state token i, since $\sum_t [e_{S_{t+1}}; R_{t+1}; 1](e_{S_t}^\top e_i)$ equals these quantities; multiplying by $\operatorname{recip}_0(N_i) = 1/N_i$ and then by γ yields $(\widehat{A}_{i,:}, \widehat{r}_i)$. Lemmas A.3 and A.4 give (4) with exactly k_L retained tokens. Let $\Pi = B\Omega^\top$, which is an orthogonal projection onto $\mathcal{K}_L$ because $\Omega = B(B^\top B)^{-1}$ for orthogonal columns. Then $Bg = \Pi \widehat{r} = \widehat{r}$, and if $BH^j g = \widehat{A}^j \widehat{r}$ with $j + 1 < L$, then $BH^{j+1} g = \Pi \widehat{A}^{j+1} \widehat{r} = \widehat{A}^{j+1} \widehat{r}$. Lemma A.2 with $(Y - B)^\top B = (H - I)\Omega^\top B = H - I$ gives $s_\ell = H^\ell g$ and $-c_\ell = e_q^\top \sum_{j<\ell} B H^j g = \widehat{v}_\ell(q)$ for $\ell \leq L$. If $k_L < L$, breakdown implies invariance of $\mathcal{K}_L$, and exactness holds at every depth. The memory counts are established in Appendix A.14. □

A.9 Continuity within Rank Strata

Let $\Sigma_k = \{(\widehat{A}, \widehat{r}) : \dim \mathcal{K}_L(\widehat{A}, \widehat{r}) = k\}$. By Lemma A.3, the recurrence (3) depends only on $(\widehat{A}, \widehat{r})$ and uses neither stochasticity nor rationality, so the second stage is exact for every real input.

Lemma A.5 (Continuity within each rank stratum)**.** *Restricted to Σ_k, the second-stage encoder is a rational function whose denominators do not vanish on Σ_k. It is therefore continuous on Σ_k. The same holds with positive ℓ_1 balancing, which replaces b by $b/\|b\|_1$ with $\|b\|_1$ computed by ReLU absolute values.*

Proof. On Σ_k the gates are identically $(1^k, 0^{h-k})$. For $j \leq k$, b_j is a rational function whose denominators are products of $d_1, \ldots, d_{j-1} > 0$, so $\operatorname{recip}_0(d_j) = 1/d_j$ is continuous there. For $j > k$ every quantity is identically zero. The retained quantities H, g, Y are polynomials in these. Balancing multiplies each active b_j by $1/\|b_j\|_1$, which is continuous away from $b_j = 0$. □

A.10 Rank Discontinuity across Strata

For $P = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ and $r_\varepsilon = (1, 1 + \varepsilon)^\top$, $\det[r_\varepsilon, \gamma P r_\varepsilon] = -\gamma\varepsilon(2 + \varepsilon)$. Hence the exact rank is 1 at $\varepsilon = 0$ and 2 for $0 < |\varepsilon| < 1$. No finite composition of continuous maps can output the corresponding binary gate on this family, which is why recip_0 and release are declared primitives. This is consistent with the memory law of Appendix C: an exact rank-adaptive encoder must be discontinuous across strata, yet it can be continuous within each stratum (Lemma A.5). On an input class whose nonzero reciprocal arguments are separated from zero by a margin $\Delta > 0$, the reciprocal can be replaced by a ReLU interpolant plus plateau gates with arbitrarily small output error and identical prefix decisions. Without a margin, tiny residuals yield large duals; positive ℓ_1 balancing removes this growth in our experiments (Appendix H.7), but a tolerance-based rank is then a numerical policy, not the algebraic rank. The numerical-rank bound of Proposition D.3(b) and the threshold certificate of Appendix D.9 give the certified replacements.

A.11 Visit-Weighted Inner Product

Replacing $b^\top b$ by $\langle b, b \rangle_w = b^\top D_w b$ with $w_i = N_i$ requires two more blocks, $\mathrm{wid} = N_i e_i$ in the state tokens and $\mathrm{DB} = D_w b_j$ in the candidates, and four heads: forming wid (key one, query visits, value id, state self-route), weighting B (key id, query B, value wid, states to candidate j), the squared norm (key B, query DB, value one, self-route), and the dual (key one, query α, value DB, self-route). The duals $\omega_j = D_w b_j / \langle b_j, b_j \rangle_w$ are invariant to rescaling w, so no division by N is needed. Lemma A.3 holds for every positive definite inner product, so exactness and Lemma A.5 are unchanged. A *budget-m* encoder runs m rounds instead of h.

A.12 Sample-and-Hold Encoder and Depth-Counter Executor

The schedule $\mathcal{T} = \mathcal{T}_m(\gamma) \cap [1, L]$ is public. Besides the state tokens, the encoder uses one trajectory token with blocks cur (initially 0) and inc, and $m' = |\mathcal{T}| \le m$ sample tokens with a block y. Five heads suffice: an initialization head (key r, query one, value id, states to trajectory, first layer only) sets $\mathrm{inc} = \widehat{r}$; in every layer an accumulation head (key one, query one, value inc, output cur, self-route), a step head (key Arow, query inc, value id, output inc, states to trajectory), and a clearing head (same as the accumulation head with $\sigma = -1$ and output inc) perform $\mathrm{cur} \leftarrow \mathrm{cur} + \mathrm{inc}$ and $\mathrm{inc} \leftarrow \widehat{A}\,\mathrm{inc}$, because heads read synchronously; and at layer τ_u a latch head (key one, query one, value cur, output y, trajectory to sample u) stores $y_u = \widehat{v}_{\tau_u}$. By induction, layer t begins with $\mathrm{cur} = \widehat{v}_t$ and $\mathrm{inc} = \widehat{A}^t\widehat{r}$. A difference head (route $u-1 \to u$, $\sigma = -1$) forms $x_u = y_u - y_{u-1}$. The encoder has $\max \mathcal{T} + 1 \le L + 1$ layers and peak $O(n^2 + nm)$ live scalars; after release the context retains $m'(n+1)$ scalars, x_u and one time stamp per token. Using repeated squaring as in Appendix E.2, the depth can be reduced to $O(\log L)$ at the same peak.

The depth-counter executor of Definition A.1 uses the following fixed sublayers: a token-wise affine map $\delta \leftarrow t + 1 - \tau$; a self-routed head computing δ^2; recip_0; a self-routed head computing $s = 1 - \delta^2\,\mathrm{recip}_0(\delta^2)$; the query head (key x, query e_q, value s, $\sigma = -1$, source tokens to query), which has the form of the TD executor's query head; and the counter increment. If one prefers to avoid recip_0, an unrolled version whose layer ℓ routes only the tokens with $\tau_u = \ell + 1$ uses pure linear attention. When γ is an input rather than public, the schedule can be computed in the network by threshold gates on $1 - \gamma^i$ maintained in a work token; we do not use this variant.

A.13 Krylov-Anchored Encoder

The anchored encoder of Appendix D.1 first runs the budget-k visit-weighted Krylov encoder and packs its k tokens. A work token then accumulates two trajectories in parallel: the TD trajectory, with the heads of Appendix A.12, and the projected trajectory $\widetilde{v}^{\mathrm{G}}_t = B\sum_{i<t} s_i$, using the TD executor's source heads on the packed Krylov tokens together with a head with key s, query one, value B, and target gcur. At layer $k + i_s$ a fixed linear map forms $\mathrm{cur} - \mathrm{gcur} = e_{k+i_s}$, and a latch head stores it; a difference head forms the increments. Execution uses one looped block containing both the TD source and query heads (for the Krylov tokens) and the depth-counter heads (for the residual tokens); all heads write the same query scalar, so the output is $\widetilde{v}^{\mathrm{G}}_\ell + e_{\sigma(\ell)}$. The implementation matches this closed form to 2.8×10^{-15} (Appendix H.1).

A.14 Retained, Peak, and Shared Memory

The stored objects of the exact construction and their treatment are as follows. Raw transition tokens occupy $O(Cn)$ locations when streamed and $O(Nn)$ in a batch, and are released after Stage 1. State tokens and statistics occupy $O(n^2)$ locations and are released after the context is packed. Candidates, duals, images, and the columns of H occupy $O(nh + h^2)$ locations at fixed capacity h (or m) and are released after packing. The retained context B, Y, g occupies $(2n+1)k_L$ locations, of which B is the decoder; the sample-and-hold tokens x_u, τ_u occupy $m(n+1)$; the query token and its accumulator occupy $2n+1$. Scores and workspace occupy $O(n^2+nh)$ temporary locations and are charged at peak. The fixed selectors and routes are shared parameters and are counted separately.

Memory bounds in Theorem 4.3(b). After packing, the context blocks occupy $nk_L + nk_L + k_L + (2n+1) = (2n+1)(k_L+1)$ locations, which is M_{ret} in Definition 4.2: the decoder $D = B$ aliases the first block and is counted once, and c_ℓ accumulates the queried value, so no cumulative coefficient vector is stored. During Stage 1, the state-matching heads need score blocks of size $C \times n$ (or $N \times n$) and output blocks of size $n \times n$, never an all-token square score matrix. After release, the model, candidates, reduced columns, and routed scores fit in $O(n^2 + nh)$ locations. Execution scores have $k_L(k_L+1)$ entries. Hence $M_{\mathrm{peak}} = O(Nn + n^2 + nh)$ in the batch form and $O(Cn + n^2 + nh)$ in the streamed form. The measured peaks of our implementation appear in Appendix H.5. □

B TWO-HEAD SOFTMAX REALIZATION OF THE EXECUTORS

This appendix realizes both executors of Definition A.1 with one looped layer of standard masked softmax attention with two heads. The construction is a *compilation*: the encoder writes log-probability codes into the tokens, and the output map applies a gain that is fixed for the whole task family. It shows that nothing in Sections 4.1–E depends on the attention being linear, and it changes the retained-memory counts by at most a constant factor. It says nothing about what a trained softmax model does; the models trained in this paper are the executor parameters of Appendix H.6 and the bottleneck encoders of Appendix H.9. Earlier softmax compilations of TD are of the same kind (Xie et al., 2026b).

Reduced systems. For this appendix an m-dimensional context is described by a triple (X, H, g) with $X \in \mathbb{R}^{n\times m}$, $H \in \mathbb{R}^{m\times m}$, and $g \in \mathbb{R}^m$; the executor runs $s_{\ell+1} = Hs_\ell$ from $s_0 = g$ and outputs $\widetilde{v}_\ell(q) = e_q^\top X \sum_{j<\ell} s_j$. The Krylov context (4) is $(B, \Omega^\top \widehat{A} B, \Omega^\top \widehat{r})$, with Y no longer needed because H is stored in the codes; the state-level prompt is $(I_n, \widehat{A}, \widehat{r})$. A family of reduced systems is *admissible for the gain* $\Lambda > 0$ if

$$m \max\{\|H - I_m\|_{\max},\ \|X\|_{\max}\} < \Lambda \qquad \text{for every member,} \tag{11}$$

where $\|\cdot\|_{\max}$ is the largest absolute entry.

Codes and tokens. Fix the target types $\mathcal{T} = \{0\} \cup [m] \cup \{(\star, q) : q \in [n]\}$, of size $d_k = 1+m+n$, and a bijection $\iota : \mathcal{T} \to [d_k]$ that depends only on n and m. Put $b^{(0)} = 0$, $b_i^{(j)} = (H - I_m)_{ji}$ for $j \in [m]$, and $b_i^{(\star,q)} = X_{qi}$. For $i \in [m]$ and $t \in \mathcal{T}$ define the probabilities

$$\pi^{\pm}_{i|t} = \frac{1}{2m} \pm \frac{b_i^{(t)}}{2\Lambda}, \qquad \pi^{\pm}_{0|t} = \frac{1}{2} \mp \frac{\mathbf{1}^\top b^{(t)}}{2\Lambda}. \tag{12}$$

Source token $i \in \{0, \dots, m\}$ (index 0 is a null source) carries two key codes $\kappa_i^{\pm} \in \mathbb{R}^{d_k}$ with $(\kappa_i^{\pm})_{\iota(t)} = \log \pi^{\pm}_{i|t}$, a tag $\chi_i = \sqrt{d_k}\, e_{\iota(i)}$, and a scalar u_i; the query token for state q carries the tag $\chi_{(\star,q)}$ and a scalar $u_\star$. The projections are fixed block selectors, $W_Q^{\pm} z = \chi$, $W_K^{\pm} z = \kappa^{\pm}$, $W_V^{\pm} z = u$, and the output map writes $\Lambda(a^+ - a^-)$ into the scalar of the target, where $a^{\pm}$ are the two head outputs. The query is masked as a source; all heads read synchronously. Initialize $u_i = g_i$ for $i \in [m]$ and $u_0 = u_\star = 0$.

Lemma B.1 (Two-head softmax executor)**.** *For a family admissible for Λ, the single set of projections above, which depends only on n, m, and Λ, satisfies $(u_{\ell,1}, \dots, u_{\ell,m}) = H^\ell g$ and $u_{\ell,\star}(q) = e_q^\top X \sum_{j<\ell} H^j g$ for all ℓ. Its context retains $(m+2)\big(3(n+m+1)+1\big)$ reals, at most five times the $(2n+1)(m+1)$ reals of the linear prompt.*

Proof. By (11), $|b_i^{(t)}| < \Lambda/m$ and $|\mathbf{1}^\top b^{(t)}| < \Lambda$, so every probability in (12) is positive, and for each t they sum to one over $i \in \{0, \dots, m\}$. The scaled score of source i at target t is $(\kappa_i^{\pm})^\top \chi_t / \sqrt{d_k} = \log \pi^{\pm}_{i|t}$, so the masked softmax returns exactly $\pi^{\pm}_{i|t}$, and the layer adds $\Lambda \sum_i (\pi^+_{i|t} - \pi^-_{i|t}) u_i = \sum_{i\ge 1} b_i^{(t)} u_i$ to the scalar of target t, using $u_0 = 0$. At the null target this is zero, at source target j it is $((H - I)s)_j$, so $s \leftarrow Hs$, and at the query it is $e_q^\top X s$. Induction and summation give the identities. Each of the $m+1$ source tokens and the query stores two codes, a tag, and a scalar, which is $3d_k + 1$ reals; since $m \le n$, $(m+2)(3(n+m+1)+1) \le (m+2)(6n+4) \le 5\,(2n+1)(m+1)$, using $(m+2)/(m+1) \le 3/2$ and $(6n+4)/(2n+1) \le 10/3$. □

Corollary B.2 (Task-uniform gains)**.** ***(a)*** *For the ℓ_1-balanced Euclidean Krylov encoder with any budget $m \le n$, on every covered n-state batch, every reward, and every $\gamma \in (0,1)$, the family is admissible for $\Lambda_n = 2n(n+1)$. The Galerkin contexts are budget-m instances of the same encoder. For the visit-weighted encoder the bound holds with Λ_n multiplied by $\sqrt{\max_i N_i / \min_i N_i}$.* ***(b)*** *The clocked executor is the same layer with $H = I_m$, with the gates $\mathbf{1}\{\ell + 1 = \tau_u\}$ of Definition A.1 as the source scalars u_u (computed by recip_0 as before), and with X the block of held increments. For sample-and-hold with $\|\widehat{r}\|_\infty \le R$ the family is admissible for $\Lambda_{m,\gamma} = 2mR/(1-\gamma)$, which is public when γ is. The anchored hold uses both heads in one looped block, writing the same query scalar.*

Proof. (a) Balancing gives $\|b_j\|_1 = 1$, so $\|B\|_{\max} \le 1$ and $\|Pb_j\|_\infty \le 1$, hence $\|\widehat{A}b_j\|_2 \le \gamma\sqrt{n}$. The dual is $\omega_i = b_i/\|b_i\|_2^2$ with $\|b_i\|_2 \ge \|b_i\|_1/\sqrt{n} = 1/\sqrt{n}$, so $|H_{ij}| = |\omega_i^\top \widehat{A}b_j| \le \|\widehat{A}b_j\|_2/\|b_i\|_2 \le \gamma n < n$ and $\|H - I\|_{\max} < n + 1$. Then $m \max\{n+1, 1\} \le n(n+1) < \Lambda_n$. With weights $w = N$, $\omega_i = D_w b_i/\langle b_i, b_i\rangle_w$ and Cauchy–Schwarz give $|H_{ij}| \le \|D_w^{1/2}\widehat{A}b_j\|_2/\|D_w^{1/2}b_i\|_2 \le \gamma n\sqrt{\max_i N_i/\min_i N_i}$. (b) With $H = I$ all source targets receive zero, and the query receives $\sum_u u_u\, e_q^\top x_u$, which is the depth-counter update. Each held increment satisfies $\|x_u\|_\infty \le \sum_{t=\tau_{u-1}}^{\tau_u - 1} \gamma^t \|\widehat{r}\|_\infty \le R/(1-\gamma)$, so $m\|X\|_{\max} \le mR/(1-\gamma) < \Lambda_{m,\gamma}$. □

What the construction does and does not show. The codes $\log \pi^{\pm}_{i|t}$ are written by the encoder, which therefore needs a logarithm as a declared primitive in addition to those of Appendix I. The softmax weights are prescribed by the codes rather than computed from a similarity between learned representations, exactly as in earlier softmax compilations of TD; this is why the main text uses linear attention, whose heads compute the same recurrence from the raw statistics without codes. The gain Λ multiplies the rounding error of the probabilities, so the realization is exact in real arithmetic but loses about $\log_{10}\Lambda$ digits in floating point, which Table 3 shows.

Table 3: **Softmax realization checks** on 60 random covered batches ($4 \le n \le 24$, $\gamma \in \{0.5, 0.8, 0.9, 0.95\}$, $10 \le L \le 60$, $\|\widehat{r}\|_\infty = 1$). Errors are maximum absolute deviations divided by $\max_{\ell,s} |\widehat{v}_\ell(s)|$. The gain ratio is the family quantity in (11) divided by the task-uniform gain of Corollary B.2; all 60 batches were admissible.

Check	Max
Exact Krylov context, softmax executor with $\Lambda_n = 2n(n+1)$ vs. direct TD	5.3×10^{-13}
same vs. the linear-attention executor on the same context	5.3×10^{-13}
Budget-m Galerkin context, softmax vs. linear executor	8.9×10^{-13}
Sample-and-hold, softmax clocked head with $\Lambda_{m,\gamma} = 2m/(1-\gamma)$ vs. $\widehat{v}_{\sigma(\ell)}$	6.0×10^{-14}
Krylov-anchored hold, softmax vs. linear executor (family gain from the context)	1.5×10^{-14}
Largest gain ratio, Krylov / clocked	0.11 / 0.25

The linear executor reaches 4×10^{-15} on the same contexts (Table 4); the difference is the gain. The implementation is `softmax_attn.py` and `check_softmax.py` in the code.

C Memory Lower Bounds: Real Registers and Bits

This appendix proves Theorem 4.3(c). The register statement is the following theorem, of which parts (1)–(4) give, respectively, the lower bound $nk + k$, its tightness, the factor-two statement for our context, and the full-rank case. The bit statement is Theorem C.7 in Appendix C.2.

Let Σ_k^+ denote the set of models $(\gamma\widehat{P}, \widehat{r})$ with strictly positive stochastic $\widehat{P}$ and $k_L = k$. A **continuous exact representation of size** M on a set $\mathcal{O} \subseteq \Sigma_k^+$ consists of a continuous map $\mathrm{Enc} : \mathcal{O} \to \mathbb{R}^M$ and an *arbitrary* map Dec with $\mathrm{Dec}(\mathrm{Enc}(\widehat{A}, \widehat{r}), s, \ell) = \widehat{v}_\ell(s)$ for all $s \in [n]$ and $\ell \le L$.

Theorem C.1 (Continuous-encoding memory law)**.** *Let $n \ge 2$, $1 \le k \le \min\{n-1, L-1\}$, and let $\mathcal{O}$ be relatively open in Σ_k^+ and contain a model with $\mathbf{1} \notin \mathcal{K}_L$ (such models exist for every such k).*

1. ***(Lower bound)** Every continuous exact representation on $\mathcal{O}$ has $M \ge nk + k$.*
2. ***(Tightness)** A continuous exact representation of size exactly $nk + k$ exists on all of Σ_k^+.*
3. ***(Attention)** On each rank stratum, the encoder of Theorem 4.3(b) is continuous in $(\widehat{A}, \widehat{r})$, and its $(2n+1)k$ context reals are within the factor $(2n+1)/(n+1) < 2$ of optimal.*
4. ***(Full rank)** If $L \ge n + 1$, every continuous exact representation on a nonempty relatively open subset of Σ_n^+ has $M \ge n^2$. The statistics $(\widehat{A}, \widehat{r})$ are therefore optimal up to a factor $(n+1)/n$, and the second stage helps only when $k_L < n$.*

Proof idea. Near a base chain whose Krylov space avoids $\mathbf{1}$, we build an $(nk+k)$-parameter family: the increments $\Delta = [\widehat{r}, \dots, \widehat{A}^{k-1}\widehat{r}]$ and the coefficients α of $\widehat{A}^k\widehat{r}$ in this basis vary freely. A stochastic matrix is continuously completed via $P(\theta) = P_0 + (T - P_0W)W^+$ with $W = [\Delta, \mathbf{1}]$. The trajectory determines (Δ, α), so Enc restricted to the family is a continuous injection from an open subset of $\mathbb{R}^{nk+k}$, and Brouwer's invariance of domain (Hatcher, 2002) forces $M \geq nk + k$. The remainder of this appendix carries this out.

Scope of the bound. In Theorem 4.3(c) an "open set of covered systems" means a relatively open subset of Σ_k^+, as in Theorem C.1. The bound concerns encoders defined on such a set: it says that no continuous encoder can do with fewer than $nk + k$ registers on *all* models of the set, because the set contains an $(nk + k)$-parameter family whose trajectories are distinct. It is not an instance-level statement, and it does not apply to restricted families on which the trajectory has fewer degrees of freedom. Two examples make the quantifier explicit. If the rewards are restricted to $\widehat{r} = c\mathbf{1}$, then $k_L = 1$ and $\widehat{v}_\ell = c\,\frac{1-\gamma^\ell}{1-\gamma}\mathbf{1}$, so a single register holding c suffices when n and γ are public. For the block-cycle tasks of Section 5, every row of $\widehat{P}$ moves to the next block and the reward is the indicator of block 0, so $\widehat{v}_\ell = \sum_{j<\ell} \gamma^j \mathbf{1}_{B_{-j}}$ depends on the model only through the block partition; here $\mathbf{1} \in \mathcal{K}_L$ and $\widehat{P}$ has zero entries, so the family lies outside the hypotheses, and an encoder for this family alone needs only the n block labels. What the theorem does control is a task-uniform encoder: the Krylov context of Theorem 4.3(b) retains $(2n+1)k_L$ registers on every rank-k_L system, and no continuous encoder that works on all dense rank-k_L systems can retain fewer than $nk_L + k_L$ on all of them. This is the sense in which the lower-bound column of Table 5 is to be read (Appendix H.5). Remark C.5 adds the corresponding caveat on the interface: the bound is stated at the level of the real-valued empirical model, not of a fixed finite batch of records.

C.1 Proof of the Memory Law

Throughout, $\gamma \in (0,1)$, L, and n are fixed. We write Σ_k^+ for the models $(\gamma P, r)$ with P row stochastic and entrywise positive and with $\dim \mathcal{K}_L(\gamma P, r) = k$. Positive P are irreducible and have strictly positive stationary distributions, so they belong to the system class of Section 4.2. The theorem concerns model-level encoders, which is the interface at which the second-stage compression operates (Remark C.5).

Lemma C.2 (A base point)**.** *Let $\eta \in (0,1)$ and let Q be the reflecting random walk on a path, with $Q_{i,i\pm1} = \frac{1}{2}$ and $Q_{11} = Q_{nn} = \frac{1}{2}$. Put $P_0 = (1-\eta)\mathbf{1}\mathbf{1}^\top/n + \eta Q$ and $\psi_j(i) = \cos(\pi j(i-\frac{1}{2})/n)$. Then P_0 is entrywise at least $(1-\eta)/n$, $P_0\mathbf{1} = \mathbf{1}$, and $P_0\psi_j = \eta\cos(\pi j/n)\psi_j$ with $\psi_j \perp \mathbf{1}$ for $1 \leq j \leq n-1$. For $1 \leq k \leq n-1$ and $r_0 = \sum_{j=1}^k \psi_j$, we have $\mathcal{K}_L(\gamma P_0, r_0) = \operatorname{span}\{\psi_1, \dots, \psi_k\}$ whenever $L \geq k$. This space has dimension k and does not contain $\mathbf{1}$.*

Proof. For interior i, the sum-to-product identity gives $(Q\psi_j)(i) = \cos(\pi j/n)\psi_j(i)$. At the two boundaries the reflection gives the same identity, using $\cos(\pi j - x) = (-1)^j \cos x$. The matrix Q is symmetric with distinct eigenvalues $\cos(\pi j/n)$, so $\psi_j \perp \psi_0 = \mathbf{1}$ and $\mathbf{1}\mathbf{1}^\top\psi_j = 0$. The Krylov matrix is $[\psi_1, \dots, \psi_k]V$, where V is the Vandermonde matrix of the distinct nodes $\gamma\eta\cos(\pi j/n)$ and is therefore invertible. The resulting space is orthogonal to $\mathbf{1} \neq 0$. □

Lemma C.3 (An $(nk+k)$-parameter family)**.** *Let $(\gamma P_0, r_0) \in \Sigma_k^+$ with $\mathbf{1} \notin \mathcal{K}_L$ and $k \leq L-1$. Put $A_0 = \gamma P_0$ and $\Delta_0 = [r_0, A_0 r_0, \dots, A_0^{k-1} r_0]$, and choose α_0 with $A_0^k r_0 = \Delta_0 \alpha_0$; it exists because $\mathcal{K}_L = \mathcal{K}_k$ is invariant. For $\theta = (\Delta, \alpha) \in \mathbb{R}^{n\times k} \times \mathbb{R}^k$, let $C(\alpha)$ be the companion matrix ($Ce_i = e_{i+1}$ for $i < k$, $Ce_k = \alpha$) and set*

$$W = [\Delta, \mathbf{1}], \qquad T = [\gamma^{-1}\Delta C(\alpha),\, \mathbf{1}], \qquad P(\theta) = P_0 + (T - P_0W)W^+, \qquad r(\theta) = \Delta e_1.$$

There is a neighborhood $\mathcal{N}$ of $\theta_0 = (\Delta_0, \alpha_0)$ on which the following hold: (a) $P(\theta)$ is continuous in θ, entrywise positive, and row stochastic; (b) $\gamma P(\theta)\Delta = \Delta C(\alpha)$; (c) $\widehat{A}^j r = \Delta C(\alpha)^j e_1$ for all j, so in particular $\widehat{A}^j r = \Delta e_{j+1}$ for $j < k$ and $\widehat{A}^k r = \Delta\alpha$; (d) $(\gamma P(\theta), r(\theta)) \in \Sigma_k^+$.

Proof. At θ_0, W_0 has full column rank because Δ_0 does and $\mathbf{1} \notin \operatorname{col}\Delta_0$. Full column rank is an open condition, so near θ_0 the map $W^+ = (W^\top W)^{-1}W^\top$ is continuous and $W^+W = I$. Hence $P(\theta)W = T$, which gives (b) and $P\mathbf{1} = \mathbf{1}$. At θ_0, $A_0\Delta_0 = \Delta_0 C(\alpha_0)$ and $P_0\mathbf{1} = \mathbf{1}$, so $T_0 = P_0W_0$

and $P(\theta_0) = P_0 > 0$; positivity persists nearby. Part (c) follows from (b) by induction. For (d), $\mathcal{K}_L \subseteq \operatorname{col}\Delta$ by (c), and $\mathcal{K}_L$ contains $\Delta e_1, \ldots, \Delta e_k$ because $k \le L$; Δ keeps full column rank nearby. □

Proof of Theorem C.1. **(1)** Take a model in $\mathcal{O}$ with $\mathbf{1} \notin \mathcal{K}_L$ and let $\iota(\theta) = (\gamma P(\theta), r(\theta))$ be the family of Lemma C.3. Since ι is continuous, $\iota(\theta_0) \in \mathcal{O}$, $\iota(\mathcal{N}) \subseteq \Sigma_k^+$, and $\mathcal{O}$ is relatively open, we may shrink $\mathcal{N}$ so that $\iota(\mathcal{N}) \subseteq \mathcal{O}$. By Lemma C.3(c), the trajectory $\Phi(\theta) = (\widehat{v}_\ell(s))_{s,\ell \le L}$ determines the increments $\widehat{v}_{j+1} - \widehat{v}_j = \Delta e_{j+1}$ for $j < k$ and $\widehat{v}_{k+1} - \widehat{v}_k = \Delta\alpha$ (using $L \ge k+1$). Hence it determines Δ and, since Δ has full column rank, α. So Φ is injective on $\mathcal{N}$. Because $\mathrm{Dec}(\mathrm{Enc}(\iota(\theta)), \cdot, \cdot) = \Phi(\theta)$, the map $F = \mathrm{Enc} \circ \iota$ is a continuous injection of the open set $\mathcal{N} \subseteq \mathbb{R}^{nk+k}$ into $\mathbb{R}^M$. If $M < nk + k$, compose with the inclusion $x \mapsto (x, 0) \in \mathbb{R}^{nk+k}$. The result is a continuous injection between open subsets of the same dimension, so its image is open by invariance of domain (Hatcher, 2002). But the image lies in the hyperplane $\mathbb{R}^M \times \{0\}$, which has empty interior, a contradiction.

(2) On Σ_k^+ set $\mathrm{Enc}(\widehat{A}, \widehat{r}) = (\Delta, \alpha)$ with $\Delta = [\widehat{r}, \ldots, \widehat{A}^{k-1}\widehat{r}]$ and $\alpha = (\Delta^\top\Delta)^{-1}\Delta^\top\widehat{A}^k\widehat{r}$. Before saturation the Krylov spaces grow strictly, so Δ has full column rank on Σ_k and α is continuous. Since $L \ge k+1$, $\mathcal{K}_{k+1} = \mathcal{K}_k$, so $\widehat{A}^k\widehat{r} = \Delta\alpha$ and $\widehat{A}\Delta = \Delta C(\alpha)$. Then $\mathrm{Dec}(\cdot, s, \ell) = e_s^\top \Delta \sum_{j<\ell} C(\alpha)^j e_1$ is exact.

(3) Continuity is Lemma A.5, and exactness is Theorem 4.3. Each source token stores b_j, y_j, g_j, i.e., $2n+1$ reals, and $(2n+1)k/(nk+k) = (2n+1)/(n+1)$.

(4) The stratum Σ_n^+ is nonempty: take P_0 from Lemma C.2 and $r_0 = \sum_{j=0}^{n-1} \psi_j$, whose n eigenvalues are distinct. Fix any model $(\gamma P_1, r_1)$ in the given open set. Parameterize positive stochastic matrices near P_1 by their first $n-1$ columns ($n(n-1)$ parameters) and rewards near r_1 (n parameters). The rank-n condition is open. The trajectory determines $r = \widehat{v}_1$ and $\widehat{A} = [\delta_1, \ldots, \delta_n][\delta_0, \ldots, \delta_{n-1}]^{-1}$ with $\delta_j = \widehat{v}_{j+1} - \widehat{v}_j$, so it is injective in these n^2 parameters. The argument of (1) gives $M \ge n^2$. □

Remark C.4 (The case $L = k$ and gauge counting). If $L = k$, fix $\alpha = \alpha_0$; the same argument gives $M \ge nk$, and storing Δ attains it. A numerical realization (B, H, g) has $nk + k^2 + k$ entries but carries a $\mathrm{GL}(k)$ gauge freedom $(B, H, g) \mapsto (BG, G^{-1}HG, G^{-1}g)$. Removing its k^2 dimensions leaves exactly $nk + k$, so the bound is tight modulo gauge.

Remark C.5 (Why continuity, and the raw-data interface). Without continuity, a set-theoretic bijection $\mathbb{R}^d \to \mathbb{R}$ (e.g., interleaving decimal digits) gives $M = 1$. Linear decoders exclude this pathology by fiat, whereas continuity of the encoder excludes it naturally and permits arbitrary decoders. Empirical models from finite batches have rational entries, so continuity is stated at the model level, where the second stage operates. If one instead fixes the observed transitions and lets only rewards vary continuously, the same argument gives only $M \ge k$. Lower bounds for record-level interfaces need counting arguments such as the bit law of Appendix C.2, which is consistent with Theorem C.1: each real carries $\Theta(\log(1/\epsilon))$ bits.

C.2 Offline Retained-Bit Complexity

This subsection concerns unrestricted offline codes, not the context of Theorem 4.3(b). It complements Theorem C.1: there, reals are counted under continuity; here, bits are counted with arbitrary encoders and decoders.

Definition C.6 (Offline coding problem)**.** For public n, $L \ge 1$, $\epsilon > 0$, and $\gamma \in (0, 1)$, let $\mathfrak{M}_n = \{(P, r) : P \ge 0,\ P\mathbf{1} = \mathbf{1},\ \|r\|_\infty \le 1\}$. A B-bit code consists of $\mathrm{Enc} : \mathfrak{M}_n \to \{0,1\}^B$ and a decoder satisfying $\max_s \sup_{\ell \le L} |\mathrm{Dec}(\mathrm{Enc}(P, r), s, \ell) - e_s^\top \sum_{j<\ell} (\gamma P)^j r| \le \epsilon$. Let $B^*_{n,L}(\epsilon, \gamma)$ be the minimum such B.

Theorem C.7 (Retained-bit law)**.** *For fixed γ and sufficiently small ϵ, uniformly in n and L (including $L = \infty$), $B^*_{n,L}(\epsilon, \gamma) = \Theta_\gamma(n \min\{n, L, \ln(1/\epsilon)\} \ln(1/\epsilon))$.*

Proof sketch with explicit γ-dependence. Let $t = \ln(1/\epsilon)$, $a = \ln(1/\gamma)$, and $V = (1-\gamma)^{-1}$. *Upper bound.* Let $J_0 = \lceil \ln(4V/\epsilon)/a \rceil$ and $J = \min\{L, J_0\}$. The *prefix code* quantizes $\widehat{v}_1, \ldots, \widehat{v}_J$ to accuracy $\epsilon/4$ with $\lceil \log_2(3 + 4V/\epsilon) \rceil$ bits per coordinate and holds $\widehat{v}_J$ afterwards, since the tail

satisfies $V\gamma^J \le \epsilon/4$. The *model code* quantizes rewards to accuracy $\epsilon(1-\gamma)/4$ and each transition row to ℓ_1-accuracy $\epsilon(1-\gamma)^2/(4\gamma)$ using integer rows summing to M, since $\|p - k/M\|_1 \le n/M$. The perturbation bound $\sup_\ell \|\widetilde{v}_\ell - v_\ell\|_\infty \le \delta_r/(1-\gamma) + \gamma\delta_P/(1-\gamma)^2$ then gives accuracy $\epsilon/2$. Using the prefix code if $J < n$ and the model code otherwise yields $O(n \min\{n, J\} \ln(V/\epsilon))$ bits. *Lower bound.* Packing rewards with $P = I$ gives $n \log_2(1/(4\epsilon))$ bits. For the delayed-reward family, use $p = n - h - 1$ source states, a deterministic delay chain $c_h \to \cdots \to c_1 \to c_0$ rewarded at c_1, and transition probabilities $\theta_{ij} \in [0, 1/(2h)]$ from source i into c_j. This family satisfies $\widehat{v}_{j+1}(u_i) - \widehat{v}_j(u_i) = \gamma^j \theta_{ij}$. Choose $h = \min\{\lfloor (n-1)/2 \rfloor, L - 1, \lfloor t/(8(1+a)) \rfloor\}$ and pack each θ_{ij} on a grid of spacing $8\epsilon\gamma^{-j}$. For $\epsilon \le \exp(-\max\{16, 8(1+a)\})$, this gives $B^* \ge n \min\{n, L, t\} t/(96(1+a)\ln 2)$. □

Remark C.8 (Dependence on γ). The constants are explicit. The upper bound's effective depth is $J_0 \approx t/(1-\gamma)$ as $\gamma \to 1$, whereas the lower bound's delay length is $h \approx t/8$. For $\ln(1/\epsilon) < \min\{n, L\}$, the two therefore differ by a factor of order $1/(1-\gamma)$ in the depth term. Closing this factor, the analogue of Theorem 4.6 for bits, is open. The law ranges over all real stochastic models; a fixed finite batch has finite information, which can saturate as $\epsilon \downarrow 0$. For comparison, coding only the fixed point $(I - \gamma P)^{-1} r$ needs $\Theta_\gamma(n \ln(1/\epsilon))$ bits.

D Limited Context: Proofs, the Krylov-Anchored Hold, and Instance-Dependent Rates

This appendix proves Theorem 4.6 and states and proves the results referred to in Section 4.2: the suboptimality of Krylov moment matching, the Krylov-anchored hold, and instance-dependent rates. Throughout, an m-dimensional context is one of Definition 4.1, and $\mathfrak{C}_m$ is the corresponding class of trajectories.

D.1 Statements

Proposition D.1 (Moment matching is not minimax)**.** ***(a)*** *Krylov truncation, i.e., the terminal-corrected context that equals TD stopped after m steps, has worst-case error exactly $\gamma^m/(1-\gamma)$ over systems with $k_L > m$. This exceeds $e^2\Psi_m$ by the factor $\min\{m, 1/(1-\gamma)\}/e^2$.* ***(b)*** *Krylov projection (the Galerkin context $\widetilde{v}_\ell = B\sum_{j<\ell} H^j g$ obtained from the budget-m encoder with Euclidean or visit-weighted Arnoldi) has error at least $0.657\,\gamma^m/\sqrt{1-\gamma} \ge 0.089\, e^2\Psi_m/\sqrt{1-\gamma}$ on a directed ring with one rewarded state, whenever $\gamma \ge \frac{1}{2}$ and $m \ge 1/(1-\gamma)$.*

Both encoders spend their whole budget on the *first* m increments, whereas the optimal context must also represent late ones. The loss comes from moment matching rather than from time invariance: on the ring of (b), numerically optimized time-invariant recurrences come within a factor 1.8 of the certified lower bound, closer than sample-and-hold itself (Appendix D.10); whether a uniformly computable time-invariant encoder attains $\Theta(\Psi_m)$ remains open. Moment matching does, however, adapt to spectral structure that sample-and-hold ignores, which motivates the following hybrid.

Krylov-anchored hold. Split $m = k + j$ with $j = 0$ if $\gamma < \frac{1}{2}$ and $j = \min\{J, \lfloor m/4 \rfloor\}$ otherwise, where $J = \lfloor 1/(1-\gamma) \rfloor$. Run k rounds of visit-weighted Arnoldi (Appendix A.11), and let $\widetilde{v}^{\mathrm{G}}$ be the resulting projected trajectory. With the depth-counter executor, hold the residual $e_\ell = \widehat{v}_\ell - \widetilde{v}^{\mathrm{G}}_\ell$ at the j depths $k + i_s$ of $\mathcal{T}_j(\gamma)$. The output is $\widetilde{v}_\ell = \widetilde{v}^{\mathrm{G}}_\ell + e_{\sigma(\ell)}$, with $e_{\sigma(\ell)} = 0$ before the first sample. One looped block contains both the TD heads and the depth-counter heads (Appendix A.13).

Theorem D.2 (Minimax optimality and instance adaptivity)**.** *Suppose the encoder weights w satisfy $\|\widehat{P}x\|_w \le \|x\|_w$ (e.g., $w = \mu$, or visit counts of a balanced batch). Let $\mathcal{E}^{\mathrm{G}}_k = \sup_\ell \|\widehat{v}_\ell - \widetilde{v}^{\mathrm{G}}_\ell\|_w / \|\widehat{r}\|_w$. Then the Krylov-anchored hold is an m-dimensional context with $k + j = m$ and satisfies*

$$\sup_{\ell \ge 0} \frac{\|\widehat{v}_\ell - \widetilde{v}_\ell\|_w}{\|\widehat{r}\|_w} \;\le\; \min\big\{2e^2\,\Psi_m(\gamma),\; 2\,\mathcal{E}^{\mathrm{G}}_k\big\}, \qquad k \ge \lceil 3m/4 \rceil.$$

Its retained memory is $(2n+1)(k+1) + j(n+1) \le (2n+1)(m+1)$, and it is exact whenever $k_L \le k$.

Proposition D.3 (Instance-dependent rates and robustness). ***(a) Reversible chains.*** *If $N_{ij} = N_{ji}$, then visit-weighted Krylov projection with context dimension m satisfies $\mathcal{E}_m^{\mathrm{G}} \le 4\beta^m/((1-\beta)\sqrt{1-\gamma^2}) \le 8\beta^m/(1-\gamma^2)$ with $\beta = \gamma/(1+\sqrt{1-\gamma^2})$. The anchored hold is therefore also at most $16\beta^{\lceil 3m/4 \rceil}/(1-\gamma^2)$.* ***(b) Numerical rank.*** *Let σ_i be the singular values of $V_L = D_w^{1/2}[\widehat{v}_1, \dots, \widehat{v}_L]$. Every m-dimensional context in $\mathfrak{C}_m$ has $\sup_{\ell \le L} \|\widehat{v}_\ell - \widetilde{v}_\ell\|_w \ge \sigma_{m+1}/\sqrt{L}$, while the best m-dimensional subspace achieves σ_{m+1}. Moreover, $|\sigma_i(V_L) - \sigma_i(V_L')| \le \sqrt{L}\sup_\ell \|\widehat{v}_\ell - \widehat{v}_\ell'\|_w$, and the latter is $O\big(\delta_r/(1-\gamma) + \delta_A\|\widehat{r}\|_w/(1-\gamma)^2\big)$ under model perturbations of size (δ_A, δ_r).*

Part (a) is a square-root acceleration: the effective horizon becomes $1/\sqrt{1-\gamma}$ instead of $1/(1-\gamma)$, because the Chebyshev coefficients of every partial sum $\sum_{j<\ell} x^j$ on $[-\gamma, \gamma]$ are nonnegative and dominated by those of $1/(1-x)$. Part (b) replaces the discontinuous exact rank by the stable ε-rank $\#\{i : \sigma_i > \varepsilon\sqrt{L}\}$, a lower bound on the context dimension that *any* method needs for accuracy ε; it is what makes compression meaningful on sampled tasks whose exact rank is full (Appendix H.4). A threshold gate on Arnoldi residuals gives a certified adaptive rank as well (Appendix D.9).

D.2 Three Tools

Lemma D.4 (Contraction). *If P is row stochastic and $\mu > 0$ satisfies $\mu^\top P = \mu^\top$, then $\|Px\|_\mu \le \|x\|_\mu$. For every row-stochastic P, $\|Px\|_\infty \le \|x\|_\infty$.*

Proof. By Jensen's inequality, $(\sum_j P_{ij}x_j)^2 \le \sum_j P_{ij}x_j^2$. Weighting by μ_i and summing gives $\sum_j \mu_j x_j^2$. The sup-norm statement is immediate. □

Lemma D.5 (Error controls block-increment width). *If $\widetilde{v} \in \mathfrak{C}_m$ and $\sup_\ell \|\widehat{v}_\ell - \widetilde{v}_\ell\| \le \varepsilon$, then $U = \mathrm{col}(D)$ has $\dim U \le m$ and satisfies $\mathrm{dist}(\widehat{v}_b - \widehat{v}_a, U) \le 2\varepsilon$ for all $0 \le a < b$.*

Proof. $\widetilde{v}_b - \widetilde{v}_a = D\sum_{a \le j < b} s_j \in U$, and the triangle inequality gives $\|(\widehat{v}_b - \widehat{v}_a) - (\widetilde{v}_b - \widetilde{v}_a)\| \le 2\varepsilon$. □

Lemma D.6 (Width of orthogonal geometric families). *Let $x_0, \dots, x_{T-1}$ be orthogonal in an inner-product norm with $\|x_t\| = a_0\rho^t$, $\rho \in (0,1)$, and $T \ge m + 1 + 1/(1-\rho^2)$. Then every subspace U of dimension at most m satisfies $\max_t \mathrm{dist}(x_t, U) \ge a_0\rho^m \max\{\sqrt{1-\rho^2}, \rho^{1/(1-\rho^2)}\} \ge a_0\rho^m/2$.*

Proof. Let Π be the orthogonal projection onto U, and suppose all distances are at most ε. Put $\eta = \varepsilon/a_0$. The unit vectors $\hat{x}_t$ are orthonormal, so $\sum_t \|\Pi\hat{x}_t\|^2 \le \mathrm{tr}\,\Pi \le m$. Each term also satisfies $\|\Pi\hat{x}_t\|^2 = 1 - \mathrm{dist}(x_t, U)^2/\|x_t\|^2 \ge 1 - \eta^2\rho^{-2t}$. Summing over $t \le m$ gives $m \ge m + 1 - \eta^2\sum_{t \le m}\rho^{-2t}$, hence $\eta \ge \rho^m\sqrt{1-\rho^2}$. Next let $a = 1/(1-\rho^2)$ and $T' = \lceil m + a \rceil \le T$. If $\eta \le \rho^{T'-1}$, then $\sum_{t<T'}(1-\eta^2\rho^{-2t}) = T' - \eta^2\frac{\rho^{-2T'}-1}{\rho^{-2}-1} > T' - \eta^2\rho^{-2(T'-1)}a \ge T' - a \ge m$, a contradiction. Thus $\eta > \rho^{T'-1} \ge \rho^{m+a}$. For the final inequality: if $\rho^2 \le 3/4$ then $\sqrt{1-\rho^2} \ge 1/2$. Otherwise $u = 1 - \rho^2 < 1/4$, and $\ln(1-u) \ge -u/(1-u)$ gives $\rho^{1/u} = \exp(\ln(1-u)/(2u)) \ge e^{-2/3} > 1/2$. □

D.3 The Lower Bound $\mathfrak{E}_m \ge \Psi_m/64$

Both instances are directed rings, $(Pv)(x) = v(x+1)$ modulo n. The empirical model is realized exactly by a covered batch containing one transition per state, and the stationary distribution μ is uniform.

Instance I (single rewarded state). Let $r = \sqrt{n}\,e_0$, so $\|r\|_\mu = 1$. The increments $\widehat{A}^t r = \gamma^t\sqrt{n}\,e_{-t}$ $(t < n)$ are orthogonal with norms γ^t. If $n \ge m + 1 + 1/(1-\gamma^2)$, Lemmas D.5 (with $b = a+1$) and D.6 give $2\varepsilon \ge \gamma^m/2$, i.e., every $\widetilde{v} \in \mathfrak{C}_m$ has error at least $\gamma^m/4$.

Instance II (rewarded arc). For integers $w, T \geq 1$, let $n = (2T+2)w$ and $r = \sqrt{n/w}\,\mathbf{1}_{[0,w)}$, so $\|r\|_\mu = 1$. The block increments $\tau_i = \widehat{v}_{(i+1)w} - \widehat{v}_{iw} = \sum_{j=iw}^{(i+1)w-1} \gamma^j P^j r$ are supported on arcs of length $2w-1$, because $P^j r$ is supported on $[-j, -j+w)$. The even blocks τ_{2t}, $t < T$, therefore have pairwise disjoint supports inside the ring and are orthogonal. Moreover, τ_{2t} is a shifted copy of $\gamma^{2tw}\tau_0$. With $S_w = \sum_{j<w} \gamma^j$ and $\mathcal{S} = \operatorname{supp} \tau_0$, Cauchy–Schwarz gives

$$\|\tau_0\|_\mu \;\geq\; \frac{\langle \tau_0, \mathbf{1}_{\mathcal{S}} \rangle_\mu}{\|\mathbf{1}_{\mathcal{S}}\|_\mu} = \frac{n^{-1}\sqrt{n/w}\; w S_w}{\sqrt{(2w-1)/n}} = S_w \sqrt{\frac{w}{2w-1}} \;\geq\; \frac{S_w}{\sqrt{2}}.$$

Applying Lemma D.6 with $a_0 = \|\tau_0\|_\mu$, $\rho = \gamma^{2w}$, and $T \geq m + 1 + 1/(1-\gamma^{4w})$, together with Lemma D.5 for $(a, b) = (2tw, (2t+1)w)$, shows that every $\widetilde{v} \in \mathfrak{C}_m$ has error at least $S_w \gamma^{2wm}/(4\sqrt{2})$.

Proof of the lower bound in Theorem 4.6. Let $q = 1/(m(1-\gamma))$, so that $\Psi_m = \gamma^m \max\{1, q\} \leq \max\{1, q\}$. If $q \leq 16$, then $\Psi_m \leq 16\gamma^m$, and Instance I gives error at least $\gamma^m/4 \geq \Psi_m/64$. If $q > 16$, then $1 - \gamma < 1/16$. Take $w = \lfloor q/4 \rfloor \geq 4$, so that $w > q/4 - 1 > 3q/16$, $w(1-\gamma) \leq 1/(4m) \leq 1/4$, and $2wm(1-\gamma) \leq 1/2$. Bernoulli's inequality gives $S_w \geq \sum_{j<w}(1 - j(1-\gamma)) \geq w(1 - w(1-\gamma)/2) \geq 7w/8$. From $\ln\gamma \geq -(1-\gamma)/\gamma \geq -\frac{16}{15}(1-\gamma)$ we get $\gamma^{2wm} \geq e^{-8/15}$. Instance II then gives error at least

$$\frac{1}{4\sqrt{2}} \cdot \frac{7}{8} \cdot \frac{3q}{16} \cdot e^{-8/15} \;\geq\; \frac{q}{59} \;\geq\; \frac{\Psi_m}{59}.$$

Both cases give at least $\Psi_m/64$. Taking the infimum over $\widetilde{v} \in \mathfrak{C}_m$ and the supremum over systems proves the claim. Both instances have $k_L > m$, so the bound also holds for systems below the exact threshold. □

D.4 The Upper Bound and the Sample-and-Hold Context

Lemma D.7 (Sample-and-hold error)**.** *Let $\|\cdot\|$ be any norm with $\|\widehat{P}x\| \leq \|x\|$. For the schedule with parameters (k, j), every $\ell \geq 0$ satisfies $\|\widehat{v}_\ell - \widehat{v}_{\sigma(\ell)}\| < \gamma^k \|\widehat{r}\|/((j+1)(1-\gamma))$.*

Proof. For $\ell \leq k$ the error is zero. Let $\ell = k + i$ and $s = \max\{s : i_s \leq i\}$ (with $i_0 = 0$), so that $\sigma(\ell) = k + i_s$. Then $\|\widehat{v}_\ell - \widehat{v}_{\sigma(\ell)}\| \leq \sum_{t=k+i_s}^{k+i-1} \gamma^t \|\widehat{r}\| = \gamma^k(\gamma^{i_s} - \gamma^i)\|\widehat{r}\|/(1-\gamma)$. By definition $1 - \gamma^{i_s} \geq s/(j+1)$. If $s < j$, then $i < i_{s+1}$ gives $1 - \gamma^i < (s+1)/(j+1)$; if $s = j$, then $1 - \gamma^i < 1$. In both cases $\gamma^{i_s} - \gamma^i < 1/(j+1)$. □

Proof of the upper bound in Theorem 4.6. The increments x_u span at most m dimensions and the executor gives $D = X$, so the sample-and-hold output lies in $\mathfrak{C}_m$. It remains to bound $F = \gamma^k/((j+1)(1-\gamma))$ by $e^2\Psi_m$; Lemma D.4 supplies the required norms. If $\gamma < \frac{1}{2}$, then $F = \gamma^m/(1-\gamma) \leq 2\gamma^m \leq 2\Psi_m$. If $\gamma \geq \frac{1}{2}$ and $m \leq J$, then $q \geq 1$, and $F < 1/(m(1-\gamma)) = \gamma^{-m}\Psi_m \leq \gamma^{-1/(1-\gamma)}\Psi_m \leq e^2\Psi_m$, using $-\ln\gamma \leq (1-\gamma)/\gamma \leq 2(1-\gamma)$. Otherwise $m \geq J+1 > 1/(1-\gamma)$, so $\Psi_m = \gamma^m$, and $F < \gamma^{m-J} \leq \gamma^{-J}\Psi_m \leq e^2\Psi_m$. □

Remark D.8 (The schedule must depend on γ). For constant reward $\widehat{r} = \mathbf{1}$, $\widehat{v}_\ell = \frac{1-\gamma^\ell}{1-\gamma}\mathbf{1}$, and any schedule $\mathcal{T}$ has error at least $\gamma^{\max\mathcal{T}}/(1-\gamma)$ as $\ell \to \infty$. If $\mathcal{T}$ is fixed independently of γ and $\gamma \geq \frac{1}{2}$ is close enough to one that $\max\mathcal{T} \leq 1/(1-\gamma)$, this is at least $e^{-2}/(1-\gamma) \geq e^{-2}m\Psi_m$. Dyadic schedules $\{1, 2, 4, \ldots\}$ and uniform schedules are therefore off by a factor $\Omega(m)$. The quantile schedule places samples where the remaining discounted mass decreases by equal fractions.

D.5 Finite Depth

This subsection states and proves the finite-depth version of Theorem 4.6 summarized in Section 4.2. Throughout, $\mathfrak{C}_m$ is the depth-L class of Definition 4.1 and the error of a context is $\max_{0\leq\ell\leq L} \|\widehat{v}_\ell - \widetilde{v}_\ell\|_\mu/\|\widehat{r}\|_\mu$, so that $\mathfrak{E}_{m,L}(\gamma)$ is the worst case of its infimum over $\mathfrak{C}_m$. Lemma D.5 applies verbatim to block increments with $b \leq L$, since its proof uses only the errors at depths a and b.

Proposition D.9 (Finite depth). *For all $m \geq 1$, $\gamma \in (0,1)$ and $L \geq 1$, with $\Psi_{m,L}(\gamma) := \gamma^m \max\{1, (1-\gamma^L)/(m(1-\gamma))\} \leq \Psi_m(\gamma)$: **(a)** $\mathfrak{E}_{m,L}(\gamma) \leq \mathfrak{E}_m(\gamma)$, and $\mathfrak{E}_{m,L}(\gamma) = 0$ if and only if $m \geq L$. **(b)** $\mathfrak{E}_{m,L}(\gamma) \leq e^2 \Psi_{m,L}(\gamma)$, attained by sample-and-hold with sampled depths at the quantiles of the discounted mass through depth L. **(c)** $\mathfrak{E}_{m,L}(\gamma) \geq \gamma^m/(2\sqrt{m+1})$ if $L \geq m+1$, and $\mathfrak{E}_{m,L}(\gamma) \geq \Psi_m(\gamma)/64$ if $L \geq m + 4/(1-\gamma)$. **(d)** If $\gamma \geq \frac{1}{2}$ and $8m \leq L \leq 1/(1-\gamma)$, then $L/(400m) \leq \mathfrak{E}_{m,L}(\gamma) \leq e^2 L/m$.*

Proof. (a) Restricting a context of depth $L' \geq L$, or of unbounded depth, to the depths $\ell \leq L$ gives a context of depth L with the same decoder, and a maximum over fewer depths is smaller; hence $\mathfrak{E}_{m,L}$ is nondecreasing in L and $\mathfrak{E}_{m,L} \leq \mathfrak{E}_m$. If $m \geq L$, the Krylov context of Theorem 4.3(b) is exact with $k_L \leq L \leq m$ tokens (padded with zero tokens), so $\mathfrak{E}_{m,L} = 0$. If $m < L$, part (c) gives $\mathfrak{E}_{m,L} \geq \gamma^m/(2\sqrt{m+1}) > 0$.

(b) Keep the split (k, j) of Section 4.2; if $k \geq L$ the context is exact. Otherwise replace the sampled depths by the quantiles of the discounted mass through depth L, $\mathcal{T}_{m,L}(\gamma) = \{1, \dots, k\} \cup \{k + i_s : 1 \leq s \leq j\}$ with $i_s = \min\{i \geq 0 : 1 - \gamma^i \geq \frac{s}{j+1}(1-\gamma^{L-k})\}$, so that $i_j \leq L - k$; the encoder and executor are those of Appendix A.12 with these stamps, and the output lies in $\mathfrak{C}_m$ as in the proof of Theorem 4.6. For $\ell \leq k$ the error is zero. For $\ell = k + i \leq L$ let $s = \max\{s : i_s \leq i\}$; as in Lemma D.7, $\|\widehat{v}_\ell - \widehat{v}_{\sigma(\ell)}\|_\mu \leq \gamma^k(\gamma^{i_s} - \gamma^i)\|\widehat{r}\|_\mu/(1-\gamma)$, and $\gamma^{i_s} - \gamma^i = (1-\gamma^i) - (1-\gamma^{i_s}) \leq (1-\gamma^{L-k})/(j+1)$: for $s < j$ because $1 - \gamma^i < \frac{s+1}{j+1}(1-\gamma^{L-k})$, and for $s = j$ because $i \leq L - k$. Hence the error is at most $F_L := \gamma^k(1-\gamma^{L-k})/((j+1)(1-\gamma)) \leq \gamma^k(1-\gamma^L)/((j+1)(1-\gamma))$, and the three cases of the proof of Theorem 4.6 give $F_L \leq e^2 \Psi_{m,L}$: for $\gamma < \frac{1}{2}$, $F_L \leq \gamma^m/(1-\gamma) \leq 2\gamma^m \leq 2\Psi_{m,L}$; for $\gamma \geq \frac{1}{2}$ and $m \leq J$, $F_L < \frac{1-\gamma^L}{m(1-\gamma)} = \gamma^{-m} \cdot \gamma^m \frac{1-\gamma^L}{m(1-\gamma)} \leq \gamma^{-J} \Psi_{m,L} \leq e^2 \Psi_{m,L}$; and for $m \geq J+1$, $F_L \leq \gamma^{m-J}/((J+1)(1-\gamma)) < \gamma^{-J}\gamma^m \leq e^2 \Psi_{m,L}$.

(c) Let $L \geq m+1$ and take Instance I with $n \geq m+2$ states. The single-step increments $x_t = \widehat{A}^t r = \gamma^t \sqrt{n}\, e_{-t}$, $t = 0, \dots, m$, are orthogonal with $\|x_t\|_\mu = \gamma^t$, and each is a block increment with $b = t+1 \leq L$. Let a context have error ε, let $U = \mathrm{col}(D)$ with $\dim U \leq m$, let Π be the orthogonal projection onto U, and let $\hat{x}_t$ be the normalized vectors. Then $\sum_{t \leq m} \|\Pi \hat{x}_t\|^2 \leq \mathrm{tr}\, \Pi \leq m$, so

$$\sum_{t \leq m} \mathrm{dist}(x_t, U)^2 = \sum_{t \leq m} \gamma^{2t}\big(1 - \|\Pi \hat{x}_t\|^2\big) \;\geq\; \gamma^{2m} \sum_{t \leq m} \big(1 - \|\Pi \hat{x}_t\|^2\big) \;\geq\; \gamma^{2m},$$

hence $\max_t \mathrm{dist}(x_t, U) \geq \gamma^m/\sqrt{m+1}$, and Lemma D.5 gives $2\varepsilon \geq \gamma^m/\sqrt{m+1}$. For the second claim, the proof of the lower bound of Theorem 4.6 uses only block increments $\widehat{v}_b - \widehat{v}_a$ with $b \leq L_0$: for Instance I, $L_0 = n = \lceil m+1+1/(1-\gamma^2) \rceil \leq m+2+1/(1-\gamma)$; for Instance II, $L_0 = (2T-1)w$ with $T = \lceil m + 1 + 1/(1-\gamma^{4w}) \rceil$, $w = \lfloor q/4 \rfloor$ and $q = 1/(m(1-\gamma)) > 16$, which forces $\gamma > \frac{1}{2}$. Since $x := 4w(1-\gamma) \leq q(1-\gamma) = 1/m \leq 1$ and $\gamma^{4w} \leq e^{-x}$, we have $1 - \gamma^{4w} \geq 1 - e^{-x} \geq x/2$, so $T \leq m + 2 + 1/(2w(1-\gamma))$ and

$$(2T-1)w \;\leq\; (2m+3)w + \frac{1}{1-\gamma} \;\leq\; \frac{2m+3}{4m(1-\gamma)} + \frac{1}{1-\gamma} \;\leq\; \frac{9}{4(1-\gamma)}.$$

Both depths are at most $m + 4/(1-\gamma)$, so for $L \geq m + 4/(1-\gamma)$ every context of depth L incurs error at least $\Psi_m/64$ on one of the two instances, exactly as in that proof.

(d) Let $\gamma \geq \frac{1}{2}$ and $8m \leq L \leq 1/(1-\gamma)$. Take Instance II with $w = \lfloor L/(4m) \rfloor \geq 2$ and $T = 2m$; its even blocks $u_t = \tau_{2t}$, $t < T$, have $b = (2t+1)w \leq (4m-1)w < L$, are orthogonal, and satisfy $\|u_t\|_\mu = \gamma^{2tw}\|\tau_0\|_\mu \geq a := \gamma^{4mw} S_w/\sqrt{2}$. For a context with error ε and $U = \mathrm{col}(D)$, the argument of part (c) gives $\sum_{t<T} \mathrm{dist}(u_t, U)^2 \geq a^2(T-m) = a^2 m$, so $\max_t \mathrm{dist}(u_t, U) \geq a/\sqrt{2}$ and, by Lemma D.5, $\varepsilon \geq a/(2\sqrt{2}) \geq \gamma^{4mw} S_w/4$. Now $S_w \geq w\gamma^w$, $(4m+1)w \leq \frac{4m+1}{4m} L \leq \frac{5}{4(1-\gamma)}$, and $-\ln \gamma \leq 2(1-\gamma)$ for $\gamma \geq \frac{1}{2}$, so $\gamma^{(4m+1)w} \geq e^{-5/2}$; with $w \geq L/(4m) - 1 \geq L/(8m)$ this gives $\varepsilon \geq e^{-5/2} L/(32m) \geq L/(400m)$. The upper bound is part (b) with $\Psi_{m,L} \leq \gamma^m \max\{1, L/m\} \leq L/m$, because $(1-\gamma^L)/(1-\gamma) \leq L$ and $L \geq m$. □

D.6 Proof of Proposition D.1

Terminal correction (Krylov truncation). For $m < k_L$ the budget-m encoder gives an upper-Hessenberg H_m with unit subdiagonal and $g_m = e_1$. Replacing H_m by $T_m = H_m - (H_m^m g_m) e_m^\top$

preserves $T_m^j g_m = H_m^j g_m$ for $j < m$ and gives $T_m^m g_m = 0$, since $e_m^\top H_m^j g_m = 0$ for $j < m-1$ and $e_m^\top H_m^{m-1} g_m = 1$. Seven further heads compute T_m by m reduced multiplications (Appendix A.5), and the output is exactly $\widehat{v}_{\min\{\ell,m\}}$, i.e., TD stopped after m steps.

Proof of (a). By Lemma D.4, $\|\widehat{v}_\ell - \widehat{v}_m\|_\mu \le \sum_{i\ge m} \gamma^i \|\widehat{r}\|_\mu = \gamma^m \|\widehat{r}\|_\mu/(1-\gamma)$. For the matching lower bound, take the ring with $n = m+2$, $L \ge m+1$, and $r_\varepsilon = \mathbf{1} + \varepsilon\sqrt{n}\, e_0$. The Krylov vectors $\gamma^j(\mathbf{1} + \varepsilon\sqrt{n}\, e_{-j})$ form, up to a column permutation, the matrix $\mathbf{1}\mathbf{1}^\top + \varepsilon\sqrt{n} I$, which is nonsingular; hence $k_L > m$. As $\ell \to \infty$ the error tends to $\|\widehat{A}^m v_\infty\|_\mu$ with $v_\infty = \frac{\mathbf{1}}{1-\gamma} + \varepsilon\sqrt{n}(I - \widehat{A})^{-1} e_0$, which is at least $\gamma^m(1-\varepsilon)/(1-\gamma)$, while $\|r_\varepsilon\|_\mu \le 1 + \varepsilon$. Letting $\varepsilon \to 0$ gives the supremum $\gamma^m/(1-\gamma)$, and $\frac{\gamma^m/(1-\gamma)}{\Psi_m} = \min\{m, \frac{1}{1-\gamma}\}$. □

Proof of (b). Take the ring with $n = m + p$ states, $p = \lceil 1/(1-\gamma) \rceil$, $r = \sqrt{n}\, e_0$, and $L \ge n$. Then $\varphi_t = P^t r$ $(t < n)$ are orthonormal in $\|\cdot\|_\mu$, and all visit counts are equal, so Euclidean and visit-weighted Arnoldi coincide. The encoder gives $b_j \propto \varphi_{j-1}$ for $j \le m$, and $\widehat{A} b_m \propto \varphi_m \perp \mathrm{span}(b_1, \dots, b_m)$, so the last column of H vanishes and H is nilpotent. Galerkin execution therefore outputs $\sum_{t<\min\{\ell,m\}} \gamma^t \varphi_t$, and at $\ell = n$ its error is $\|\sum_{t=m}^{n-1} \gamma^t \varphi_t\|_\mu = \gamma^m \sqrt{(1-\gamma^{2p})/(1-\gamma^2)}$. Since $\gamma^{2p} \le e^{-2}$ and $1 - \gamma^2 \le 2(1-\gamma)$, this is at least $\sqrt{(1-e^{-2})/2}\, \gamma^m/\sqrt{1-\gamma} \ge 0.657\, \gamma^m/\sqrt{1-\gamma}$. For $m \ge 1/(1-\gamma)$ we have $\Psi_m = \gamma^m$, which gives the second form. The instance has $k_L = \min\{n, L\} > m$. □

D.7 Proof of Theorem D.2

Lemma D.10 (Normalized Galerkin form)**.** *Suppose the budget-k w-Arnoldi encoder does not break down. Let Q be its w-orthonormalized basis, $Q^* = Q^\top D_w$, and $\widehat{H} = Q^* \widehat{A} Q$. Then the executed trajectory is $\widetilde{v}_\ell^{\mathrm{G}} = \sum_{i<\ell} Q \widehat{H}^i Q^* \widehat{r}$, the matrix satisfies $\|\widehat{H}\|_2 \le \|\widehat{A}\|_w \le \gamma$, and $Q\widehat{H}^i Q^* \widehat{r} = \widehat{A}^i \widehat{r}$ for $i < k$. Diagonal rescalings of the basis, including ℓ_1 balancing, leave the trajectory unchanged.*

Proof. With $D_g = \mathrm{diag}(\|b_i\|_w)$ and $Q = B D_g^{-1}$, we have $\Omega = D_w B D_g^{-2}$, $H = D_g^{-1} \widehat{H} D_g$, and $g = D_g^{-1} Q^* \widehat{r}$, so $BH^i g = Q \widehat{H}^i Q^* \widehat{r}$. Q is an isometry from $(\mathbb{R}^k, \ell_2)$ into $(\mathbb{R}^n, \|\cdot\|_w)$, which gives the norm bound. Moment matching for $i < k$ follows as in Theorem 4.3, with the w-orthogonal projection QQ^*. □

Proof of Theorem D.2. If breakdown occurs within k rounds, $\mathcal{K}_L$ is invariant, all residuals vanish, and the output is exact. Otherwise let $e_\ell = \widehat{v}_\ell - \widetilde{v}_\ell^{\mathrm{G}}$. By Lemma D.10, $e_\ell = 0$ for $\ell \le k$, and for $k \le a \le b$,

$$\|e_b - e_a\|_w \le \sum_{i=a}^{b-1} \big(\|\widehat{A}^i \widehat{r}\|_w + \|Q \widehat{H}^i Q^* \widehat{r}\|_w \big) \le 2 \sum_{i=a}^{b-1} \gamma^i \|\widehat{r}\|_w.$$

The anchored output error is $\|e_\ell - e_{\sigma(\ell)}\|_w$, where $\sigma(\ell) \in \{k\} \cup \{k + i_s\}$ and $e_k = 0$. The argument of Lemma D.7 bounds it by $2\gamma^k \|\widehat{r}\|_w/((j+1)(1-\gamma))$, and trivially by $2 \sup_\ell \|e_\ell\|_w = 2\mathcal{E}_k^{\mathrm{G}} \|\widehat{r}\|_w$. The increments lie in $\mathrm{col}(B)$ plus the span of j held residual increments, so the output is in $\mathfrak{C}_m$.

It remains to show $F := 2\gamma^k/((j+1)(1-\gamma)) \le 2e^2 \Psi_m$. If $\gamma < \frac{1}{2}$, then $j = 0$ and $F \le 4\gamma^m \le 4\Psi_m$. Let $\gamma \ge \frac{1}{2}$, $J = \lfloor 1/(1-\gamma) \rfloor \ge 2$, and $x = m(1-\gamma)$. If $m \le 3$, then $j = 0$ and $F/\Psi_m = 2\min\{m, 1/(1-\gamma)\} \le 6$. If $j = J < \lfloor m/4 \rfloor$, then $J + 1 > 1/(1-\gamma)$ gives $F < 2\gamma^{-J}\gamma^m \le 2e^2 \Psi_m$. Otherwise $m \ge 4$ and $j = \lfloor m/4 \rfloor \le J$. Then $j + 1 \ge (m+1)/4$ gives $F < 8\gamma^{-j}\gamma^m/x$, hence $F/\Psi_m < 8\gamma^{-j} \min\{1, 1/x\}$. Here $\gamma^{-j} \le \min\{\gamma^{-m/4}, \gamma^{-J}\} \le \min\{e^{x/2}, e^2\}$. The function $e^{x/2}/x$ is convex on $[1, 4]$, so its maximum there is at an endpoint. The bound is therefore at most $8e^{1/2}$ for $x \le 1$, at most $\max\{8e^{1/2}, 2e^2\} = 2e^2$ for $1 < x \le 4$, and at most $8e^2/x < 2e^2$ for $x > 4$. Finally, $k = m - j \ge m - \lfloor m/4 \rfloor = \lceil 3m/4 \rceil$, and each residual token stores n values and one time stamp. □

D.8 Proof of Proposition D.3

Proof of (a). If $N_{ij} = N_{ji}$, then $\sum_i N_{ij} = N_j$, so $w = (N_i)$ is stationary; moreover $w_i \widehat{P}_{ij} = N_{ij}$ is symmetric, so $\widehat{A}$ is w-self-adjoint with spectrum in $[-\gamma, \gamma]$. $\widehat{H}$ is then symmetric with spectrum in $[-\gamma, \gamma]$. If $m \geq k_L$ the Galerkin output is exact. Otherwise, for every polynomial q of degree less than m, $Qq(\widehat{H})Q^*\widehat{r} = q(\widehat{A})\widehat{r}$. With $p_\ell(x) = \sum_{j<\ell} x^j$,

$$\widehat{v}_\ell - \widetilde{v}_\ell^{\mathrm{G}} = (p_\ell - q)(\widehat{A})\widehat{r} - Q(p_\ell - q)(\widehat{H})Q^*\widehat{r}, \qquad \|\widehat{v}_\ell - \widetilde{v}_\ell^{\mathrm{G}}\|_w \leq 2 \sup_{|x|\leq\gamma} |p_\ell(x) - q(x)|\, \|\widehat{r}\|_w.$$

Write $x = \gamma t$ with $t \in [-1, 1]$ and expand in Chebyshev polynomials $T_k(t)$. The Poisson-kernel identity gives $\frac{1}{1-\gamma\cos\theta} = \frac{1}{\sqrt{1-\gamma^2}}(1 + 2\sum_{k\geq 1} \beta^k \cos k\theta)$, so $f(x) = 1/(1-x)$ has coefficients $c_k(f) = 2\beta^k/\sqrt{1-\gamma^2}$ for $k \geq 1$. Every monomial t^j has nonnegative Chebyshev coefficients. Hence $0 \leq c_k(p_\ell) \leq c_k(f)$, since $c_k(p_\ell)$ is nondecreasing in ℓ and converges to $c_k(f)$ by uniform convergence on $[-1, 1]$. Truncating p_ℓ after degree $m-1$ gives $\sup|p_\ell - q_\ell| \leq \sum_{k\geq m} c_k(f) = 2\beta^m/((1-\beta)\sqrt{1-\gamma^2})$ uniformly in ℓ, which proves the first bound (Trefethen, 2013). For the second, $s = \sqrt{1-\gamma^2}$ and $1 - \beta = (1 + s - \gamma)/(1 + s)$, and $(1-\beta)s \geq s^2/2$ is equivalent to $(1-\gamma)^2 + s \geq 0$. The anchored bound follows from Theorem D.2 with $k \geq \lceil 3m/4 \rceil$. □

Proof of (b). Every context in $\mathfrak{C}_m$ has $\widetilde{v}_\ell \in U = \mathrm{col}(D)$, since the initial output is zero. For any U of dimension at most m, Eckart–Young gives $\max_{\ell\leq L} \mathrm{dist}_w(\widehat{v}_\ell, U)^2 \geq \frac{1}{L}\sum_{\ell\leq L} \mathrm{dist}_w(\widehat{v}_\ell, U)^2 = \frac{1}{L}\|(I - \Pi_U)V_L\|_F^2 \geq \sigma_{m+1}^2/L$. Projecting onto the top m left singular vectors gives residual at most $\|(I-\Pi)V_L\|_2 = \sigma_{m+1}$ in every column. For stability, Weyl's inequality gives $|\sigma_i(V_L) - \sigma_i(V_L')| \leq \|V_L - V_L'\|_F \leq \sqrt{L}\sup_\ell \|\widehat{v}_\ell - \widehat{v}_\ell'\|_w$. The recursion $e_{\ell+1} = (\widehat{r} - \widehat{r}') + \widehat{A}e_\ell + (\widehat{A} - \widehat{A}')\widehat{v}_\ell'$ with $\|\widehat{v}_\ell'\|_w \leq \|\widehat{r}'\|_w/(1-\gamma)$ then gives $\sup_\ell \|e_\ell\|_w \leq \delta_r/(1-\gamma) + \delta_A\|\widehat{r}'\|_w/(1-\gamma)^2$ when both models are w-contractive. □

D.9 A Certified Adaptive Rank

Proposition D.11 (Threshold certificate)**.** *Add the declared gate* $\mathrm{thr}_c(x) = \mathbf{1}\{x > c\}$*. In round* j *of* w*-Arnoldi, multiply* b_{j+1} *by* $\mathrm{thr}_{\tau^2}(d_{j+1}/d_j)$*, computed as* $d_{j+1}\,\mathrm{recip}_0(d_j)$*. Let* $j^\star$ *be the number of retained directions. If the gate fired, i.e.,* $\rho := \|b_{j^\star+1}\|_w/\|b_{j^\star}\|_w \leq \tau$*, then under* w*-contraction*

$$\sup_{\ell\geq 0} \|\widehat{v}_\ell - \widetilde{v}_\ell^{\mathrm{G}}\|_w \leq \frac{\tau}{(1-\gamma)^2}\|\widehat{r}\|_w.$$

Otherwise the output is exact through depth L*. Always* $j^\star \leq k_L$*, and* $\tau = 0$ *recovers the exact encoder.*

Proof. With $q_{j+1} = b_{j+1}/\|b_{j+1}\|_w$, the Arnoldi relation reads $\widehat{A}Q = Q\widehat{H} + \rho\, q_{j^\star+1}e_{j^\star}^\top$, where $b_{j^\star+1}$ is the residual before gating. Let $\epsilon_i = \widehat{A}^i\widehat{r} - Q\widehat{H}^iQ^*\widehat{r}$, so that $\epsilon_0 = 0$ and $\epsilon_{i+1} = \widehat{A}\epsilon_i + \rho\, q_{j^\star+1}e_{j^\star}^\top\widehat{H}^iQ^*\widehat{r}$. Hence $\|\epsilon_{i+1}\|_w \leq \gamma\|\epsilon_i\|_w + \rho\gamma^i\|\widehat{r}\|_w$, and induction gives $\|\epsilon_i\|_w \leq i\rho\gamma^{i-1}\|\widehat{r}\|_w$. Summing over $i \geq 1$ gives $\rho/(1-\gamma)^2$. □

The certificate controls the error, not the size of $j^\star$, because Arnoldi residuals can stagnate. Proposition D.3(b) gives the method-independent lower bound against which $j^\star$ can be compared.

D.10 Numerically Optimized Time-Invariant Recurrences

For $\widetilde{v}_\ell = D\sum_{j<\ell} K^j s_0$ with context dimension m, the increment sequences $j \mapsto K^j s_0$ span a space of exponential sums whose modes are the eigenvalues of K. We parameterize $m/2$ damped complex modes $\rho_i e^{\pm \mathrm{i}\theta_i}$. For fixed modes, the minimax problem over D is convex, and we solve it with Lawson iterations. The modes are optimized by Nelder–Mead from 12 starts. This is a numerical search, so it certifies achievability of the reported values but not global optimality. On Instance I with $\gamma = 0.9$:

m	2	4	6	8
Best time-invariant found	0.535	0.341	0.251	0.194
Sample-and-hold	1.570	1.345	1.000	1.000
Certified lower bound (any method)	0.307	0.231	0.180	0.143

Time-invariant recurrences with instance-optimized modes are therefore within $1.8\times$ of optimal on this instance, and better than sample-and-hold, whose first sample occurs after depth one. The suboptimality in Proposition D.1 is thus a property of moment matching, not of looping without a clock. What remains open is whether some encoder computable in the forward pass produces time-invariant contexts with error $O(\Psi_m)$ on *every* system. The depth-counter constructions give a positive answer with a simple, fixed encoder.

E Execution Depth

The time-invariant executor (5) uses L layers and emits every depth along the way. For a single-depth query we ask how few layers suffice, and how few are necessary. A *polynomial executor* of depth t composes t maps $F(Z) = W_0 Z + c_0 + \sum_h P_h Z[\mathcal{R}_h \odot (Z^\top Q_h Z)]$ with fixed parameters that may depend on n, L, m but not on γ or the task, followed by a fixed linear readout. The class covers every executor of Appendix A.7 without recip_0.

Theorem E.1 (Depth bounds)**.** ***(a)*** *Suppose every context entry is a Laurent polynomial in γ with exponents in $[-e, e]$, and a depth-t polynomial executor outputs $\widehat{v}_L(q)$ for all γ in an open interval on a task with $e_q^\top \widehat{P}^{L-1}\widehat{r} \neq 0$. Then $3^t e \geq L-1$. The unnormalized Krylov context (4) has $e = k_L$, whereas the ℓ_1-balanced Krylov context used in Section 5 and the state-level context (9) have $e = 1$ (Lemma E.3); hence $t \geq \log_3 \frac{L-1}{k_L}$ in general and $t \geq \log_3(L-1)$ for the balanced and state-level contexts.* ***(b)*** *Retaining the reduced columns of H, a fixed executor with two layer types (repeated squaring of H and binary accumulation of $\sum_{j<L} H^j g$) outputs $\widehat{v}_L(q)$ exactly in $\lfloor \log_2 L \rfloor + 2$ layers. Its retained memory is $(n+h+2)k_L + O(n) \leq 2(nk_L + k_L) + O(n)$.*

For single-depth queries on the balanced context, the optimal depth therefore lies between $\log_3(L-1)$ and $\log_2 L + 2$, which agree up to the factor $\log_2 3 \approx 1.58$ and an additive constant (a further $\log_3 k_L$ for the unnormalized context); only the base of the logarithm is open. The lower bound is for polynomial executors, which excludes recip_0, so it constrains the doubling and time-invariant executors but not the depth-counter one, whose unrolled recip_0-free variant has depth L. Compression reduces memory (Theorem 4.3) but not depth, since the uncompressed state-level context can also be squared in $O(\log L)$ layers at n^2 memory. The looped executor's linear depth buys all L depths in one pass.

E.1 Structure of the Context in γ

Lemma E.2 (Laurent structure in γ)**.** *Fix real P, r with $k = \dim \mathcal{K}_L(P, r) \geq 1$, and write $\tilde{b}_j, \tilde{\omega}_j, \tilde{H}$ for the unbalanced encoder outputs on (P, r). Then on $(\gamma P, r)$ with $\gamma \neq 0$, the outputs are $b_j = \gamma^{j-1}\tilde{b}_j$, $\omega_j = \gamma^{-(j-1)}\tilde{\omega}_j$, $H_{ij} = \gamma^{j-i+1}\tilde{H}_{ij}$, and $g = e_1$. Every entry of B, Y, H, and g is a Laurent polynomial in γ with exponents in $[-k, k]$.*

Proof. Induct on rounds. If $b_i = \gamma^{i-1}\tilde{b}_i$ for $i \leq j$, then $d_i = \gamma^{2(i-1)}\tilde{d}_i$, $\omega_i = \gamma^{-(i-1)}\tilde{\omega}_i$, and $\gamma P b_j = \gamma^j P \tilde{b}_j$. Substituting into (3) gives $b_{j+1} = \gamma^j \tilde{b}_{j+1}$, and the gates do not depend on γ. Then $H_{ij} = \omega_i^\top \gamma P b_j = \gamma^{j-i+1}\tilde{H}_{ij}$, and $g = \Omega^\top b_1 = e_1$ by duality. H is upper Hessenberg, since $\widehat{A} b_j \in \mathrm{span}(b_1, \ldots, b_{j+1})$, so its nonzero exponents lie in $[0, k]$. The three terms of $y_j = b_j + \sum_i \omega_i (H_{ji} - \delta_{ji})$ have exponents $j-1$, $2-j$ (from $\gamma^{-(i-1)}\gamma^{i-j+1}$), and $1-j$. All lie in $[-k, k]$. □

Lemma E.3 (The balanced context is affine in γ)**.** *With positive ℓ_1 balancing, write $\tilde{b}_j, \tilde{\omega}_j, \tilde{H}, \tilde{g}$ for the balanced encoder outputs on (P, r). On $(\gamma P, r)$ with $\gamma > 0$ the outputs are $b_j = \tilde{b}_j$, $\omega_j = \tilde{\omega}_j$, $g = \tilde{g}$, $H = \gamma \tilde{H}$, and $Y = \tilde{B} - \tilde{\Omega} + \gamma \tilde{\Omega} \tilde{H}^\top$, and the gates do not depend on γ. Every entry of B, Y, H, and g is therefore a polynomial of degree at most one in γ, i.e., $e = 1$ in Theorem E.1(a).*

Proof. Induct on rounds. The first candidate is $\widehat{r}$, which does not involve γ. If $b_i = \tilde{b}_i$ and $\omega_i = \tilde{\omega}_i$ for $i \le j$, the next candidate before balancing is $\gamma P b_j - \sum_{i\le j} b_i \omega_i^\top \gamma P b_j = \gamma\big(P\tilde{b}_j - \sum_{i\le j} \tilde{b}_i \tilde{\omega}_i^\top P\tilde{b}_j\big)$, a positive multiple of the candidate on (P, r). Balancing divides by the ℓ_1 norm, which removes the factor γ, so $b_{j+1} = \tilde{b}_{j+1}$; the squared norm, the dual, and the gate of round $j+1$ are functions of b_{j+1} alone. Hence $H_{ij} = \omega_i^\top \gamma P b_j = \gamma \tilde{H}_{ij}$, $g = \Omega^\top \widehat{r} = \tilde{g}$, and $y_j = b_j + \sum_i \omega_i (H_{ji} - \delta_{ji})$ is affine in γ. Numerically, on 30 random covered batches the balanced B and Ω at $\gamma \in \{0.5, 0.8, 0.95\}$ agree to 7.1×10^{-15} and g to 1.8×10^{-15}, H/γ agrees to 1.3×10^{-15}, and Y is affine to 2.8×10^{-14} (`check_balanced_affine.py`). □

E.2 Doubling Executor

Each retained Krylov token j carries b_j, a block $m_j \in \mathbb{R}^h$ initialized to column j of H, a fixed tag $p_j = e_j$, and scalars a_j (initialized to g_j) and c_j (initialized to 0). Let M denote the matrix with columns m_j. Six token-to-token heads and one readout head are used. The squaring head (key tag, query m, value m, output m, $\sigma = +$) adds $\sum_i m_i (p_i^\top m_t) = M^2 e_t$, and a clearing head (key tag, query tag, value m, output m, $\sigma = -$) removes the old column, so together they map $M \mapsto M^2$. The advance head (key m, query tag, value a, output a) adds $(Ma)_t$, so $a \mapsto a + Ma$. The accumulation heads add $(Mc)_t$ to c (key m, query tag, value c), add a_t to c (key tag, query tag, value a, output c), and subtract c_t (key tag, query tag, value c, output c, $\sigma = -$), so together $c \mapsto a + Mc$. With synchronous reads, a layer of type Λ_0 (squaring, clearing, advance) maps (M, a, c) to $(M^2, a+Ma, c)$, and a layer of type Λ_1 (all six heads) maps it to $(M^2, a+Ma, a+Mc)$. Layer $s = 0, \dots, \lfloor \log_2 L \rfloor$ uses Λ_{β_s}, where $L = \sum_s \beta_s 2^s$, and a final readout head (key b, query e_q, value c) outputs $e_q^\top B c$.

E.3 Proofs

Lemma E.4 (Degree growth)**.** *Each output entry of $F(Z) = W_0 Z + c_0 + \sum_h P_h Z[\mathcal{R}_h \odot (Z^\top Q_h Z)]$ is a polynomial of degree at most 3 in the entries of Z. A composition of t such maps followed by a linear readout therefore has degree at most 3^t.*

Proof. $P_h Z$ is linear, $Z^\top Q_h Z$ is quadratic, and the masked product is cubic. Degrees multiply under composition. □

Proof of Theorem E.1(a). By Lemma E.4, a polynomial of degree at most 3^t in Laurent polynomials with exponents in $[-e, e]$ is a Laurent polynomial $O(\gamma)$ with exponents in $[-3^t e, 3^t e]$. The target $V(\gamma) = \sum_{j<L} \gamma^j e_q^\top \widehat{P}^j \widehat{r}$ is a polynomial whose coefficient of γ^{L-1} is nonzero. The polynomial $\gamma^{3^t e}(O - V)$ vanishes on an open interval, so it is identically zero. Hence $O = V$ as Laurent polynomials, and $L - 1 \le 3^t e$. For the unnormalized Krylov context, Lemma E.2 gives $e = k_L$; for the balanced context used in the experiments, Lemma E.3 gives $e = 1$; the query token is constant. The state-level context (9) is affine in γ, so $e = 1$. Tasks satisfying the hypothesis exist, for example the base point of Lemma C.2 with all $\cos(\pi j/n) \neq 0$. □

Proof of Theorem E.1(b). Use the heads of Appendix E.2. Since $p_i^\top p_t = \delta_{it}$ and M is supported on the first k_L coordinates, the head `sq` gives $\sum_i m_i (p_i^\top m_t) = M^2 e_t$, and `adv` gives $\sum_i a_i (m_i^\top p_t) = (Ma)_t$. Let $p_x(H) = \sum_{j<x} H^j$ and $x_s = \sum_{s'<s} \beta_{s'} 2^{s'}$. We claim that at the start of step s, $M = H^{2^s}$, $a = p_{2^s}(H) g$, and $c = p_{x_s}(H) g$. This holds at $s = 0$, and it is preserved because $p_{x+y} = p_y + H^y p_x$ gives $a + Ma = p_{2^{s+1}} g$ and $a + Mc = p_{x_s + 2^s} g$. After step $\lfloor \log_2 L \rfloor$ we have $c = p_L(H) g$, and the readout gives $e_q^\top B p_L(H) g = e_q^\top \sum_{j<L} \widehat{A}^j \widehat{r}$ by moment matching. The retained data are B (nk_L entries), the columns of H (hk_L), a and c ($2k_L$), and the query ($O(n)$). Since $h \le n$, the total is at most $2(nk_L + k_L) + O(n)$. Numerically, the recurrence of Appendix E.2 evaluated as matrix products (`verify_all.py`) has relative error at most 7.0×10^{-15} for $L \le 200$. □

F Learnability of the Shared Executor

Algorithm 1 Pretraining procedure for the effective TD parameters (C, W).

Require: Pretraining task distribution $\mathcal{D}_{\mathrm{pre}}$, step size $\eta > 0$, number of iterations T, fixed $P = P_{\mathrm{TD}}$, fixed encoder, initial $C_0, W_0 \in \mathbb{R}^{n\times n}$

1: **for** $i = 0, \dots, T-1$ **do**
2: For each task τ, run the encoder to obtain $(\widehat{A}_\tau, \widehat{r}_\tau)$ and form $F_\tau(C_i, W_i) = I_n + C_i^\top + \widehat{A}_\tau W_i$.
3: Evaluate the full-batch objective $\mathcal{J}(C_i, W_i)$ in (7) with the calibration input u.
4: Update $C_{i+1} = C_i - \eta \nabla_C \mathcal{J}(C_i, W_i)$ and $W_{i+1} = W_i - \eta \nabla_W \mathcal{J}(C_i, W_i)$.
5: **end for**
6: **return** $(P_{\mathrm{TD}}, Q_{\mathrm{eff}}(C_T, W_T))$ with Q_{eff} as in Section 4.3.

Interface. The encoder is frozen. Its first stage produces a state context with static blocks I_n and $\widehat{A}_\tau^\top$. A calibration call replaces the source scalars by a probe u and resets the query accumulator. By direct multiplication with $Q_{\mathrm{eff}}(C, W)$, the readout increment is $-C^\top u$ and the next source state is $(I + C^\top + \widehat{A}_\tau W)u$. Both targets in (7) are computed from input statistics and the probe; neither a solved value function nor a target parameter enters the loss.

Proof of Theorem 4.9. Let $E_C = C + I$ and $E_W = W - I$. Isotropy of u gives $2\mathcal{J} = \|E_C\|_F^2 + \mathbb{E}_\tau \|E_C^\top + \widehat{A}_\tau E_W\|_F^2$. For equally shaped arrays a, b, the quadratic form $\|a\|_F^2 + \|a+b\|_F^2$ has coefficient matrix $\left[\begin{smallmatrix} 2 & 1 \\ 1 & 1 \end{smallmatrix}\right]$, whose eigenvalues are $c_\pm = (3 \pm \sqrt{5})/2$. Therefore $c_-(\|E_C\|_F^2 + \mathbb{E}\|\widehat{A}_\tau E_W\|_F^2) \le 2\mathcal{J} \le c_+(\|E_C\|_F^2 + \mathbb{E}\|\widehat{A}_\tau E_W\|_F^2)$. Since $\mathbb{E}\|\widehat{A}_\tau E_W\|_F^2 = \mathrm{tr}(E_W^\top G E_W)$ with $\kappa I \preceq G \preceq \beta' I$, the Hessian satisfies $\lambda_- I \preceq \mathcal{H} \preceq \lambda_+ I$. The unique zero is $E_C = E_W = 0$, and a GD step is $e_{t+1} = (I - \eta\mathcal{H})e_t$, whose eigenvalues lie in $[0, 1 - \eta\lambda_-]$ for $\eta \le 1/\lambda_+$. □

Necessity of operator diversity. If G is singular, take z in its nullspace; then $\widehat{A}_\tau z = 0$ almost surely. The pair $C = -I$, $W = I + za^\top$ has zero loss for every a, so the probe objective cannot identify unexcited directions. Conversely, if discounted permutations appear with probability $p > 0$, then $G \succeq p\gamma_0^2 I$ even when all rewards coincide.

Parameter error and trajectory error. For a fixed exact context with $K_\theta = I + B^\top C^\top B + Y^\top W B$ and $v_\ell^\theta = -C^\top B \sum_{j<\ell} K_\theta^j g$, let $\delta_C = \|C + I\|_2$, $\delta_W = \|W - I\|_2$, $\Delta = \|B\|_2^2 \delta_C + \|Y\|_2 \|B\|_2 \delta_W$, and $b = \max\{\|K_\theta\|_2, \|H\|_2\}$. Then $\|K_\theta - H\|_2 \le \Delta$ and $\|v_\ell^\theta - \widehat{v}_\ell\|_2 \le \|B\|_2 \|g\|_2 [\delta_C \sum_{j<\ell} b^j + \Delta \sum_{1\le j<\ell} j b^{j-1}]$, by telescoping $K_\theta^j - H^j = \sum_i K_\theta^{j-1-i}(K_\theta - H)H^i$. Hence gradient descent also converges in output on every fixed finite-depth context.

Why an endpoint target differs. Scaling to $C = -aI$, $W = aI$ $(0 < a \le 1)$ gives $K_a = (1-a)I + aH$ and increments $aBK_a^j g$. When H is stable, $aB(I - K_a)^{-1} g = B(I - H)^{-1} g$ for every a. Different unit-step rules can therefore share a limit, which is why terminal prediction and identification of the TD step are different objectives.

Proof of Proposition 4.7. Write $C = -I + E_C$ and $W = I + E_W$, so that $-C^\top = I - E_C^\top$ and $F_\tau = \widehat{A}_\tau + \delta F_\tau$ with $\delta F_\tau = E_C^\top + \widehat{A}_\tau E_W$. To first order in (E_C, E_W),

$$\delta v_{\tau,1}^\theta = -E_C^\top \widehat{r}_\tau, \qquad \delta v_{\tau,2}^\theta = -E_C^\top(\widehat{r}_\tau + \widehat{A}_\tau \widehat{r}_\tau) + \delta F_\tau \widehat{r}_\tau = -E_C^\top \widehat{A}_\tau \widehat{r}_\tau + \widehat{A}_\tau E_W \widehat{r}_\tau.$$

If the Jacobian applied to (E_C, E_W) vanishes in $L^2(\tau)$, then $\mathbb{E}_\tau \|E_C^\top \widehat{r}_\tau\|^2 = \mathrm{tr}(E_C^\top \mathbb{E}[\widehat{r}_\tau \widehat{r}_\tau^\top] E_C) = 0$ forces $E_C = 0$, and then $\mathbb{E}_\tau \|\widehat{A}_\tau E_W \widehat{r}_\tau\|^2 = \mathrm{vec}(E_W)^\top \mathbb{E}[(\widehat{r}_\tau \widehat{r}_\tau^\top) \otimes (\widehat{A}_\tau^\top \widehat{A}_\tau)] \,\mathrm{vec}(E_W) = 0$ forces $E_W = 0$. Hence the Jacobian is injective, the Gauss–Newton Hessian of $\mathcal{J}_{\mathrm{traj}}$ at the target is positive definite, and the target, at which $\mathcal{J}_{\mathrm{traj}} = 0$, is a strict local minimizer. For the endpoint map, on the curve $(C, W) = (-aI, aI)$ we have $-C^\top = aI$ and $F_\tau = (1-a)I + a\widehat{A}_\tau$, so $v_{\tau,\infty}^\theta = aI\,(I - F_\tau)^{-1} \widehat{r}_\tau = (I - \widehat{A}_\tau)^{-1} \widehat{r}_\tau$ for every a; differentiating at $a = 1$ shows that the Jacobian of $(C, W) \mapsto v_{\tau,\infty}^\theta$ annihilates $(-I, I)$. □

The calibration objective (7) is the same two-depth identification with an isotropic input in place of $\widehat{r}_\tau$: it removes the reward moments from the conditions and makes the objective exactly quadratic, which is what Theorem 4.9 exploits.

G Connections to Classical Statistical and Reduced Models

LSTD and iLSTD. With features x_t, batch TD depends on the data only through $G_N = \sum_t x_t(x_t - \gamma x_{t+1})^\top$ and $b_N = \sum_t x_t R_{t+1}$. LSTD solves the projected fixed point, and recursive variants update these statistics online (Bradtke & Barto, 1996; Boyan, 2002; Geramifard et al., 2006). Our first stage is the tabular, visitation-normalized instance and is not a new principle. Theorem C.1(4) shows that these statistics are optimal up to a factor $(n+1)/n$ among continuous encoders when $k_L = n$.

BEBFs, Krylov spaces, and model reduction. Reward-initialized Krylov, Bellman-error, and model-error features span the same spaces under the conditions of Parr et al. (2008) (see also Parr et al., 2007; Petrik, 2007). The sequence $\widehat{A}^j\widehat{r}$ is the impulse response of a linear system with full-state output. Theorem 4.3(a) is therefore a finite-horizon minimal-realization statement (Ho & Kalman, 1966; Freund, 2003; Antoulas, 2005). Classical Krylov model reduction matches moments and controls H_2 or H_∞ errors; our error is the supremum over depths of partial sums. Under this error, moment matching is not minimax (Proposition D.1).

Lumpability. If $PF = F\overline{P}$ and $r = F\overline{r}$ for a partition matrix F, then $\sum_{j<\ell}(\gamma P)^j r = F\sum_{j<\ell}(\gamma\overline{P})^j\overline{r}$ (Kemeny & Snell, 1976; Givan et al., 2003). A quotient restricts decoding directions to block indicators. Krylov directions are reward specific and exist without a compatible partition; the block-cycle experiments are lumpable, and the sampled experiments are not.

Widths. Lemma D.6 is a Kolmogorov-width computation for orthogonal families (Pinkus, 1985). The new ingredient is the choice of block increments at the scale $1/(m(1-\gamma))$, which is what removes the γ-dependence of the constant.

Constructions for other in-context RL problems. Table 1 lists only constructions that execute TD policy evaluation in the forward pass, because our retained object, the Krylov context of the TD trajectory, is a compression of that computation. Three further constructions retain other objects and are not comparable in retained memory: Lin et al. (2024) implement LinUCB and Thompson sampling for bandits and UCB-VI for tabular MDPs from T rounds of interaction with $|\mathcal{A}|$ actions, retaining $\Theta(Td|\mathcal{A}|)$; Liang & Lai (2026) implement semi-gradient SARSA and actor–critic on N transitions with d-dimensional state–action features, retaining $\Theta(Nd)$; and Chen et al. (2025) implement the MASTER strategy for non-stationary bandits over T rounds, retaining $\Theta(Td\log T)$. Measured with the protocol of Table 1 ($T = 2000$, $d = 24$, $|\mathcal{A}| = 4$, padding to 2048 tokens for the last), their encoded buffers are at least 2000 KiB, 578.41 KiB, and 2824 KiB, respectively; the first excludes unspecified auxiliary padding. Whether the control computations they execute admit a Krylov-type compression is open, since the relevant subspace changes with the policy.

H Experimental Details and Additional Results

All experiments start from raw records (S_t, R_{t+1}, S_{t+1}). The implementation stores tokens by role, with one array per named field, and evaluates every task-dependent operation as a routed selector head (2) with synchronous reads, plus recip_0, release, the ReLU absolute value used by ℓ_1 balancing, and the clock gate. It does not call QR, SVD, a linear solver, or a rank routine to build any compiled context. Such routines appear only in tests, in the method-free references of Figure 3, and in the learned time-invariant read-out of Appendix H.9, which rescales its recurrence by a numerically computed spectral norm. Computation is in float64 on a CPU; the NumPy scripts were run with Python 3.9.6, NumPy 2.0.2, and SciPy 1.13.1, and the PyTorch scripts of Appendices H.6 and H.9 with Python 3.11.16, NumPy 1.26.4, and PyTorch 2.2.2. Other recent versions reproduce the reported values up to floating-point rounding.

H.1 Identity and Bound Checks

Table 4: **Implementation identities and bound checks.** "Max" is the maximum over the stated random instances. Ratios below one mean that the bound holds with room to spare.

Check (instances)	Max
Stage-1 heads vs. direct counting of $(\widehat{A}, \widehat{r})$ (60 random batches)	2.2×10^{-16}
Exact Krylov context + TD executor vs. direct TD, balanced two-pass (60)	3.8×10^{-15} rel.
same, literal unbalanced one-pass encoder (60)	8.0×10^{-13} rel.
Visit-weighted Galerkin execution vs. closed form; dual biorthogonality (60)	2.4×10^{-15}; 4.4×10^{-16}
Sample-and-hold encoder + depth-counter executor vs. $\widehat{v}_{\sigma(\ell)}$ (60)	8.9×10^{-16}
Krylov-anchored encoder + executor vs. $\widetilde{v}_\ell^{\mathrm{G}} + e_{\sigma(\ell)}$ (40)	2.8×10^{-15}
Doubling executor (matrix recurrence of Appendix E.2) vs. $\widehat{v}_L(q)$, $L \le 200$ (50)	7.0×10^{-15} rel.
Theorem C.1 family: Jacobian rank of $\theta \mapsto$ trajectory for $(n,k,L) = (8,3,6), (10,5,6), (12,2,9), (7,4,4)$	$27, 55, 26, 28$ (full)
Sample-and-hold error $/\,\gamma^k((j+1)(1-\gamma))^{-1}$; $/\,e^2\Psi_m$ in $\|\cdot\|_\mu$ and $\|\cdot\|_\infty$ (300 + 120 systems)	1.00; 0.34, 0.30
Anchored error $/\min\{2e^2\Psi_m, 2\gamma^k((j+1)(1-\gamma))^{-1}\}$; $/(2\mathcal{E}_k^{\mathrm{G}})$ (120 count-symmetric)	0.17; 0.50
Lanczos error $/\,4\beta^m((1-\beta)\sqrt{1-\gamma^2})^{-1}$, $\gamma \le 0.999$ (100 count-symmetric)	0.082
Threshold certificate error $/\,\tau(1-\gamma)^{-2}$ (120)	0.30
Truncation on the constant-plus-spike ring vs. $\gamma^m/(1-\gamma)$ at $(\gamma, m) = (0.9, 10), (0.99, 10)$	3.4868, 90.438 (exact)

All entries are float64 comparisons. The first six rows run the attention-level implementation (the Stage-1 heads, the Krylov encoder, the executors, and the sample-and-hold and anchored encoders) on random covered batches against direct numerical references (counting, the TD recurrence, and closed forms). The remaining rows check the stated inequalities numerically: the doubling executor, the terminal-corrected truncation values, and the threshold certificate are evaluated by direct matrix recurrences of the corresponding formulas rather than through selector heads, and the Jacobian row is a finite-difference rank computation on the family of Appendix C, with the perturbed model kept inside the open set of positive kernels. Rank decisions use the finite-precision tolerance of Appendix A.10, so "exact" throughout this appendix means agreement to float64 rounding, not exact arithmetic. The scripts are `check_impl.py`, `check_anchored.py`, `check_softmax.py`, `verify_all.py`, `verify_lower_bound.py`, and `check_bounds.py`.

H.2 Context–Accuracy Tradeoff (Figure 1)

Panels (a) and (b) use 100 tasks each with $n = 24$, $\gamma = 0.9$, $L = 30$, and $N = 960$ covered records, and report the mean and interquartile range of the normalized all-depth error $\max_{\ell \le L} \|\widehat{v}_\ell - \widetilde{v}_\ell\|_w / \|\widehat{r}\|_w$ with visit weights w, for the Krylov projection, Krylov truncation, sample-and-hold, and anchored contexts built from the raw records by the attention-level implementation. *Cyclic MRPs:* 12 blocks of two states; every record from block b moves to a uniformly random state of block $b+1$ modulo 12; the reward is a random multiple in $\{0.5, \dots, 1.5\}$ of the indicator of block 0; state labels are permuted (seed 2027). Row sums into the next block are exactly one and rewards are block-constant, so $k_L = 12$ algebraically, and the balanced encoder returns $k_L = 12$ on all 100 tasks. *Heterogeneous reversible MRPs:* symmetric random weights with density 0.4 plus 0.05 on the diagonal, normalized by rows, with deterministic per-state rewards scaled to $\|r\|_\infty = 1$, observed along one trajectory of 960 steps (seed 2028; one extra visit is appended to any unvisited state). All 100 tasks have $k_L = 24$. Panel (c) is the $\gamma = 0.99$ panel of Figure 2, whose instances and certificate are described in the next subsection. The mean Krylov projection error on the cyclic tasks is 0.86 at $m = 11$ and 6×10^{-16} for $m \ge 12$; on the reversible tasks it is 1.1×10^{-1}, 2.0×10^{-4}, 2.2×10^{-7}, and 1.9×10^{-14} at $m = 4, 8, 12, 20$, while truncation gives 1.2, 0.73, 0.45, and 0.15 and sample-and-hold 1.1, 1.0, 0.12, and 0.053. The scripts are `exp_threshold.py` and `make_figure_context.py`.

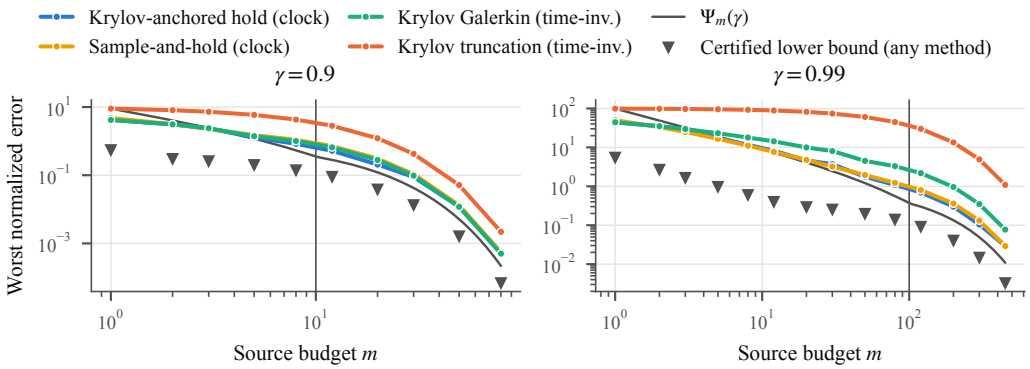


Figure 2: γ**-uniform context–accuracy law.** Worst normalized all-depth error over structured ring instances, built from raw batches through attention heads, at $\gamma = 0.9$ and $\gamma = 0.99$. Triangles give a certified lower bound for any m-dimensional context; the vertical line marks $m = 1/(1-\gamma)$. Sample-and-hold and the anchored hold follow Ψ_m, whereas Krylov truncation plateaus until the effective horizon. The right panel is reproduced in Figure 1(c).

H.3 The γ-Uniform Law (Figure 2)

For $\gamma \in \{0.9, 0.99\}$, the errors of sample-and-hold and the anchored hold stay within $[0.45, 2.7] \times \Psi_m$ and within $21\times$ of the certified bound. Below the horizon at $\gamma = 0.99$ ($2 \le m \le 50$), their log–log slopes are -0.89 and -0.93, against the predicted -1. Krylov truncation has slope -0.14 there and is up to $37\times$ worse, close to the predicted $1/(e(1-\gamma)) \approx 36.8$. Krylov projection is up to $2.7\times$ worse, growing with $1/\sqrt{1-\gamma}$. For each (γ, m), the instance bank contains the following, all realized as covered raw batches with one transition per state (so $\widehat{P}$ is the ring and μ is uniform): (i) Instance I with $n = \lceil m + 1/(1-\gamma^2) \rceil + 2 + \lfloor 2/(1-\gamma) \rfloor$; (ii) the constant-plus-spike ring of Proposition D.1(a) with $n = m+2$ and $\varepsilon = 10^{-6}$; (iii) rewarded-arc rings with $B = 2m$ even blocks, $n = (2B+1)w+1$, and $w = \mathrm{round}(1/(c\,m(1-\gamma))) \ge 2$ for $c \in \{2, 4, 8\}$. Errors use depths up to $n + 15/(1-\gamma)$. The certified lower bound is the largest, over the bank, of half the exact water-filling width of the orthogonal (block) increments, i.e., the smallest ε with $\sum_t (1 - \varepsilon^2/\|x_t\|^2)_+ \le m$; this is valid for every method in $\mathfrak{C}_m$ by Lemma D.5. Krylov truncation is evaluated through its exact identity with TD stopped after m steps. Galerkin uses the balanced two-pass Arnoldi encoder and the TD executor; sample-and-hold and the anchored hold use their encoders and the depth-counter executor.

H.4 Sampled Noisy Tasks (Figure 3)

Figure 3 uses single trajectories with Gaussian reward noise ($\sigma = 0.5$), $\gamma = 0.99$, depths up to 300, and five seeds. On all three tasks the exact rank is generically $k_L = n$, so exact compression gives nothing, but the ε-rank is small. On the two spectrally benign tasks the anchored hold stays within twice the projection error at context dimension $\lceil 3m/4 \rceil$, as Theorem D.2 predicts; on the goal task, where projection stalls, it follows sample-and-hold; neither pure method does both. It falls below the statistical error of $\widehat{v}$ (horizontal line) with $m = 12$, 30, and 5, against $n = 144$, 126, and 200. The shaded band, from Proposition D.3(b), bounds what any m-dimensional context can achieve on each task.

Reversible grid walk: a 12×12 grid with uniform moves, where blocked moves stay in place (symmetric P). The mean reward is the difference of two Gaussian bumps centered at relative positions $(0.25, 0.3)$ and $(0.7, 0.75)$, with widths $0.12a$ and $0.15a$ and weights 1 and -0.6. *Goal-directed gridworld:* a 12×12 grid with two interior walls, each with one gap ($n = 126$ free cells). The policy is ϵ-greedy ($\epsilon = 0.2$) toward the goal along BFS shortest paths; the goal resets to the start cell. The reward is -0.02 per step and $+1$ at the goal. The chain is nonreversible. *Dense drifting kernel:* $n = 200$ points on a circle; the transition from x follows a Gaussian kernel of width 0.08

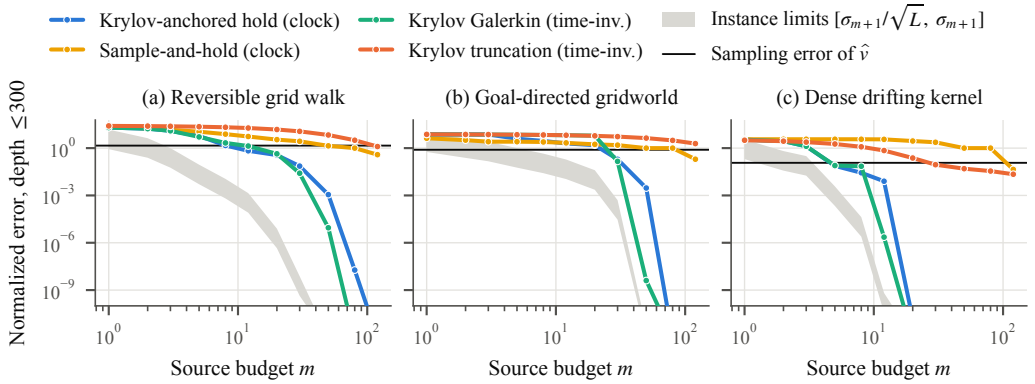


Figure 3: **Sampled noisy tasks.** Mean normalized error over five raw trajectories versus the context dimension. The horizontal line is the sampling error of the empirical trajectory $\widehat{v}$ relative to the population TD trajectory; the band is the numerical-rank window of Proposition D.3(b).

centered at $x + 0.05$ (dense and nonreversible); the mean reward is $\sin 2\pi x + 0.5 \cos 6\pi x$. Each run samples a single trajectory of length N (10^5, 10^5, 2×10^5) with i.i.d. Gaussian reward noise of standard deviation 0.5 (seeds 1000–1004); full coverage held in every run. With $\gamma = 0.99$ and $L = 300$, the error is $\max_{\ell \leq 300} \|\widehat{v}_\ell - \widetilde{v}_\ell\|_w / \|\widehat{r}\|_w$ with visit weights w. The sampling error is the same functional of $\widehat{v}_\ell - v_\ell^{\mathrm{pop}}$, where v^{pop} is the TD trajectory of the population model. The band plots $[\sigma_{m+1}/\sqrt{L}, \sigma_{m+1}]$ of the normalized V_L. In these runs the exact rank is $k_L = n$, while the number of singular values of the normalized V_L above 10^{-2} is only 9, 25, and 5.

H.5 Memory Accounting (Table 5)

Table 5: **Retained memory (KiB, float32) on exact and sampled tasks.** Exact rows: dense-block cycles, $\gamma = 0.9$, $L = 100$, $N = 40n$; LB is $nk_L + k_L$ (Theorem 4.3(c)). Approximate rows: $\gamma = 0.99$, $L = 300$, smallest context dimension with error $\leq 10^{-2}$ (Krylov projection / anchored hold). Bold marks the smallest entry in each row.

Task	n	Raw	Statistics	Sparse TD	Answers	Ours	LB
Block cycle, exact	24	11.44	2.44	**0.75**	9.38	2.49	1.17
Block cycle, exact	240	114.4	226.9	36.2	93.8	**24.4**	11.3
Block cycle, exact	480	228.8	903.8	102.2	187.5	**48.8**	22.5
Dense kernel, $\epsilon = 10^{-2}$	200	2345	157.8	139.0	234.4	**15.7** / 17.2	–
Goal gridworld, $\epsilon = 10^{-2}$	126	1173	63.0	**5.7**	147.7	38.5 / 44.5	–

Table 5 extends Table 2 with an answer table storing $\widehat{v}_1, \ldots, \widehat{v}_L$ (nL values) and with the sampled tasks of Figure 3. On the dense kernel, contexts accurate to 10^{-2}, well below the statistical error of 0.22, are $9\times$ smaller than sparse TD and $150\times$ smaller than the raw batch; on the sparse goal-directed gridworld, sparse TD remains the most compact representation. Storage counts float32 values and int32 indices of the retained representation only, excluding shared parameters. Raw TD keeps records (12 bytes each), visit counts, the current value vector, and a query: $12N + 8n + 4$ bytes. Statistics keep $\widehat{A}$, $\widehat{r}$, and the value vector: $4(n^2 + 2n)$ bytes. Sparse TD keeps CSR row pointers, column indices and values, reward, value, and an output buffer: $4(n + 1) + 8\,\mathrm{nnz} + 12n$ bytes. Answers keep $\widehat{v}_1, \ldots, \widehat{v}_L$: $4nL$ bytes. Krylov keeps $4(2n + 1)(k + 1)$ bytes, and the anchored hold keeps $4[(2n + 1)(k + 1) + (n + 1)j]$ bytes. Dense-block cycles have 12 blocks of size $n/12$; each record's successor is uniform in the next block; the reward is 1 on block 0; there are $N = 40n$ records, and state labels are permuted (seeds $54000 + n$). Row sums into the next block are exactly one and rewards are block-constant, so $k_L = 12$ algebraically; the attention encoder

retains exactly 12 tokens with error 0 relative to TD. Approximate rows use seed 2000 and search $m \in \{1, \ldots, 39\} \cup \{40, 45, \ldots\}$ for the first budget with error at most 10^{-2}.

Peak memory. The Peak column of Table 2 is the peak task memory of Definition 4.2, the maximum over both encoding stages and execution; Table 6 lists the three terms. The two encoding peaks are measured in the prototype as the live fields of all token groups plus the working storage of the heads of the current layer (every pending output and the largest score block, an upper bound on what is live at any instant). Stage 1 streams the raw records in chunks of $C = 2n$ records, so its live storage is the $O(n^2)$ state tokens, one chunk of $C(2n + 2)$ scalars, and the $C \times n$ score block and $n \times (n + 2)$ outputs of the three heads; Stage 2 holds the state tokens, the $h = \min\{n, L\}$ candidate tokens, and the scores and outputs of its heads. The execution term is the analytic budget of one query, the retained context plus the $k_L(k_L + 1)$ source–source scores; the prototype executor instead answers all n coordinate queries at once and keeps the whole trajectory for the error metric, which is not what Definition 4.2 counts. All three terms are $\Theta(n^2)$, the same order as the statistics, and the measured total is 12–16$\times$ the statistics because the prototype stores five width-n blocks per state token (identity, counts, empirical and discounted rows, and visit weights), fixed-capacity candidate arrays, and the full head outputs; a dense numerical implementation would keep two blocks. The benefit claimed is the retained context, i.e., what must persist to answer all later state and depth queries, not peak memory or speed.

Table 6: **Retained and peak live storage** (KiB, float32) of the exact dense-block runs (`exp_memory_precision.py`), with Stage 1 streamed in chunks of $2n$ records; "chunk" is the part of the Stage-1 peak occupied by one chunk of one-hot records, and both stage peaks include the heads' working storage. Execution is the analytic budget of one query. Total is the peak task memory of Definition 4.2.

n	Retained	Statistics	Peak, Stage 1	of which chunk	Peak, Stage 2	Peak, execution	Total
24	2.49	2.44	28.1	9.4	37.5	3.1	37.5
96	9.80	36.75	436.5	145.5	582.0	10.4	582.0
240	24.43	226.88	2711.3	903.8	1947.2	25.0	2711.3
480	48.80	903.75	10822.5	3607.5	5985.0	49.4	10822.5

Precision. The identity checks of Appendix H.1 and all curves are computed in float64, whereas the storage in Table 2 counts float32 values. Table 7 reports the same tasks under one data type at a time. Storing the float64 context in float32 and executing in float32 reproduces the TD trajectory to between 2.0×10^{-7} and 6.2×10^{-7} of its largest entry, i.e., to float32 precision; running the whole pipeline in float32 (statistics heads, Krylov encoder with rank tolerance 10^{-5}, executor, and a reciprocal that keeps the float32 type, so that no float64 intermediate is used) retains $k_L = 12$ tokens on all four sizes and is accurate to between 2.0×10^{-7} and 4.5×10^{-7}; under a different BLAS the float32 figures move within the same order of magnitude. In float64 every storage count doubles, so the savings relative to dense statistics are unchanged, and the reproduction is accurate to between 1.6×10^{-16} and 2.9×10^{-15}. The script is `exp_memory_precision.py`.

Table 7: **Memory and TD error under one data type.** Retained memory of the Krylov context and total peak in float32 and float64 (KiB), and the maximum absolute error of the executed trajectory relative to $\max_{\ell,s} |\widehat{v}_\ell(s)|$ for the float64 pipeline, for the float64 context stored and executed in float32, and for the whole pipeline in float32.

	Retained		Total peak		Max normalized error		
n	float32	float64	float32	float64	float64	float32 context	float32 pipeline
24	2.49	4.98	37.5	75.0	1.6×10^{-16}	2.0×10^{-7}	2.0×10^{-7}
96	9.80	19.60	582.0	1164.0	1.7×10^{-15}	2.4×10^{-7}	2.7×10^{-7}
240	24.43	48.85	2711.3	5422.5	9.6×10^{-16}	6.2×10^{-7}	4.5×10^{-7}
480	48.80	97.60	10822.5	21645.0	2.9×10^{-15}	3.3×10^{-7}	4.2×10^{-7}

The lower-bound column. The column LB in Table 5 is the bound $nk_L + k_L$ of Theorem 4.3(c), which holds for every encoder that is continuous on a relatively open set of strictly positive rank-k_L models (Theorem C.1). It is a bound on what such an encoder must retain on *some* model of that class, not on what the block-cycle tasks themselves require: on these tasks every row of $\widehat{P}$ moves to the next block and the reward is block-constant, so $\widehat{v}_\ell = \sum_{j<\ell} \gamma^j \mathbf{1}_{B_{-j}}$ depends on the model only through the block partition, $\mathbf{1} \in \mathcal{K}_L$, and $\widehat{P}$ has zero entries; an encoder built for this family alone would retain the n block labels and nothing else (Appendix C, scope of the bound). The comparison in Table 5 is therefore between the retained memory of the task-uniform encoder of Theorem 4.3(b), which depends on the task only through (n, k_L), and the smallest memory that any continuous encoder can achieve uniformly over rank-k_L dense models. The block-cycle tasks serve to instantiate $k_L = 12$ exactly; the ratio $2.2\times$ is a property of the encoder at rank 12, not a certificate of instance-level optimality on these tasks.

H.6 Learnability Experiments

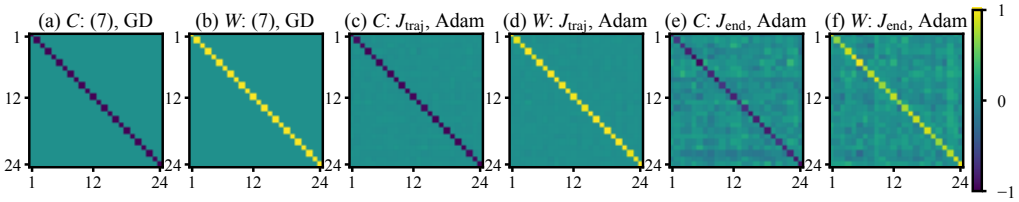


Figure 4: **Parameter identification.** Learned C and W (first seed, common color scale; targets $-I_{24}$ and I_{24}) for the calibration objective (7) with full-batch gradient descent, the trajectory objective $\mathcal{J}_{\mathrm{traj}}$ with Adam, and the endpoint objective $\mathcal{J}_{\mathrm{end}}$ with Adam. The endpoint solution matches the depth-30 output of TD but not its trajectory.

Each of 2,400 training tasks has $n = 24$ and eight observations per state ($N = 192$). The family of each task is drawn uniformly from six (dense Dirichlet, sparse with both cycle neighbours and one further distinct successor, symmetric-weight, six-block, identical-row, and permutation; 410, 394, 405, 410, 384, and 397 tasks), with statewise rewards in $\{-1, -\frac{3}{4}, \dots, 1\}$ and $\gamma \in \{0.5, 0.7, 0.8, 0.9\}$; every generated kernel is verified to be row-stochastic, and 120 tasks are held out. The operator Gram matrix $\mathbb{E}_\tau[\widehat{A}_\tau^\top \widehat{A}_\tau]$ of the training bank has eigenvalues in $[0.199, 0.545]$, so Assumption 4.8 holds with $\kappa = 0.199$. Five paired initializations with entries $\mathcal{N}(0, 0.1^2/n)$ are used for every objective, without checkpoint selection. The executor is run on the state-level context of Stage 1 with $Q_{\mathrm{eff}}(C, W)$, so that $v^\theta_{\tau,\ell} = -C^\top \sum_{j<\ell} F_\tau(C, W)^j \widehat{r}_\tau$. The three objectives are (7), whose expectation over the isotropic calibration input is taken in closed form, $2\mathcal{J} = \|C + I\|_F^2 + \mathbb{E}_\tau\|(C + I)^\top + \widehat{A}_\tau(W - I)\|_F^2$ as in the proof of Theorem 4.9, so that gradient descent is the exact deterministic iteration of the theorem (a self-test in the script compares this closed form with the literal formula on random non-symmetric parameters); $\mathcal{J}_{\mathrm{traj}}$, normalized as $\frac{1}{L}\sum_{\ell \le L} \|v^\theta_{\tau,\ell} - \widehat{v}_{\tau,\ell}\|^2 / \|\widehat{r}_\tau\|^2$ with $L = 30$; and $\mathcal{J}^{(30)}_{\mathrm{end}} := \mathbb{E}_\tau\|v^\theta_{\tau,30} - \mathrm{sg}(\widehat{r}_\tau + \widehat{A}_\tau v^\theta_{\tau,30})\|^2 / \|\widehat{r}_\tau\|^2$, the depth-30 bootstrapped surrogate of $\mathcal{J}_{\mathrm{end}}$ with a stopped-gradient target (the infinite-depth $\mathcal{J}_{\mathrm{end}}$ of Section 4.3 is not computable at finite depth, so the experiment tests the surrogate that a practitioner would minimize). Full-batch gradient descent on $\mathcal{J}/n$ uses step size 4, below the certified $n/\lambda_+ \approx 9.17$, for 3,000 steps. Adam (Kingma & Ba, 2015) uses minibatches of 128 tasks, learning rate 10^{-2} with cosine decay to 10^{-4} over 6,000 steps, and gradient clipping at 10, for all three objectives. We report the normalized parameter error $\sqrt{(\|C + I\|_F^2 + \|W - I\|_F^2)/(2n)}$ (equal to 1 at initialization) and the held-out error $\max_{\ell \le 30} \|v^\theta_{\tau,\ell} - \widehat{v}_{\tau,\ell}\| / \|\widehat{r}_\tau\|$ averaged over the 120 held-out tasks.

Results. Gradient descent on (7) reaches parameter error 1.0×10^{-15} on all five seeds (held-out error 6.8×10^{-15}), at the geometric rate of Theorem 4.9; Adam on the same objective reaches 3.2×10^{-16}. Adam on $\mathcal{J}_{\mathrm{traj}}$ reaches 7.4×10^{-2} (range 6.5–8.4 $\times 10^{-2}$ over seeds; held-out error

7.6×10^{-2}), with the diagonals of C and W at -0.992 and 0.992 and off-diagonal entries of root-mean-square size 1.4×10^{-2}; continuing the same schedule to 20,000 steps on three seeds lowers the parameter error to 3.3×10^{-3}. The slow tail is a conditioning effect, not a failure of identification: along the scale family $(-aI, aI)$ the loss $\mathcal{J}_{\mathrm{traj}}$ is 0.15 at $a = 0.5$ and 5×10^{-3} at $a = 0.9$, a shallow valley, whereas leaving the family changes the fixed point and is penalized by every deep term, so first-order methods progress slowly along the valley; the isotropic objective (7) has no such valley, which is why it carries the global guarantee. Adam on $\mathcal{J}_{\mathrm{end}}^{(30)}$ stops at parameter error 0.51 (range 0.50–0.54), with diagonals at -0.81 and 0.82 and off-diagonal entries of size 1.0×10^{-1}. This is the degeneracy of Proposition 4.7 in action: on held-out tasks the depth-30 output is within 0.038 of TD and its bootstrap residual is 0.011 (TD's own residual is 0.004), while the trajectory is wrong, with errors 0.50, 0.42, 0.29, 0.17 at depths 1, 2, 5, 10 and a maximum of 0.57 (0.63 averaged over seeds); the endpoint objective fixes the fixed point of the learned update, not the update. The scripts are `exp_learnability.py` and `make_figure_params.py`.

H.7 Finite-Precision Rank Boundary

On the two-state swap of Appendix A.10 with rewards $(1, 1+\varepsilon)$, $\gamma = 0.9$, and $L = 10$, we compare the literal unbalanced one-pass float64 encoder (L) with the balanced two-pass encoder with tolerance 10^{-12} (B). Unbalanced duals grow like $1/\varepsilon$, whereas balancing keeps context entries bounded

Table 8: **Rank-boundary sensitivity.** Retained rank (L/B), largest context entry, and maximum all-state, all-depth error. At $\varepsilon = 10^{-16}$ the float64 input already equals one.

ε	Rank L/B	Prompt L	Prompt B	Error L	Error B
0	1/1	1.0	2.0	3.6×10^{-15}	0
10^{-4}	2/2	1.1×10^{4}	2.0	2.0×10^{-12}	3.6×10^{-15}
10^{-8}	2/2	1.1×10^{8}	2.0	1.5×10^{-8}	8.9×10^{-16}
10^{-12}	2/1	1.1×10^{12}	2.0	1.5×10^{-12}	3.1×10^{-12}
10^{-16}	1/1	1.0	2.0	3.6×10^{-15}	0

(`exp_stability.py`). The tolerance then suppresses a genuinely nonzero but negligible direction, which is exactly the behavior that the ε-rank of Proposition D.3(b) makes principled.

H.8 Feature-Space Policy Evaluation (Table 2)

The bottom rows of Table 2 run the un-normalized TD update (6) of Theorem 4.4 on single sampled trajectories. ***Boyan's chain*** (Boyan, 2002): 13 states, four hat features, reward -3 per step (-2 on the last transition), $\gamma = 0.95$, one trajectory of $N = 20{,}000$ steps, $L = 100$. ***Mountain car***: the standard dynamics with a fixed energy-pumping policy ($a = \mathrm{sign}(\dot{x})$, $\epsilon = 0.1$ random actions), reward -1 per step, termination at $x \geq 0.5$ with zero successor features, $\gamma = 0.99$, $L = 200$; features are eight 8×8 tilings of the position–velocity box with random offsets ($d = 512$, eight active tiles per state), of which 417–452 occur in the trajectories of $N \in \{5{,}000, 20{,}000, 80{,}000, 320{,}000\}$ steps. Stage 1 forms A_N and b_N on all 512 coordinates by the two heads of Appendix A.6, streamed in chunks of $2d$ records; coordinates that never occur have zero rows and never move, and Stage 2 runs on the active ones. The step size is a public input chosen before encoding from the spectrum of A_N, $\alpha = c/\lambda_{\max}(\mathrm{sym}\, A_N)$ with the largest $c \in \{0.9, 0.45, \ldots\}$ for which $|1 - \alpha\lambda| < 1$ for every nonzero eigenvalue λ of A_N; this choice is made by the experimenter, not by the selector-head network, which receives α as a constant. Eigenvalue zero arises from the tilings, whose indicators sum to one, and is harmless because b_N has no component along the corresponding left null vectors. Errors are $\max_{\ell \leq L} \|\Phi(\theta_\ell - \widetilde{\theta}_\ell)\|_w / \max_\ell \|\Phi\theta_\ell\|_w$ over the visited states with uniform weights, i.e., relative to the scale of the trajectory. On Boyan's chain the exact context has $k_L^\phi = 4 = d$ and reproduces the trajectory to 9.4×10^{-16}; Krylov projection with $m = 1, 2, 3$ has error 0.95, 0.34, 0.14. On mountain car the exact encoder retains $h = \min\{d, L\} = 200$ tokens at tolerance 10^{-12}, but the trajectory matrix has at most 10 singular values above $10^{-2}\sqrt{L}$, and Krylov projection reaches relative error 9.1×10^{-3}–1.06×10^{-2} at $m = 32$ (the largest value at $N = 5{,}000$), at most 3.0×10^{-7} at $m = 48$, and at most 6×10^{-15} at $m = 64$ over the four values of N. Sample-and-hold uses the finite-depth schedule of Proposition D.9(b) with the measured contraction factor

$\rho = \max_\lambda |1-\alpha\lambda| \in [0.99998, 0.99999]$ of $I - \alpha A_N$; since $\lfloor 1/(1-\rho) \rfloor > L$, all m tokens are hold tokens at the quantiles of the mass through depth L, which here are nearly uniformly spaced, and the encoder keeps exactly the requested number of tokens up to $m = 192$ and 200 tokens for $m \geq 200$. Its relative error is 0.31–0.33 at $m = 32$, 0.19–0.21 at $m = 64$, 0.07–0.08 at $m = 128$ and $m = 192$, and below 10^{-15} once every depth is sampled (requested $m \geq 200$, 200 tokens): the increments do not decay within $L = 200$, so holding misses one full increment wherever two consecutive depths are not both sampled, the short-horizon regime of Proposition D.9(d). The retained memory of the dimension-32 context is 108–117 KiB against 681–800 KiB for the LSTD statistics A_N, b_N, and it does not depend on N, whereas the sparse raw prompt grows from 332 KiB to 21 MiB. The peak column of Table 2 is measured as in Appendix H.5, including the heads' working storage: on mountain car it is the Stage-1 peak on all 512 coordinates (9,226 KiB, of which one chunk of $2d$ raw records is 4,100 KiB), and on Boyan's chain the Stage-2 peak (1.1 KiB). The script is `exp_features.py`.

H.9 Learned Bottleneck Contexts (Figure 5)

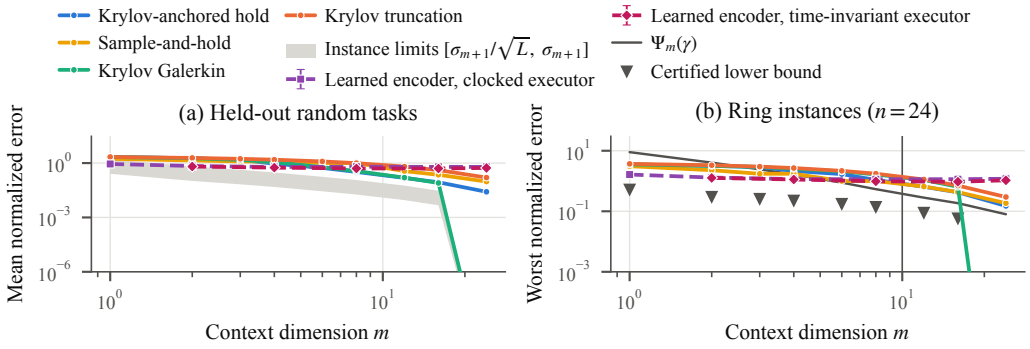


Figure 5: **Learned bottleneck encoders against compiled contexts.** (a) Mean normalized error on 256 held-out tasks and (b) worst error over three ring instances, for standard Transformer encoders with a clocked or time-invariant executor (three seeds per point, 8,000 updates each; error bars are one standard deviation across seeds), the compiled contexts on the same batches, Ψ_m, and the certified lower bound for every m-dimensional context. The vertical line marks $m = 1/(1-\gamma)$.

Model. The input is the output of the fixed first stage of Section 4.1, the iterates $\widehat{v}_1, \ldots, \widehat{v}_L$. Inside the training loop they are produced by a NumPy reference of that stage (direct counting and the TD recurrence), which agrees with the selector-head implementation to 6.3×10^{-15} on the training tasks; the heads themselves are not executed during training. The encoder reads the L tokens $(\widehat{v}_\ell, \ell/L) \in \mathbb{R}^{n+1}$. They are embedded to width $d = 64$ and processed by two standard pre-LayerNorm Transformer encoder layers (four softmax heads, GELU MLP of width 128, full attention over the L tokens). m learned latent vectors then cross-attend to the encoded records in two rounds, each followed by self-attention among the latents and an MLP, as in Perceiver-style bottlenecks (Jaegle et al., 2021); a final LayerNorm gives $z_u \in \mathbb{R}^d$ for $u \in [m]$. The latents are turned into a context for the executors of Definition A.1. A linear map sends z_u to the decoder column $d_u \in \mathbb{R}^n$, giving $D \in \mathbb{R}^{n \times m}$. In the *clocked* model the source scalars are depth indexed, $s_{\ell,u} = \mathrm{MLP}(z_u, e(\ell))$ with a sinusoidal depth code $e(\ell)$, which generalizes the gates $\mathbf{1}\{\ell+1 = \tau_u\}$ of the depth-counter executor. In the *time-invariant* model a linear readout gives $s_0 \in \mathbb{R}^m$, a bilinear form of the latents gives $K \in \mathbb{R}^{m \times m}$, rescaled so that $\|K\|_2 \leq \gamma$ (a numerically computed spectral norm, used only in this learned read-out), and $s_{\ell+1} = K s_\ell$, which is the recurrence of the time-invariant executor with (D, K, s_0) produced by the encoder. In both cases the output is $\widetilde{v}_\ell = D \sum_{j<\ell} s_j$, so the model lies in $\mathfrak{C}_m$ and the lower bounds of Section 4.2 apply. Nothing is hand-set; each model has about 1.9×10^5 parameters. Two other interfaces were tried with the same encoder and read-out and are reported below: raw transition tokens $(e_{S_t}, e_{S_{t+1}}, R_{t+1})$, and the n state-statistics tokens $(e_i, \widehat{A}_{i,:}, \widehat{r}_i)$. A pilot with a looped standard Transformer block as the

coefficient generator, trained through $L = 40$ loops from raw records, did not leave the plateau either (error 1.9 for every m, against 2.6 for the zero predictor and 2.1 for the best rank-one context along $\widehat{r}$), which is why the executor classes are used as the read-out; this pilot is not part of the code release.

Tasks and training. $n = 24$, $\gamma = 0.9$, $L = 40$, $N = 192$ records per task with every state visited at least once; the empirical model and its trajectory are computed from these records by the NumPy reference of the fixed first stage (direct counting and the TD recurrence), which Appendix H.1 verifies against the selector-head implementation. Tasks are drawn uniformly from six families: dense Dirichlet rows, sparse rows with two to four successors, symmetric-weight (reversible) chains, four-block cycles, single n-cycles, and random permutations; rewards are Gaussian on all states or on one to three states, scaled to $\|\widehat{r}\|_\infty = 1$. The target is the empirical trajectory $\widehat{v}_1, \dots, \widehat{v}_L$ of the batch, and the loss is $\frac{1}{L}\sum_\ell \|\widehat{v}_\ell - \widetilde{v}_\ell\|_w^2 / \|\widehat{r}\|_w^2$ with visit weights w. One model is trained per $(m, \text{executor type}, \text{seed})$ with AdamW (learning rate 10^{-3}, cosine decay to 10^{-5} after 200 warm-up steps, weight decay 0.01, gradient clipping at 1) for 8,000 steps of 32 fresh tasks each, on a CPU in float32 with two threads per run, with three seeds (0, 1, 2) for every point of the grid $m \in \{1, 2, 4, 8, 16, 24\}$ (clocked) and $m \in \{2, 4, 8, 16, 24\}$ (time-invariant); an earlier single-seed sweep with 4,000 steps is kept in the code release. No checkpoint or seed selection is used. Evaluation uses the metric of Section 5, $\max_{\ell \le L} \|\widehat{v}_\ell - \widetilde{v}_\ell\|_w / \|\widehat{r}\|_w$, on 256 held-out tasks from the same distribution (seed 777) and on three deterministic ring instances of size $n = 24$ realized as covered batches: the single-reward ring and rewarded arcs of width 2 and 3 (Appendix D). The certified lower bound in Figure 5(b) is the water-filling width of the orthogonal block increments of these instances (Lemmas D.5 and D.6), valid for every context in $\mathfrak{C}_m$. The compiled contexts are evaluated on exactly the same batches through the attention implementation.

Results. Table 9 lists the numbers behind Figure 5 as means and standard deviations over the three seeds. Four facts stand out. First, below the effective horizon the learned encoders are better than the compiled contexts on typical tasks: for $m \le 8$ their held-out error (0.50–0.89) is below sample-and-hold (1.00–1.72), and for $m \le 4$ below the anchored hold (0.93–2.17); on the rings they are below sample-and-hold for $m \le 4$ and far below the worst-case guarantee $e^2\Psi_m$. The compiled contexts are tuned to the worst case and spend their first tokens on the earliest increments, which a learned encoder need not do. Second, the learned error does not decrease with m: on held-out tasks it stays at 0.61–0.63 for every $m \ge 4$ with the clocked executor and at 0.50–0.54 for every $m \ge 8$ with the time-invariant one, and on the rings at 1.14–1.19 and 0.95–1.05, with log–log slopes over $2 \le m \le 8$ of -0.07 and -0.18 on held-out tasks and -0.09 and -0.19 on the rings, against -0.59 for sample-and-hold and -1.46 for Ψ_m. From $m = 8$ on the ring error exceeds Ψ_m, and at $m = 24$ it exceeds the upper bound $e^2\Psi_m$ by 2.0× (clocked) and 1.8× (time-invariant): a learned encoder with the full budget is worse than the sample-and-hold guarantee, while the compiled contexts reach 0.19 (SH) and 0.15 (anchored) and the held-out error of the anchored hold is 0.026. Third, the seed spread is small compared with these gaps: the standard deviations are 0.003–0.06 on held-out tasks (0.11 at $m = 2$ for the time-invariant model) and 0.01–0.11 on the rings, so the flatness in m is not a seed effect. Fourth, every learned error lies above the certified lower bound, by 3.1× at $m = 1$ and 16–19× at $m = 16$, as Lemma D.5 requires of any context in $\mathfrak{C}_m$. Per family, the learned encoders reach 0.35–0.55 on the dense, reversible, and block tasks, 0.53–0.64 on the sparse ones, and 0.81–0.94 on the deterministic cycles and permutations, at every m; the compiled contexts are worse than this on the deterministic families at $m \le 4$, whereas by $m = 24$ the anchored hold is exact on the dense, reversible, block, and sparse families and reaches 0.03–0.11 on the deterministic ones. The time-invariant executor is better than the clocked one on held-out tasks for $m \ge 4$ (0.50–0.56 against 0.61–0.63) and on the rings; with (K, s_0) read off the latents it is a learned counterpart of the Krylov context, and its error is also flat in m. Doubling the training budget from the earlier 4,000-step single-seed sweep (held-out errors 0.74–0.77 for the clocked and 0.67–0.71 for the time-invariant models at $m \ge 4$) lowered every curve by 0.1–0.2 but did not change its shape, and at the end of training the held-out error was still decreasing by at most 0.02 per 1,000 steps, so a larger budget would lower the curves further but, on this evidence, not make them decrease with m. The two other interfaces of the 4,000-step pilots give 1.91 (raw records) and 1.82 (statistics tokens) at $m = 8$ with the clocked executor, against 0.75 from the iterates; both are close to the zero predictor (2.58) and to the best rank-one context along $\widehat{r}$ (2.09), so they did not leave a rank-one plateau. We state the finding plainly: a standard Transformer with a bottleneck learns to place a few tokens

better than the worst-case-optimal schedules do on typical tasks, but it does not discover the Krylov or clock structure that turns additional tokens into accuracy; supplying that structure is what the compiled encoders of Sections 4.1 and 4.2 contribute.

Table 9: **Learned bottleneck encoders** (trajectory-token interface; mean $\pm$ one standard deviation over three seeds, $8{,}000$ updates each) against the compiled sample-and-hold (SH) and anchored contexts on the same batches, the upper bound $e^2\Psi_m$ of Theorem 4.6, and the certified lower bound (LB) of the ring instances. Dashes: budget not trained, or no certificate.

	Held-out tasks, mean error				Ring instances, worst error				
m	clocked	time-inv.	SH	anchored	clocked	time-inv.	SH	$e^2\Psi_m$	LB
1	0.89 ± 0.01	–	1.72	2.17	1.64 ± 0.01	–	3.08	66.50	0.53
2	0.69 ± 0.00	0.64 ± 0.11	1.39	1.85	1.30 ± 0.01	1.26 ± 0.02	2.28	29.93	0.31
4	0.61 ± 0.03	0.56 ± 0.04	1.23	0.93	1.16 ± 0.03	1.12 ± 0.11	1.73	12.12	0.23
8	0.63 ± 0.03	0.51 ± 0.00	1.00	0.33	1.14 ± 0.02	0.98 ± 0.04	1.00	3.98	0.14
16	0.62 ± 0.06	0.50 ± 0.00	0.23	0.087	1.15 ± 0.06	0.95 ± 0.05	0.43	1.37	0.060
24	0.61 ± 0.01	0.54 ± 0.05	0.096	0.026	1.19 ± 0.02	1.05 ± 0.09	0.19	0.59	–

I Scope and Limitations

We collect the qualifications here rather than repeating them in each proof. **Computational model.** Exact statements use real arithmetic and declared primitives: recip_0, token release, and, for certified adaptive rank, a threshold gate. Exact rank selection over arbitrary reals requires a discontinuous primitive (Appendix A.10). Clocked contexts treat γ as public, although in-network schedules are possible. Appendix B realizes both executors with standard two-head masked softmax attention; that realization is a compilation whose encoder writes log-probability codes into the tokens, and it says nothing about what trained softmax models do. The bottleneck encoders of Appendix H.9 are the only trained encoders in this paper; they read the iterates of the fixed first stages and are trained on our task distribution, not on natural pretraining data. **What is optimal.** Theorem C.1 counts reals for encoders that are continuous in the model on dense model neighborhoods; sparse models can be stored more compactly. Theorem 4.6 is a worst-case statement with absolute but unoptimized constants (measured gaps are within $20\times$). Proposition D.3 supplies instance-dependent guarantees. Proposition D.1 concerns moment-matching encoders, not all time-invariant recurrences. The depth bounds of Theorem E.1 are tight up to the base of the logarithm ($\log_2 3$) and additive constants; the lower bound is for polynomial executors, which excludes recip_0, so it constrains the doubling and time-invariant executors but not the depth-counter one, whose unrolled recip_0-free variant has depth L. **Resources not reduced.** Peak memory and arithmetic are not reduced below those of the statistics or direct TD; the benefit is the retained, reusable context. For sparse models, sparse numerical TD can be smaller. **Learning.** Only the shared executor is shown to be learnable; joint training of encoders is open. **Setting.** The results cover fixed-policy evaluation of empirical, visitation-normalized TD. In control, the Krylov space changes with the policy, and extending the laws there is open.