Linear approximations to HMM filtering

Andrew Mah*
Flatiron Institute
New York, NY
amah@flatironinstitute.org

Joshua L. Pughe-Sanford*
Flatiron Institute
New York, NY
Jpughesanford@gmail.com

Sarah E. Harvey
Flatiron Institute
New York, NY
sarah.e.harvey1@gmail.com

Alex H. Williams
Flatiron Institute, New York University
New York, NY
awilliams@flatironinstitute.org

Abstract

Complex sequence models, such as transformers and state space models (SSMs), learn to represent latent belief states when trained on next-token prediction. Surprisingly, this phenomenon can also be captured by a recurrent neural network (RNN) with linear dynamics and a static nonlinear readout, which we call a *linear RNN*. This raises a natural question: when are linear RNNs sufficient for optimal belief state approximation? We study this question in the canonical setting of hidden Markov models (HMMs), where Bayes-optimal prediction requires nonlinear filtering. Under mild assumptions, we characterize the full class of HMMs whose filtering dynamics are exactly realizable by linear RNNs with a softmax readout. Outside this class, we prove a dissipation relation bounding the approximation error achievable by such linear RNNs in terms of properties of the HMM. This bound suggests two mechanisms that make HMMs "simple" to filter: fast contraction and concentration about the uniform belief state. We validate our theoretical results through experiments on both randomly sampled HMMs and constructed adversarial examples. Together, these findings help delineate when linear RNNs suffice for optimal inference and when more expressive models are needed.

1 Introduction

Sequence models, including recurrent neural networks [RNNs; 1, 2, 3], transformers [4] and deep state-space models [SSMs; 5, 6, 7, 8], can achieve strong empirical performance when trained on next-token prediction. When sequences are generated by a latent-state process, Bayes-optimal prediction is naturally expressed in terms of a belief state: a probability distribution over the unobserved latent state, conditioned on past observations [9]. For hidden Markov models [HMMs; 10], the Bayes-optimal filter updates this belief recursively, yielding a deterministic but nonlinear dynamical system whose predictions minimize next-token prediction error.

Recent work suggests that these sequence models represent the latent belief state [11, 12, 13] and may implicitly learn a form of Bayesian filtering [14, 15, 16], although this view has been contested [17]. Given the scale and complexity of these models, proving analytical results about their underlying computations is difficult, even though such results are a central goal of mechanistic interpretability. Interestingly, we find that linear RNNs, which are far simpler and analytically tractable, can learn belief state representations nearly identical to those of transformers on recently investigated sequence tasks [11, 12], as illustrated in Figure 1. This observation adds to a growing

*Equal Contributions

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

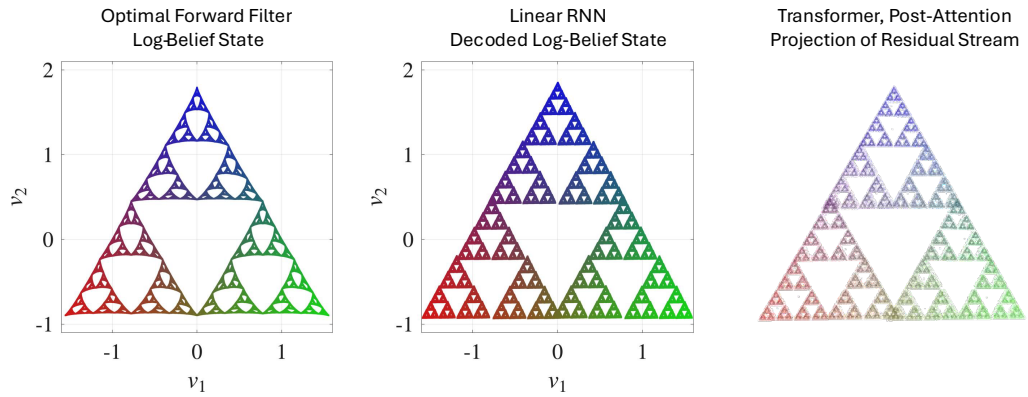


Figure 1: **Linear RNNs learn similar belief state geometry to optimal forward filters and transformers.** Log-belief states of the optimal forward filter (left) and of a trained linear RNN (center) on the Mess3 HMM (3 latent states), projected onto the plane orthogonal to $\mathbf{1}$: $v_1 = \log(\mathbf{p})^\top(\sqrt{1/2}, -\sqrt{1/2}, 0)$ and $v_2 = \log(\mathbf{p})^\top(-\sqrt{1/6}, -\sqrt{1/6}, \sqrt{2/3})$. The RNN's log-beliefs are read out through its own learned affine map $M\mathbf{h}_t + \mathbf{b}$. Color corresponds to the belief state $\mathbf{p}$ interpreted as an RGB triplet. Right: belief state representation of a trained transformer, reproduced from Figure 1(d) of Piotrowski et al. [12] (CC BY-NC-ND 4.0). See Appendix B for more details.

body of work demonstrating the strong performance of linear models on sequence-to-sequence tasks [18, 19].

Here, we seek to determine circumstances under which Bayesian filtering can be accurately approximated by a linear RNN. We study this question at the level of representational capacity, independent of training dynamics [20] or parameter identifiability [21, 22, 23]. First, under mild assumptions, we enumerate all HMMs whose forward filter can be exactly realized by a linear RNN with a softmax readout. Second, for general HMMs, we prove a dissipation relation that upper-bounds the approximation error achievable by such linear RNNs. This bound suggests two mechanisms that make an HMM easy for linear RNNs to filter: fast contraction of the transition dynamics and, for approximately doubly stochastic transitions, weakly informative emissions that keep beliefs near uniform. We validate these predictions empirically across a variety of HMMs.

2 Prior Work

Linear approximations to Bayesian filtering. Linear approximations to nonlinear filtering have been extensively studied in continuous-state systems. The extended Kalman filter, a nonlinear extension of the linear Kalman filter [24], approximates nonlinear dynamics using a Jacobian-linearization scheme. Recent advances in Koopman operator theory have produced approximation error bounds for both extended dynamic mode decompositions [25, 26] and kernel-based approximations [27, 28, 29]. This body of work and ours share the broad goal of understanding when linear dynamical systems can approximate nonlinear evolution, particularly in the partially observed setting. However, Koopman-based methods typically seek a linear evolution law for observation time series after lifting them into an infinite-dimensional function space. In contrast, filtering tracks the evolution of probability distributions on the simplex. Filtering presents distinct challenges, wherein evolution remains nonlinear and the natural geometry is projective rather than Euclidean.

For discrete state HMMs specifically, Rao [30] constructed linear recurrent networks that approximate Bayesian filtering for three hand-engineered examples, demonstrating empirically that approximation error can remain small on test sequences. However, this work provided no general theory relating approximation error to HMM properties and analyzed only a handful of specific cases.

Neural networks as approximate filters. Modern sequence models are typically trained end-to-end on predictive objectives (e.g., next-token prediction) rather than explicit parameter estimation of the underlying data-generative process. As such, understanding what these models learn remains challenging. Recent work has begun to analyze the internal representations of trained neural networks through the lens of Bayesian filtering. Studies of transformers [11, 31] and recurrent networks

[32, 33, 34] have found evidence that these models develop internal representations resembling belief states over latent variables that are updated in response to observations. Relatedly, networks trained on predictive objectives can learn to output approximate Bayesian posteriors [35, 36].

These results motivate our focus on understanding linear approximations to optimal filters. If neural networks trained on prediction tasks implicitly learn to approximate filtering, then characterizing the limits of linear approximations provides insight into what representations and computations are fundamentally necessary for accurate sequential prediction. In Section 4.1, we identify which HMM properties govern how well linear RNNs approximate filtering, suggesting when more expressive architectures may be needed.

3 Forward Filters, Recurrent Neural Networks, and Readout Functions

Consider an HMM with latent states $x_t \in \{1, \ldots, n\}$ and emissions $y_t \in \{1, \ldots, m\}$. The latent state dynamics are Markovian, with transition probability matrix $T \in \mathbb{R}^{n \times n}$ and elements

$$T_{ij} = \mathbb{P}(x_{t+1} = i \mid x_t = j).$$

Emissions are observed with probability

$$E_{ij} = \mathbb{P}(y_t = i \mid x_t = j).$$

Given a sequence of observations $y_{1:t} = (y_1, y_2, \ldots, y_t)$ from this generative model, the belief state $p_{t,i} = \mathbb{P}(x_t = i \mid y_{1:t})$ is a sufficient statistic for the observation history [37]; predictive and control tasks can be solved optimally using only this statistic [9]. The forward algorithm [38] computes the belief state recursively via the joint probability $\rho_{t,i} = \mathbb{P}(x_t = i, y_{1:t})$,

$$\boldsymbol{\rho}_t = (E^\top \mathbf{u}_t) \odot (T \boldsymbol{\rho}_{t-1}) \qquad\qquad \mathbf{p}_t = \frac{\boldsymbol{\rho}_t}{\mathbf{1}_n^\top \boldsymbol{\rho}_t}, \tag{1}$$

where $u_{t,i} = \delta_{i y_t}$ is a one-hot encoding of the observation at time t and $\odot$ denotes the Hadamard product, $(\boldsymbol{\rho} \odot \boldsymbol{\rho}')_i = \rho_i \rho'_i$. The belief state is recovered at every step via normalization of the joint probability. A derivation of the forward recursion is provided in the appendix.

We compare this dynamical system to that of an RNN with linear recurrent dynamics:

$$\mathbf{h}_t = F \mathbf{h}_{t-1} + G \mathbf{u}_t \qquad\qquad \hat{\mathbf{p}}_t = \boldsymbol{\phi}(M \mathbf{h}_t + \mathbf{b}), \tag{2}$$

where $\mathbf{h}_t \in \mathbb{R}^k$ are the latent activations; $\mathbf{u}_t$ is the same as above; $F \in \mathbb{R}^{k \times k}$, $G \in \mathbb{R}^{k \times m}$, and $M \in \mathbb{R}^{n \times k}$ are finite matrices; and $\mathbf{b} \in \mathbb{R}^n$ is a bias. Finally, $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^n$ is a static function whose output estimates the belief state. We refer to Equation 2 as a *linear RNN*: its output is a linear state space model observed through $\boldsymbol{\phi}$. The readout, $\boldsymbol{\phi}$, plays the role of a change of variables, although it need not be invertible. We seek to understand when Equation 2 can integrate Equation 1.

Definition 1. We call an HMM, with parameters T and E, linearly $\boldsymbol{\phi}$-integrable over $\mathcal{P} \subseteq \Delta^n$ if there exist a latent dimension k and parameters M, $\mathbf{b}$, F, and G such that

I. $\mathcal{P}$ is in the range of the readout, $\mathcal{P} \subseteq \boldsymbol{\phi}(M\mathbb{R}^k + \mathbf{b})$, and

II. $\hat{\mathbf{p}}_0 = \mathbf{p}_0$ implies $\hat{\mathbf{p}}_t = \mathbf{p}_t$ for every $t > 0$, $\mathbf{p}_0 \in \mathcal{P}$, and $y_{1:t}$ with $\mathbb{P}(y_{1:t} \mid \hat{\mathbf{p}}_0) > 0$.

Above, $\Delta^n = \{ \mathbf{p} \in \mathbb{R}^n : \mathbf{p}^\top \mathbf{1}_n = 1, p_i \geq 0 \}$ denotes the probability simplex in $\mathbb{R}^n$. Property I ensures that the readout can adequately represent every belief state in $\mathcal{P}$. Property II ensures that, if the RNN and HMM share the same beliefs at time $t = 0$, then they share beliefs for all time and over all admissible input sequences.

A simple starting point is $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, corresponding to the *affine* readout:

$$\hat{\mathbf{p}} = M \mathbf{h} + \mathbf{b}. \tag{3}$$

This choice renders the RNN into a true state space model. The affine readout is not amenable to empirical study, ensuring neither normalization nor positivity during training. As a result, training under a KL loss is very unstable. These difficulties are ameliorated by the softmax function,

$\mathrm{softmax}(\mathbf{x}) = \exp(\mathbf{x})/\mathbf{1}_n^\top \exp(\mathbf{x})$, whose output is guaranteed to be a strictly positive probability vector for all inputs. Note that, throughout this manuscript, $\exp$ and $\log$ are always evaluated component-wise, even on matrix arguments. Selecting $\boldsymbol{\phi} = \mathrm{softmax}$ gives

$$\hat{\mathbf{p}} = \mathrm{softmax}(M\mathbf{h} + \mathbf{b}), \tag{4}$$

which we refer to as the *softmax* readout.

Observe that Equation 1 is the composition of two maps: an *update* step $\boldsymbol{\rho} \mapsto T\boldsymbol{\rho}$ that contracts the belief state towards equilibrium and a *conditioning* step $\boldsymbol{\rho} \mapsto (E^\top \mathbf{u}_t) \odot \boldsymbol{\rho}$ that pushes the belief state out of equilibrium, towards a state of more certainty. Equation 1 is a forced-dissipative dynamical system on Δ^n.

By definition of the readout function, the affine-RNN represents the belief state within its latent state directly, $\hat{\mathbf{p}}_t = M\mathbf{h}_t + \mathbf{b}$. In these coordinates, it is rather simple to find an M and F that reproduce the update step, $MF \approx TM$. In contrast, the conditioning step is multiplicative in $\mathbf{u}_t$, while Equation 2 is additive. This makes conditioning a challenge for the affine-RNN.

In turn, Equation 4 enforces that a softmax-RNN represents log-beliefs. Define $\hat{\boldsymbol{\rho}} := \exp(M\mathbf{h} + \mathbf{b})$, such that Equation 4 reads $\hat{\mathbf{p}}_t = \hat{\boldsymbol{\rho}}_t/(\mathbf{1}_n^\top \hat{\boldsymbol{\rho}}_t)$, matching Equation 1. It follows that $\hat{\boldsymbol{\rho}}$ is a strictly-positive, unnormalized belief state, and that the hidden state represents its log up to an affine transformation: $\log(\hat{\boldsymbol{\rho}}) = M\mathbf{h}+\mathbf{b}$. In log-coordinates, addition *is* multiplication (of the exponentiated values). In particular, vector addition can implement the Hadamard product,

$$\log(\boldsymbol{\rho}) + \log(\boldsymbol{\rho}') = \log(\boldsymbol{\rho} \odot \boldsymbol{\rho}').$$

Hence, conditioning is simple to implement in the softmax-RNN. However, the update step, which was previously simple for the affine-RNN, becomes difficult to approximate in log coordinates [30].

Log-coordinates also introduce a challenge in representing zero probabilities. Since $\hat{\boldsymbol{\rho}}$ is strictly positive for finite $\mathbf{h}$, *a softmax-RNN will never be able to represent probability vectors with zero components* with finite network weights. This observation foreshadows many of our theoretical and empirical findings explained below.

Appendix D provides examples of both softmax- and affine-integrable HMMs. Indeed, we find that the affine-RNN can perform the update step, but struggles to implement conditioning. In turn, the softmax-RNN struggles with the update step but not conditioning.

For the remainder of this manuscript, we restrict our analysis to the softmax readout and aim to exhaustively understand this regime. The affine regime is explored, in part, by Piotrowski et al. [12]. Their constrained belief state is assembled as a baseline plus one contribution per token, each determined by that token's symbol and its lag alone, which is exactly the form a belief state takes under an affine readout, when Equation 2 is rolled out.

4 Softmax-integrable HMMs

In this section, we enumerate all HMMs that are softmax-integrable over the interior of the simplex $\mathcal{P} = \Delta^n_\circ = \{\, \mathbf{p} \in \Delta^n : \mathbf{p} > 0 \,\}$. Taking $\mathcal{P} = \Delta^n_\circ$ imposes a generalization requirement: the RNN must track the forward filter from every strictly positive prior, including counterfactual ones. Ideally one would take $\mathcal{P} = \Delta^n$, but the boundary of the simplex is not representable by a softmax readout with finite arguments, and must be excluded.

In our proofs, we assume the following:

Assumption 1. $\mathcal{P} = \Delta^n_\circ$, as stated above.

Assumption 2. T has no rows that are identically zero.

Assumption 3. $E > 0$ is strictly positive.

Assumption 4. The input, $\mathbf{u}_t$, is one-hot.

Assumption 2 and Assumption 3 are used to ensure that $\mathbf{p}_t \in \Delta^n_\circ$ remains strictly positive for all time: a requirement of softmax-integrability if F and G are to remain finite.

These are both mild assumptions. Zero rows of T correspond to hidden states that are probability zero for all $t \geq 1$. Such states can reasonably be excluded from the model. Assumption 3 ensures that

all symbol sequences are admissible, $\mathbb{P}(y_{1:t}) > 0$ for all $y_{1:t}$. For any E that is not strictly positive, one can take $E_\epsilon = (1-\epsilon)E + (\epsilon/m)\mathbf{1}_m\mathbf{1}_n^\top$, for an arbitrarily small $0 < \epsilon < 1$. Such perturbations correspond to passing each emitted symbol through a noisy channel, and have vanishing impact on HMM statistics as $\epsilon \to 0$ (see Section A.3).

Assumption 4 asserts that all observation symbols are provided their own input dimension upon which to act. In the discrete and linear settings considered here, this is purely a notational choice. The term $G\mathbf{u}_t \in \mathbb{R}^k$ can be thought of as the token embedding of y_t.

Our results proceed as follows. Given Assumption 1, Property I states that $\mathrm{softmax}(M\mathbf{h} + \mathbf{b})$ must be surjective on $\Delta_\circ^n$. This holds if and only if (see Proposition 1):

$$\mathrm{rank}([M\ \mathbf{1}_n]) = n. \tag{5}$$

This bounds the dimension of the latent state, $k \geq \mathrm{rank}(M) \geq n-1$. The minimal case, $k = n-1$, corresponds to a latent state tracking log-*odds* rather than log-*probabilities*. Beyond the rank constraint, k, M and $\mathbf{b}$ are arbitrary.

We then suppose Property I holds, in addition to the assumptions listed above. In this setting, Property II holds if and only if T is a permutation or a dense rank-one matrix (c.f. Lemma 1 and Lemma 2). For such T, RNNs that integrate the forward filter are exactly (see Theorem 1 and Theorem 2):

$$F = M^\dagger \left[\mathrm{diag}(T\mathbf{1}_n)^{-1}T\right] M + (I_k - M^\dagger M)K_1 \tag{6}$$

$$G = M^\dagger \left[\log\left(\mathrm{diag}(T\mathbf{1}_n)E^\top\right) + (MFM^\dagger - I_n)\mathbf{b}\mathbf{1}_m^\top\right] + (I_k - M^\dagger M)K_2, \tag{7}$$

where $M^\dagger$ is a right inverse of M (guaranteed to exist by Proposition 1), and $K_1 \in \mathbb{R}^{k\times k}$ and $K_2 \in \mathbb{R}^{k\times m}$ are arbitrary matrices. F and G are finite under Assumption 2 and Assumption 3.

We say that parameters $(k, F, G, M, \mathbf{b})$ are members of the *solution family* when M and k satisfy Equation 5 and F and G satisfy Equation 6 and Equation 7, respectively; $\mathbf{b}$ is arbitrary. Notice that Equation 1 is invariant under $(E, T) \mapsto (E\Lambda, \Lambda^{-1}T)$ for any diagonal and invertible matrix Λ. The exact value of Λ is unobservable to the RNN. The solution family selects $\Lambda = \mathrm{diag}(T\mathbf{1}_n)$, which is invertible under Assumption 2.

In summary, an HMM is softmax-integrable over $\Delta_\circ^n$ if and only if T is a permutation or a dense rank-one matrix. For such T, the readout is unconstrained beyond Equation 5, which requires $k \geq n-1$. The RNN weights, F and G, are given by Equation 6 and Equation 7, respectively, up to the free choice of K_1 and K_2.

4.1 Approximation error in non-exact cases

In this section, we bound the approximation error of softmax-RNNs when T is not a permutation or rank-one, dense matrix. We do so by investigating the error of the solution family, derived above, for generic T and E. This analysis bounds the error from above, as a trained RNN should perform at least as well as this reference point. We continue to reference the assumptions above as needed. For reasons discussed below, we will additionally assume

Assumption 5. $\mathbf{p}_0 \in \Delta_\circ^n$ is strictly positive.

We define the Hilbert distance between $\mathbf{p}$ and $\hat{\mathbf{p}}$ [39, 40],

$$d(\mathbf{p}, \hat{\mathbf{p}}) = \max_{ij} \log\left(\frac{p_i\hat{p}_j}{p_j\hat{p}_i}\right) = \|\log(\mathbf{p}) - \log(\hat{\mathbf{p}})\|_{\mathrm{osc}}, \qquad \mathbf{p}, \hat{\mathbf{p}} \in \mathbb{R}^n_{++}. \tag{8}$$

The Hilbert distance measures the spread, $\|\mathbf{x}\|_{\mathrm{osc}} = \max_i x_i - \min_i x_i$, of log-probabilities. It is an ideal measure of distance in log-coordinates, but it requires its inputs to be strictly positive. Assumption 2, 3, and 5 ensure that $\mathbf{p}_t \in \Delta_\circ^n$ is strictly positive for all finite t (see Proposition 8); in turn, $\mathbf{h}$ being finite ensures that the RNN output remains strictly positive. The Hilbert distance bounds the Kullback–Leibler (KL) divergence (see Lemma 4):

$$D_{\mathrm{KL}}(\mathbf{p}||\hat{\mathbf{p}}), D_{\mathrm{KL}}(\hat{\mathbf{p}}||\mathbf{p}) \leq \min\left\{d(\mathbf{p}, \hat{\mathbf{p}}), \frac{1}{8}d(\mathbf{p}, \hat{\mathbf{p}})^2\right\}. \tag{9}$$

It is also normalization invariant, $d(z\mathbf{p}, \hat{z}\hat{\mathbf{p}}) = d(\mathbf{p}, \hat{\mathbf{p}})$, for all $z, \hat{z} > 0$. Consequently, $d(\mathbf{p}, \hat{\mathbf{p}}) = d(\boldsymbol{\rho}, \hat{\boldsymbol{\rho}})$. Finally, positive linear operators (like transition matrices) admit a uniform operator bound in this metric

$$d(T\mathbf{p}, T\hat{\mathbf{p}}) \leq c_T \, d(\mathbf{p}, \hat{\mathbf{p}}) \qquad\qquad c_T = \tanh\left(\frac{1}{4}\max_{i,j} d(T_{:i}, T_{:j})\right), \tag{10}$$

where $0 \leq c_T \leq 1$ is the Birkhoff contraction coefficient of T. Please see Appendix E for a deeper discussion of the Hilbert distance.

Let T and E be generic matrices satisfying Assumptions 2 and 3; T need not be permutation or rank one. Let $(k, F, G, M, \mathbf{b})$ be a member of the solution family derived above. Then, the Hilbert distance between the true forward filter $\mathbf{p}_t$ and the RNN estimate $\hat{\mathbf{p}}_t$ satisfies a dissipation relation (see Proposition 9):

$$d(\mathbf{p}_t, \hat{\mathbf{p}}_t) \leq c_T \, d(\mathbf{p}_{t-1}, \hat{\mathbf{p}}_{t-1}) + s_{t-1} \tag{11}$$

where

$$s_t = d\big(U\hat{\mathbf{p}}_t, \exp\left(U \log(\hat{\mathbf{p}}_t)\right)\big), \qquad\qquad U = \mathrm{diag}(T\mathbf{1}_n)^{-1}T, \tag{12}$$

denotes an injected error at every time step. The bound decays geometrically in time, at a rate governed by the Birkhoff contraction coefficient, c_T. E enters only through its implicit role in selecting the belief states, $\hat{\mathbf{p}}_t$, at which s_t is evaluated.

To visualize the structure of injected error, we plot its amplitude over a family of two-state HMMs

$$T = \left(\begin{smallmatrix} 1-\tau_1 & \tau_2 \\ \tau_1 & 1-\tau_2 \end{smallmatrix}\right) \qquad E = \left(\begin{smallmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{smallmatrix}\right) \qquad \hat{\mathbf{p}} = \left(\begin{smallmatrix} 1-p \\ p \end{smallmatrix}\right). \tag{13}$$

For now, we fix $\tau_1 = \tau_2 = \tau$ to be identical, noting that $c_T = |1 - 2\tau|$ in this regime. Figure 2A plots s over τ and p. Recall that s does not depend on ϵ. Recall also that a Hilbert distance of 10^{-3} corresponds to a KL divergence less than 10^{-6}, by Equation 9. There are three vertical minima, corresponding to permutation T ($\tau = 0, 1$) and rank-one, dense T ($\tau = 1/2$). Indeed, writing $\mathbf{p} = \mathrm{softmax}(\mathbf{x})$, where $\mathbf{x} = M\mathbf{h} + \mathbf{b}$, it follows that s is zero at all belief states when U commutes through softmax

$$\mathrm{softmax}(U\mathbf{x}) = U \, \mathrm{softmax}(\mathbf{x}). \tag{14}$$

Permutation and rank-one, dense T are exactly the matrices whose U commutes for all $\mathbf{x}$. Figure 2A also contains a horizontal minimum about $p = 1/2$, which corresponds to $\mathbf{x} = \mathbf{1}_n$ ($\mathbf{p} = \mathbf{1}_n/n$) being a solution of Equation 14 for all U, since U is row stochastic.

We define the time-averaged Hilbert distance as

$$\ell_t = \tfrac{1}{t}\textstyle\sum_{\tau=1}^{t} d(\mathbf{p}_\tau, \hat{\mathbf{p}}_\tau) \quad \leq \quad \tfrac{c_T}{t}\left(\tfrac{1-c_T^t}{1-c_T}\right) d(\mathbf{p}_0, \hat{\mathbf{p}}_0) + \tfrac{1}{t}\textstyle\sum_{\tau=0}^{t-1}\left(\tfrac{1-c_T^{t-\tau}}{1-c_T}\right) s_\tau, \tag{15}$$

which we bound with $s_{0:t}$ using Equation 11. When $\hat{\mathbf{p}}_0 = \mathbf{p}_0$, this loss is zero for all softmax-integrable HMMs, and will remain small for approximately softmax-integrable HMMs. The upper bound present in Equation 15 predicts two mechanisms by which ℓ_t can be driven to zero. These mechanisms are validated in Figure 2, which computes ℓ_t empirically over a range of HMM parameters.

Mechanism 1, small c_T. Permutation and rank-one, dense matrices are both zeros of s for all belief states. However, if we compute ℓ_t numerically, see Figure 2B, we observe that transition matrices *near* rank one ($\tau \approx 1/2$) remain minima, while *near* permutation ($\tau \approx 0, 1$) do not. In other words, only the minima with $c_T \approx 0$ remain broad minima of ℓ_t as t increases.

To see why this is, suppose that T is perturbed slightly away from a rank-one, dense matrix. In the limit $c_T \to 0$, Equation 15 reduces to $\ell_t \leq \frac{1}{t}\sum_{\tau=0}^{t-1} s_\tau$. In this regime, as t increases, the time-averaged error becomes bounded by the time-average of s_t. In contrast, suppose T is near permutation. Then, in the limit $c_T \to 1$, Equation 15 becomes $\ell_t \leq d(\mathbf{p}_0, \hat{\mathbf{p}}_0) + \frac{1}{t}\sum_{\tau=0}^{t-1}(t-\tau)\, s_\tau$. The bound now integrates s_τ, becoming vacuous. Figure 2B indicates that ℓ_t diverges along with its upper bound, in this regime. This result suggests that HMM approximation difficulty scales with c_T. As c_T approaches unity, errors may become integrated over time, leading to increasingly sharp minima about softmax-integrable HMMs.

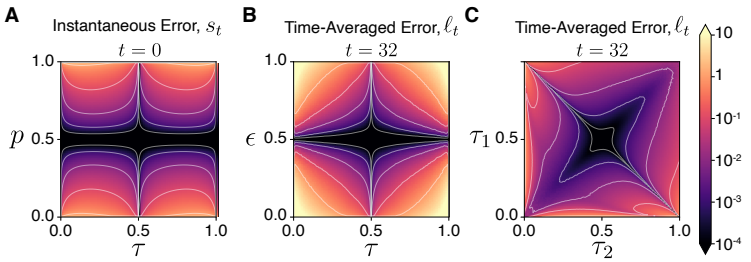


Figure 2: **Approximation error landscape across parameters in a two-state HMM. A**. Instantaneous error, s_t, Equation 12, over τ and p from Equation 13. **B–C** Time-averaged error, ℓ_t, from Equation 15, computed with $\mathbf{p}_0 = \hat{\mathbf{p}}_0 = (1/2, 1/2)^\top$ plotted (B) over τ and ϵ from Equation 13 and (C) over τ_1 and τ_2 at fixed $\epsilon = .45$. The color scale is logarithmic and shared across all panels.

Mechanism 2, uniform beliefs. Equation 12 is zero when $\mathbf{p} = \mathbf{1}_n/n$, for all T. This is illustrated in Figure 2A by the broad horizontal minimum about $p = 1/2$. Hence, if $\mathbf{p}_t$ remains approximately uniform for all time, s_t will be small as well. Furthermore, if s_t remains sufficiently small relative to the contraction rate, then ℓ_t will be small as well.

Let c_E denote the Birkhoff contraction coefficient of E. Suppose $c_T < 1$. Then, $\boldsymbol{\pi} = T\boldsymbol{\pi}$ will be the unique stationary vector of T, and it will be strictly positive (see Proposition 10). Supposing $\mathbf{p}_0 \in \Delta_\circ^n$, it follows that the distance from $\mathbf{p}_t$ to $\boldsymbol{\pi}$ will be bounded (see Theorem 4):

$$\limsup_{t\to\infty} d(\mathbf{p}_t, \boldsymbol{\pi}) \leq 4\frac{\operatorname{arctanh}(c_E)}{1 - c_T}. \tag{16}$$

This bound tends to zero with $c_E \to 0$. If we further assume T is, at least approximately, doubly stochastic, it follows that $\boldsymbol{\pi} \approx \mathbf{1}_n/n$, and this bound confines $\mathbf{p}_t$ to be near uniform. So long as $\hat{\mathbf{p}}_t$ tracks $\mathbf{p}_t$, s_t will be small as well.

In Figure 2B, where all T are doubly stochastic, we expect ℓ_t to vanish at small $c_E = |1 - 2\epsilon|$. Indeed, this mechanism explains the horizontal minimum at $\epsilon = 1/2$. Figure 2C reruns (B) for T that are not doubly stochastic, scanning over τ_1 and τ_2 at a fixed $\epsilon = .45$ ($c_E = .1$). A deep minimum sits at $\tau_1 + \tau_2 = 1$, where T is rank-one. A second, broader but shallower minimum sits at $\tau_1 = \tau_2$ where T is doubly stochastic.

To summarize, near softmax-integrability is broadly characterized by small c_T and small c_E, particularly so when T is doubly stochastic. As $c_E \to 0$, the belief state becomes bound to the neighborhood of $\boldsymbol{\pi}$ which, if near uniform, results in small s_t. As $c_T \to 1$, error is integrated over longer timescales, which can lead to large ℓ_t even when s_t is relatively small. Although softmax-integrable HMMs litter the $c_T c_E$-plane, only those with small c_T appear as broad minima over long time horizons.

4.2 Bespoke softmax-integrable HMMs

In Section 4, we assumed $\mathcal{P} = \Delta_\circ^n$, and enumerated all softmax-integrable HMMs therein. Indeed, there also exist HMMs that are softmax-integrable over more bespoke choices of $\mathcal{P} \subset \Delta_\circ^n$. These HMMs appear to implement shortcut learning, with T and E tuned such that the forward filter, restricted to admissible belief states, reduces to a simpler update rule. These HMMs are often characterized by confinement of the belief state to a lower-dimensional subset of the probability simplex.

As an example, consider

$$T = \frac{1}{10}\begin{pmatrix} 4 & 2 & 4 \\ 3 & 4 & 3 \\ 3 & 4 & 3 \end{pmatrix} \qquad\qquad E = \frac{1}{4}\begin{pmatrix} 3 & 2 & 1 \\ 1 & 2 & 3 \end{pmatrix}. \tag{17}$$

This T has rank two and is neither permutation nor rank-one, dense, so it is not softmax-integrable over $\Delta_\circ^3$. However, the point set $\mathcal{P} = \left\{ \frac{1}{6}(3,2,1)^\top, \frac{1}{6}(1,2,3)^\top \right\}$ is closed under the forward filter dynamics. For both $\mathbf{p} \in \mathcal{P}$, $T\mathbf{p} = \frac{1}{3}\mathbf{1}_3$ is uniform. As a result, the forward filter reduces to a linear map from emission to belief state, $\mathbf{p}_t = \frac{2}{3}E^\top \mathbf{u}_t$. Hence, a softmax-RNN can easily reproduce the output of the forward filter over $\mathcal{P}$. For example: $k = 1$, $F = 0$, $G = (1\ \ 0)$, $M = \log(3)(1, 0, -1)^\top$

and $\mathbf{b} = \log\left(\frac{1}{6}(1,2,3)^\top\right)$. This solution is not a member of the solution family we derived above. Rather, G, M, and $\mathbf{b}$ simply interpolate between the two elements of $\mathcal{P}$. We are unable to enumerate all such HMMs in this manuscript.

4.3 Next-token prediction

We now focus on next-token prediction, a standard objective for sequence models. The belief state, $\mathbf{p}_t$, is a sufficient statistic for all forward prediction tasks over emission sequences. The next-token posterior, $q_{t,i} = \mathbb{P}(y_{t+1} = i \mid y_{1:t})$, may be computed as $\mathbf{q}_t = ET\mathbf{p}_t$ (see Appendix A). It follows from Equation 4 that next-token statistics may be decoded from the latent state of a softmax-RNN as $\hat{\mathbf{q}}_t = ET\,\mathrm{softmax}(M\mathbf{h}_t + \mathbf{b})$.

This motivates us to investigate next-token prediction in RNNs of the form,

$$\mathbf{h}_t = F\mathbf{h}_{t-1} + G\mathbf{u}_t, \qquad \hat{\mathbf{q}}_t = H\,\mathrm{softmax}(M\mathbf{h}_t + \mathbf{b}), \tag{18}$$

where $H \in \mathbb{R}^{m\times n}$ is a learned column-stochastic matrix, and we train the network to minimize $\mathbb{E}_t[D_{\mathrm{KL}}(\mathbf{q}_t \| \hat{\mathbf{q}}_t)]$. However, it is not obvious that an RNN trained to minimize next-token prediction error will continue to represent log-beliefs in its latent state.

Empirical evidence suggests that it does. As illustrated in Figure 1, when Equation 18 is trained on the Mess3 HMM studied by Piotrowski et al. [12], its latent state represents $\log(\mathbf{p}_t)$ up to an affine transformation. Moreover, we find that H converges to ET (relative error 0.5%, see Appendix B), which is the optimal value of H when Equation 4 holds. When $H = ET$, the bounds of Section 4.1 carry over to next-token prediction, since $D_{\mathrm{KL}}(\mathbf{q}_t \| \hat{\mathbf{q}}_t) \le d(\mathbf{q}_t, \hat{\mathbf{q}}_t) \le c_E c_T\, d(\mathbf{p}_t, \hat{\mathbf{p}}_t)$ by Equation 9 and Equation 10.

Although Mess3 is neither a permutation nor rank-one, and hence not softmax-integrable, we find that the trained network lies close to the solution family derived in Section 4 (relative errors over weight matrices $< 7\%$, see Appendix B). Mess3 has $c_T \approx 0.65$ and $c_E = 0.5$, both far from unity, and a doubly stochastic transition matrix, such that its stationary state is uniform. Hence, both Mechanisms 1 and 2 are active: approximation error is rapidly forgotten, and the belief state remains near the uniform state, where the injected error s_t is small. This might explain why the solution family performs well here. That said, the solution family is not optimal: the loss evaluated on it (3.7×10^{-5}) is larger than that of the trained network (8.3×10^{-6}).

In contrast to Equation 18, most sequence models use the readout

$$\hat{\mathbf{q}}_t = \mathrm{softmax}(A\mathbf{h}_t + \mathbf{d}), \tag{19}$$

where $A \in \mathbb{R}^{m\times k}$ and $\mathbf{d} \in \mathbb{R}^m$. This is the readout we will explore empirically in Section 5, since it is standard in the broader literature. We find that, on randomly sampled HMMs, networks that use Equation 19 as their readout tend to perform about five times worse on the next-token prediction task than those that use Equation 18. We suspect this is because A must approximate the action of ET commuted through the softmax, reintroducing the approximation problem of Equation 14. However, this distinction should matter less in deep architectures, where H may be absorbed into the input weights of the following layer rather than commuted through the softmax.

5 Experiments

5.1 Randomly generated HMMs

To evaluate the generic behavior of the linear RNNs, we next consider unstructured randomly generated HMMs, as in prior work [41, 42, 43, 44].

Generating HMMs. We sampled the columns of the transition matrix, T, and the emission matrix, E, independently from Dirichlet distributions with varying concentration parameters (to control the sparseness) and ratios of observables to latent states:

$$T_{:,i} \sim \mathrm{Dir}(\beta_T \mathbf{1}_n), \quad E_{:,i} \sim \mathrm{Dir}(\beta_E \mathbf{1}_m)$$

Recent work by Agarwal et al. [41] fit transformers and deep state space models to sequences generated by HMMs that were drawn from this ensemble with $\beta_T = \beta_E = 1$.

Experimental design. We generated HMMs with $\beta_T, \beta_E \in \{0.01, 0.25, 0.5, 0.75, 1\}$, $m/n \in \{0.5, 1, 2\}$, and $n = 16, 32$ for a total of 150 unique HMMs. For each HMM, we trained five RNNs with the same number of hidden units as latent HMM states and distinct random initializations. Models were trained to predict the next token by minimizing the categorical cross-entropy of the next token (additional training details can be found in Appendix F). Figures denote average error over initializations, with full results reported in Section G.2. Our results were consistent across values of m and n, so we report $n = m = 16$ with the full set of results in Appendix G.

Results. To begin, we find that linear RNNs closely match the Bayes-optimal next-token posterior, achieving a median KL divergence of 1.55×10^{-3} across all values of β_T, β_E (Figure 3A; Table 1). However, our theoretical results apply specifically to models that track the latent belief state, so we next assessed whether the models learn to represent the belief state. We trained a linear decoder to recover the true log-belief state, $\log(\boldsymbol{\rho}_t)$, from the RNN hidden activations, $\mathbf{h}_t$. These decoders performed well, achieving an average $D_{\mathrm{KL}}(\mathbf{p}_t \| \hat{\mathbf{p}}_t) = 0.09$, where $\hat{\mathbf{p}}_t$ is the softmax of the decoder output, compared to 0.27 for decoders trained on the activations of randomly initialized linear RNNs. Moreover, next-token prediction performance is strongly correlated with posterior decoding performance across models (Pearson's $r = 0.986$). Together, these results support the applicability of our theoretical framework to the trained models.

We next examine how approximation error varies with the HMM parameters β_T and β_E. As β_T approaches zero, it becomes increasingly likely that T contains a near zero element, such that c_T approaches one. The same logic applies to β_E. As predicted by the theory, approximation error is largest as β_E or β_T approach zero. Indeed, linear RNNs trained to filter HMMs with either c_T or c_E equal to one (up to machine precision) exhibit significantly larger next-token prediction error than those trained on HMMs with both contraction coefficients strictly less than 1 (Figure 3B; Mann–Whitney U-test: $U = 98.0$, $p = 1.052 \times 10^{-18}$).

5.2 Clustered latent-state HMMs

Next, we consider a class of adversarially constructed HMMs with a clustered latent-state structure. Such structure arises in domains including activity recognition, topic dynamics, speech modeling, and navigation, where systems remain in a coherent latent mode for extended periods before transitioning [e.g., 45, 46, 47, 48]. Our theory predicts that these HMMs are particularly challenging for linear RNNs to learn because clustered latent states necessarily induce transition probabilities that are near zero.

Generating HMMs. For an HMM with K clusters, each consisting of s latent states, we define the transition matrix, T, as the convex sum of two matrices $T = \epsilon_T T_{\text{cluster}} + (1 - \epsilon_T) T_{\text{rand}}$, where $T_{\text{cluster}} = \text{blockdiag}(T_1, \ldots, T_K) \in \mathbb{R}^{n \times n}$ is a block-diagonal matrix that captures intra-cluster dynamics with each $T_i \in \mathbb{R}^{s \times s}$ sampled column-wise from a uniform Dirichlet, and $T_{\text{rand}} \in \mathbb{R}^{n \times n}$ is a random transition matrix sampled column-wise from a uniform Dirichlet. Varying $\epsilon_T \in [0, 1]$

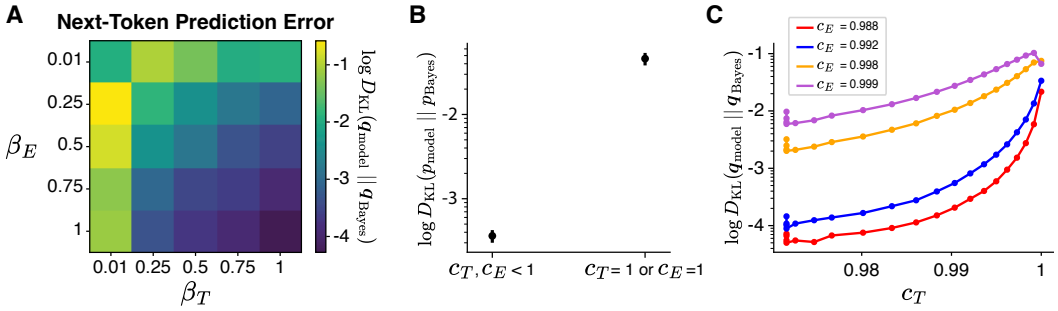


Figure 3: **Linear RNN performance is broadly consistent with theoretical predictions. A**. Mean KL-divergence between linear RNN and optimal next-token prediction increases as $\beta_E, \beta_T \to 0.01$. **B**. Mean KL-divergence between linear RNN and optimal next-token prediction is significantly higher when c_T or $c_E = 1$, as predicted by theory. Data are mean $\pm$ S.E.M. **C**. Mean KL-divergence between linear RNN and optimal next-token prediction increases as $c_T \to 1$, except in the highly observable case (pink line).

interpolates between a fully random HMM ($\epsilon_T = 0$) and a clustered HMM with no inter-clique transitions ($\epsilon_T = 1$).

We similarly varied the informativeness of the emission matrix as the convex combination of a sparse and dense matrix as $E = \epsilon_E E_{\text{sparse}} + (1 - \epsilon_E) E_{\text{dense}}$, where E_{sparse} is sampled row-wise from $\text{Dir}(0.01 \cdot \mathbf{1})$ while E_{dense} is sampled column-wise from the uniform Dirichlet distribution.

Experimental Design. We generated HMMs with the number of clusters, $K = 2, 4, 8$, and the number of states per clique, $s = 2, 4, 8$ and fixed $m = n = sK$. The results were consistent across pairs of (K, s) so we report $(K, s) = (4, 4)$ here with the full set of HMMs in Appendix G. For each pair (K, s), we fixed $T_{\text{cluster}}, T_{\text{rand}}, E_{\text{sparse}}, E_{\text{dense}}$ and generated transition and emission matrices using 20 evenly spaced ϵ_T values between 0 and 1 and $\epsilon_E \in \{0, 0.25, 0.75, 1\}$ for a total of 80 HMMs per pair. $c_T, c_E \to 1$ as $\epsilon_T, \epsilon_E \to 1$, but the full range of c_T, c_E depended on the initial component matrices and is not necessarily a monotonic function of ϵ.

Results. As predicted by theory, performance generally degrades as $c_T, c_E \to 1$ (Figure 3C, 6). However, in contrast with the random HMM case, the linear RNN performance actually improves with complete cluster separation ($\epsilon_T = \epsilon_E = 1$). In this case, each sequence contains observations from only a single cluster, effectively reducing the problem to filtering within a smaller, densely connected HMM, and highly informative emissions enable rapid identification of the active cluster from few observations. We therefore hypothesize a two-stage inference process in which the model first identifies the cluster generating the sequence and then performs filtering within that densely connected cluster. Consistent with this hypothesis, next-token prediction error in the HMM with fully disconnected clusters ($\epsilon_T = 1$) is initially high but decreases with additional observations. In contrast, prediction error remains approximately constant across sequence positions when clusters are more strongly connected (Figure 7).

These findings suggest that, for certain structured HMMs, linear systems may be able to exploit problem-specific structure to partially overcome the approximation limitations implied by our bounds. Characterizing when such advantages arise and the computations the models perform remains an important direction for future work.

6 Discussion

Our results show that, for a broad class of HMMs, linear models can approximate Bayes-optimal filtering, when provided a softmax readout. This finding is surprising in light of recent work showing that substantially more complex nonlinear architectures learn strong representations on similar HMM-based prediction tasks [41, 42] and suggests that some benchmarks used to study emergent Bayesian inference may already be largely solvable by linear dynamics. Our work highlights an important caveat for model and task design: for certain problems, tracking log-beliefs is approximately linear, so gains from more expressive architectures may be limited unless the task fundamentally requires nonlinear inference.

Moreover, these findings may help clarify the representational capacity of deep state space models and linear recurrent units (LRUs), which evolve linearly over time yet remain competitive with state-of-the-art architectures with the inclusion of static nonlinearities [6, 8, 7]. Our work provides a principled step toward understanding the capabilities and limitations of linear recurrent networks. A deeper understanding of the computational capacity of linear RNNs may help characterize how linear layers can facilitate meaningful computations in deep networks performing more complex tasks.

Limitations. Our theory assumes the softmax-RNN to track the forward filter from every strictly positive belief state, including counterfactual ones, and for every emission (see Remark 2). When these assumptions are relaxed, there exist other bespoke HMMs that can be filtered exactly over restricted domains (c.f. Section 4.2). We were unable to enumerate all such HMMs in this manuscript. These cases nonetheless offer interesting examples of shortcut learning.

Our results focus entirely on the softmax readout and say little about other readouts. Moreover, we consider only single-layer architectures, while hierarchically structured HMMs, such as the clustered HMMs of Section 5, may be better served by deeper models. Both of these limitations offer interesting avenues for future work.

7 Acknowledgements

This work was supported by the Simons Foundation. AHW was supported by a McKnight Scholars Award.

References

[1] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[5] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[6] Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

[7] Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR, 2023.

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

[9] Karl Johan Åström. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, February 1965. ISSN 0022-247X. doi: 10.1016/0022-247x(65)90154-x. URL http://dx.doi.org/10.1016/0022-247X(65)90154-X.

[10] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Stat.*, 37(6):1554–1563, December 1966.

[11] Adam Shai, Lucas Teixeira, Alexander Oldenziel, Sarah Marzen, and Paul Riechers. Transformers represent belief state geometry in their residual stream. *Advances in Neural Information Processing Systems*, 37:75012–75034, 2024.

[12] Mateusz Piotrowski, Paul M Riechers, Daniel Filan, and Adam S Shai. Constrained belief updates explain geometric structures in transformer representations. *arXiv preprint arXiv:2502.01954*, 2025.

[13] Keenan Pepper. Rnns represent belief state geometry in their hidden states, 2024. Submitted to Computational Mechanics Hackathon, an Apart Research Sprint.

[14] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

[15] Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 15614–15638. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3255a7554605a88800f4e120b3a929e1-Paper-Conference.pdf.

[16] Hui Jiang. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

[17] Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Are Large Language Models Bayesian? A Martingale Perspective on In-Context Learning. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL https://openreview.net/forum?id=z4YTZO1NT4.

[18] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[19] Antonio Orvieto, Soham De, Caglar Gulcehre, Razvan Pascanu, and Samuel L Smith. Universality of linear recurrences followed by non-linear projections: finite-width guarantees and benefits of complex eigenvalues. *arXiv preprint arXiv:2307.11888*, 2023.

[20] Yedi Zhang, Andrew M Saxe, and Peter E. Latham. Saddle-to-saddle dynamics explains a simplicity bias across neural network architectures. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Vit5M0G5Gb.

[21] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[22] Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

[23] Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, Matus Telgarsky, et al. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15(1):2773–2832, 2014.

[24] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960. ISSN 0021-9223. doi: 10.1115/1.3662552. URL http://dx.doi.org/10.1115/1.3662552.

[25] Igor Mezić. On numerical approximations of the Koopman operator. *Mathematics*, 10(7):1180, 2022.

[26] Christophe Zhang and Enrique Zuazua. A quantitative analysis of Koopman operator methods for system identification and predictions. *Comptes Rendus. Mécanique*, 351(S1):1–31, 2023.

[27] Frederik Köhne, Friedrich M Philipp, Manuel Schaller, Anton Schiela, and Karl Worthmann. L^∞-error bounds for approximations of the koopman operator by kernel extended dynamic mode decomposition. *SIAM journal on applied dynamical systems*, 24(1):501–529, 2025.

[28] Friedrich M Philipp, Manuel Schaller, Karl Worthmann, Sebastian Peitz, and Feliks Nüske. Error analysis of kernel EDMD for prediction and control in the Koopman framework. *Journal of Nonlinear Science*, 35(5):92, 2025.

[29] Friedrich Philipp, Manuel Schaller, Karl Worthmann, Sebastian Peitz, and Feliks Nueske. Error bounds for kernel-based approximations of the Koopman operator. *arXiv preprint arXiv:2301.08637*, 2023.

[30] Rajesh PN Rao. Bayesian computation in recurrent neural circuits. *Neural computation*, 16(1): 1–38, 2004.

[31] Edward S Hu, Kwangjun Ahn, Qinghua Liu, Haoran Xu, Manan Tomar, Ada Langford, Jayden Teoh, Bryon Xu, David Yan, Dinesh Jayaraman, et al. The belief state transformer. *arXiv preprint arXiv:2410.23506*, 2024.

[32] Pedro A Ortega, Jane X Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, et al. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

[33] Gaspard Lambrechts, Adrien Bolland, and Damien Ernst. Recurrent networks, hidden states and beliefs in partially observable environments. *arXiv preprint arXiv:2208.03520*, 2022.

[34] Jay A Hennig, Sandra A Romero Pinto, Takahiro Yamaguchi, Scott W Linderman, Naoshige Uchida, and Samuel J Gershman. Emergence of belief-like representations through reinforcement learning. *PLOS Computational Biology*, 19(9):e1011067, 2023.

[35] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

[36] A Emin Orhan and Wei Ji Ma. Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback. *Nature communications*, 8(1):138, 2017.

[37] Charlotte Striebel. Sufficient statistics in the optimum control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12(3):576–592, December 1965. ISSN 0022-247X. doi: 10.1016/0022-247x(65)90027-2. URL `http://dx.doi.org/10.1016/0022-247X(65)90027-2`.

[38] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1558601244.

[39] Garrett Birkhoff. Extensions of jentzsch's theorem. *Transactions of the American Mathematical Society*, 85(1):219–227, 1957.

[40] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer New York, 1981. ISBN 9780387327921. doi: 10.1007/0-387-32792-4. URL `http://dx.doi.org/10.1007/0-387-32792-4`.

[41] Naman Agarwal, Siddhartha R Dalal, and Vishal Misra. The Bayesian geometry of transformer attention. *arXiv preprint arXiv:2512.22471*, 2025.

[42] Reginald Zhiyan Chen, Heng-Sheng Chang, and Prashant G Mehta. Belief Net: A filter-based framework for learning hidden Markov models from observations. *arXiv preprint arXiv:2511.10571*, 2025.

[43] Jiachen Hu, Qinghua Liu, and Chi Jin. On limitation of transformer for learning HMMs. *arXiv preprint arXiv:2406.04089*, 2024.

[44] Maoying Qiao, Wei Bian, Richard Yi Da Xu, and Dacheng Tao. Diversified hidden Markov models for sequential labeling. *IEEE Transactions on Knowledge and Data Engineering*, 27 (11):2947–2960, 2015.

[45] Hung H Bui, Dinh Q Phung, and Svetha Venkatesh. Hierarchical hidden Markov models with general state hierarchy. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 324–329. AAAI Press, 2004.

[46] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. An hdp-hmm for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pages 312–319, 2008.

[47] Thomas Stepleton, Zoubin Ghahramani, Geoffrey Gordon, and Tai-Sing Lee. The block diagonal infinite hidden Markov model. In *Artificial intelligence and statistics*, pages 552–559. PMLR, 2009.

[48] Luca Mazzucato, Alfredo Fontanini, and Giancarlo La Camera. Dynamics of multistable states during ongoing and evoked cortical activity. *Journal of Neuroscience*, 35(21):8214–8231, 2015.

[49] Sarah E. Marzen and James P. Crutchfield. Nearly maximally predictive features and their dimensions. *Physical Review E*, 95(5), May 2017. ISSN 2470-0053. doi: 10.1103/physreve.95.051301. URL http://dx.doi.org/10.1103/PhysRevE.95.051301.

[50] J. Aczél. *Lectures on Functional Equations and Their Applications*, volume 19 of *Mathematics in Science and Engineering*. Academic Press, New York, NY, 1966. Translated from the German edition (Birkhäuser Verlag, 1961).

[51] F. Radó and John A. Baker. Pexider's equation and aggregation of allocations. *Aequationes Mathematicae*, 32(1):227–239, December 1987. ISSN 1420-8903. doi: 10.1007/bf02311311. URL http://dx.doi.org/10.1007/BF02311311.

[52] Rolando Cavazos-Cadena. An alternative derivation of birkhoff's formula for the contraction coefficient of a positive matrix. *Linear Algebra and its Applications*, 375:291–297, December 2003. ISSN 0024-3795. doi: 10.1016/j.laa.2003.05.002. URL http://dx.doi.org/10.1016/j.laa.2003.05.002.

[53] Tiberiu Popoviciu. Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, 9:129–145, 1935.

[54] Patrick Kidger and Cristian Garcia. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021.

A Deriving the forward filter and next token prediction

To derive the forward filter [38], consider the joint probability $\rho_{t,i} = \mathbb{P}(x_t = i, y_{1:t})$ for $i = 1, \ldots, n$, which satisfies a recurrence relation

$$\mathbb{P}(x_t, y_{1:t}) = \sum_{x_{t-1}} \mathbb{P}(x_t, x_{t-1}, y_{1:t}) \tag{20}$$

$$= \sum_{x_{t-1}} \mathbb{P}(y_t \mid x_t, x_{t-1}, y_{1:t-1}) \mathbb{P}(x_t \mid x_{t-1}, y_{1:t-1}) \mathbb{P}(x_{t-1}, y_{1:t-1}) \tag{21}$$

$$= \sum_{x_{t-1}} \mathbb{P}(y_t \mid x_t) \mathbb{P}(x_t \mid x_{t-1}) \mathbb{P}(x_{t-1}, y_{1:t-1}), \tag{22}$$

where in the third line we have used that x_t is a sufficient statistic for y_t, such that $\mathbb{P}(y_t \mid x_t, x_{t-1}, y_{1:t-1}) = \mathbb{P}(y_t \mid x_t)$, and x_{t-1} is a sufficient statistic for x_t, such that $\mathbb{P}(x_t \mid x_{t-1}, y_{1:t-1}) = \mathbb{P}(x_t \mid x_{t-1})$.

Identifying $\mathbb{P}(y_t \mid x_t) = E_{y_t, x_t}$ and $\mathbb{P}(x_t \mid x_{t-1}) = T_{x_t, x_{t-1}}$, this reads,

$$\rho_{t,i} = E_{y_t i} \sum_j T_{ij} \rho_{t-1,j}. \tag{23}$$

Noting that $E_{y_t i} = [E^\top \mathbf{u}_t]_i$ and $\sum_j T_{ij} \rho_{t-1,j} = (T \boldsymbol{\rho}_{t-1})_i$, we may write this in vector form as

$$\boldsymbol{\rho}_t = (E^\top \mathbf{u}_t) \odot (T \boldsymbol{\rho}_{t-1}).$$

The belief state is recovered as

$$\mathbb{P}(x_t = i \mid y_{1:t}) = \frac{\mathbb{P}(x_t = i, y_{1:t})}{\sum_j \mathbb{P}(x_t = j, y_{1:t})} = \frac{\rho_{t,i}}{\sum_j \rho_{t,j}}, \tag{24}$$

or in vector form,

$$\mathbf{p}_t = \frac{\boldsymbol{\rho}_t}{\mathbf{1}^\top \boldsymbol{\rho}_t}.$$

Together, these give Equation 1.

A.1 Next token prediction

The next token posterior can be computed from the belief state as

$$\mathbb{P}(y_{t+1} | y_{1:t}) = \sum_{x_{t+1}} \mathbb{P}(y_{t+1}, x_{t+1} | y_{1:t}) \tag{25}$$

$$= \sum_{x_{t+1}} \mathbb{P}(y_{t+1} | x_{t+1}) \mathbb{P}(x_{t+1} | y_{1:t}) \tag{26}$$

$$= \sum_{x_{t+1}, x_t} \mathbb{P}(y_{t+1} \mid x_{t+1}) \mathbb{P}(x_{t+1} \mid x_t) \mathbb{P}(x_t | y_{1:t}), \tag{27}$$

where we have used the same sufficiency arguments as above. Noting that $q_{t,i} = \mathbb{P}(y_{t+1} = i | y_{1:t})$, it follows that $q_{t,i} = \sum_{jk} E_{ij} T_{jk} p_{t,k}$, or in vector form

$$\mathbf{q}_t = ET\mathbf{p}_t. \tag{28}$$

A.2 Alternative forms

The forward filter can be written directly as a function of $\mathbf{p}_t$ as

$$\mathbf{p}_t = \frac{(E^\top \mathbf{u}_t) \odot (T \mathbf{p}_{t-1})}{\mathbf{u}_t^\top ET \mathbf{p}_{t-1}}. \tag{29}$$

This follows directly from combining Equation 1. The denominator is the probability of the observed symbol, $\mathbf{u}_t^\top ET\mathbf{p}_{t-1} = \mathbb{P}(y_t \mid y_{1:t-1})$, which is nonzero over every symbol sequence the HMM emits.

For one-hot inputs, Equation 29 can be written in a Kalman-like form. Let

$$[\mathrm{diag}(E\mathbf{p})^\dagger]_{ii} = \begin{cases} [E\mathbf{p}]_i^{-1} & [E\mathbf{p}]_i > 0 \\ 0 & \text{else,} \end{cases} \tag{30}$$

denote the Moore–Penrose pseudoinverse of $\mathrm{diag}(E\mathbf{p})$, which inverts $\mathrm{diag}(E\mathbf{p})$ on its range and vanishes elsewhere. Then, we may define the gain matrix

$$\Gamma(\mathbf{p}) = \mathrm{diag}(\mathbf{p})E^\top \mathrm{diag}(E\mathbf{p})^\dagger, \tag{31}$$

which satisfies two properties. First, since $\mathbf{u}_t$ is one-hot at a symbol of nonzero probability, $\mathrm{diag}(ET\mathbf{p}_{t-1})^\dagger \mathbf{u}_t = \mathbf{u}_t/(\mathbf{u}_t^\top ET\mathbf{p}_{t-1})$, and

$$\Gamma(T\mathbf{p}_{t-1})\mathbf{u}_t = \mathrm{diag}(T\mathbf{p}_{t-1})E^\top \mathrm{diag}(ET\mathbf{p}_{t-1})^\dagger \mathbf{u}_t = \frac{(E^\top \mathbf{u}_t) \odot (T\mathbf{p}_{t-1})}{\mathbf{u}_t^\top ET\mathbf{p}_{t-1}} = \mathbf{p}_t. \tag{32}$$

For any symbol of zero probability, $\mathbf{u}^\top ET\mathbf{p}_{t-1} = 0$, this quantity is zero, $\Gamma(T\mathbf{p}_{t-1})\mathbf{u} = \mathbf{0}_n$. Second, for every probability vector $\mathbf{p}$, whether or not $E\mathbf{p}$ is strictly positive,

$$\Gamma(\mathbf{p})E\mathbf{p} = \mathrm{diag}(\mathbf{p})E^\top \mathrm{diag}(E\mathbf{p})^\dagger E\mathbf{p} = \mathrm{diag}(\mathbf{p})E^\top \mathbf{s} = \mathbf{p}, \tag{33}$$

where $\mathbf{s} = \mathrm{diag}(E\mathbf{p})^\dagger E\mathbf{p}$ indicates the support of $E\mathbf{p}$, with $s_i = 1$ where $[E\mathbf{p}]_i > 0$ and $s_i = 0$ otherwise. The final equality holds because $p_i \sum_j E_{ji}s_j = p_i$ has two options. Either $p_i = 0$, such that the equation holds trivially, or $p_i > 0$, such that $\sum_j E_{ji}s_j = 1$. The latter follows because any j dropped by $\mathbf{s}$ has $[E\mathbf{p}]_j = \sum_k E_{jk}p_k = 0$, forcing $E_{ji} = 0$ whenever $p_i > 0$. The dropped terms therefore contribute nothing, and $\sum_j E_{ji}s_j = \sum_j E_{ji} = 1$ by the column stochasticity of E.

Taken together, Equation 32 and Equation 33 allow one to write

$$\begin{aligned} \mathbf{p}_t &= \Gamma(T\mathbf{p}_{t-1})\mathbf{u}_t && (34)\\ &= T\mathbf{p}_{t-1} + \Gamma(T\mathbf{p}_{t-1})\mathbf{u}_t - \Gamma(T\mathbf{p}_{t-1})ET\mathbf{p}_{t-1} && (35)\\ &= T\mathbf{p}_{t-1} + \Gamma(T\mathbf{p}_{t-1})\big(\mathbf{u}_t - ET\mathbf{p}_{t-1}\big), && (36) \end{aligned}$$

where in the second line we have added zero $\mathbf{0}_n = T\mathbf{p}_{t-1} - \Gamma(T\mathbf{p}_{t-1})ET\mathbf{p}_{t-1}$, via Equation 33. Above, $\mathbf{u}_t - ET\mathbf{p}_{t-1}$ represents an innovation-like term: the error between the predicted next token distribution and the observed measurement. The matrix Γ plays the role of a state-dependent, Kalman-like gain.

A.3 Perturbations and statistics

For an HMM with E that does not satisfy Assumption 3, one can take $E_\epsilon = (1-\epsilon)E + (\epsilon/m)\mathbf{1}_m\mathbf{1}_n^\top$, for an arbitrarily small $1 > \epsilon > 0$. E_ϵ can be understood as a noisy observation channel. Relative to E, E_ϵ will have vanishing impact on the HMM statistics in the limit $\epsilon \to 0$.

To see this, notice that a symbol sequence is left uncorrupted with probability $(1-\epsilon)^t$. So $\mathbb{P}_\epsilon(y_{1:t}) = (1-\epsilon)^t\mathbb{P}(y_{1:t}) + C$, where $(1-\epsilon)^t\mathbb{P}(y_{1:t})$ is the probability that $y_{1:t}$ gets emitted and observed without corruption, and $C > 0$ accounts for the summed probability of some other sequence $y'_{1:t}$ being emitted, but then corrupted such that $y_{1:t}$ is observed. Hence, $\mathbb{P}_\epsilon(y_{1:t}) \geq (1-\epsilon)^t\mathbb{P}(y_{1:t})$. It follows that the Kullback–Leibler (KL) divergence is bounded by

$$\begin{aligned} D_{\mathrm{KL}}\big(\mathbb{P}(y_{1:t})\|\mathbb{P}_\epsilon(y_{1:t})\big) &= \sum_{y_{1:t}} \mathbb{P}(y_{1:t}) \log\left(\frac{\mathbb{P}(y_{1:t})}{\mathbb{P}_\epsilon(y_{1:t})}\right) && (37)\\ &\leq \sum_{y_{1:t}} \mathbb{P}(y_{1:t}) \log\left(\frac{\mathbb{P}(y_{1:t})}{(1-\epsilon)^t\mathbb{P}(y_{1:t})}\right) && (38)\\ &= t\log\frac{1}{1-\epsilon}, && (39) \end{aligned}$$

which tends to zero with ϵ for all finite time-horizons, t.

B Mess3 HMM

Piotrowski et al. [12] explore the three-state "Mess3" edge-emitting HMM. Let us briefly show that the Mess3 edge-emitting HMM is representable as a three-state, node-emitting HMM.

Mess3 HMMs [49] are a family of HMMs parametrized by their transition rate, $\nu \in [0, 1/2]$, and observability, $\alpha \in [0, 1]$. Piotrowski et al. [12] define the Mess3 process via the joint transition matrices $\tilde{T}_{ijk} = \mathbb{P}(y_t = i, x_t = j | x_{t-1} = k)$:

$$\tilde{T}_{1::} = \begin{pmatrix} \alpha\mu & \alpha\nu & \alpha\nu \\ \beta\nu & \beta\mu & \beta\nu \\ \beta\nu & \beta\nu & \beta\mu \end{pmatrix} \qquad \tilde{T}_{2::} = \begin{pmatrix} \beta\mu & \beta\nu & \beta\nu \\ \alpha\nu & \alpha\mu & \alpha\nu \\ \beta\nu & \beta\nu & \beta\mu \end{pmatrix} \qquad \tilde{T}_{3::} = \begin{pmatrix} \beta\mu & \beta\nu & \beta\nu \\ \beta\nu & \beta\mu & \beta\nu \\ \alpha\nu & \alpha\nu & \alpha\mu \end{pmatrix}, \tag{40}$$

where $\mu = 1 - 2\nu$ and $\alpha = 1 - 2\beta$. Note that we write these matrices as acting from the left, while [12] defines them as acting from the right (transposed). One can compute the latent transition matrix by marginalizing over y_t,

$$T = \sum_i \tilde{T}_{i::} = \begin{pmatrix} \mu & \nu & \nu \\ \nu & \mu & \nu \\ \nu & \nu & \mu \end{pmatrix}. \tag{41}$$

The edge-emission probabilities can be computed as

$$\tilde{E}_{ijk} = \mathbb{P}(y_t = i | x_t = j, x_{t-1} = k) = \frac{\mathbb{P}(y_t = i, x_t = j | x_{t-1} = k)}{\mathbb{P}(x_t = j | x_{t-1} = k)} = \frac{\tilde{T}_{ijk}}{T_{jk}}.$$

This gives

$$\tilde{E}_{1::} = \begin{pmatrix} \alpha & \alpha & \alpha \\ \beta & \beta & \beta \\ \beta & \beta & \beta \end{pmatrix} \qquad \tilde{E}_{2::} = \begin{pmatrix} \beta & \beta & \beta \\ \alpha & \alpha & \alpha \\ \beta & \beta & \beta \end{pmatrix} \qquad \tilde{E}_{3::} = \begin{pmatrix} \beta & \beta & \beta \\ \beta & \beta & \beta \\ \alpha & \alpha & \alpha \end{pmatrix}, \tag{42}$$

which illustrates that the edge-emission probabilities $\tilde{E}_{ijk}$ are independent of k (i.e. emissions are independent of x_{t-1}). Hence, $\tilde{E}_{ijk} = E_{ij}$, where

$$E = \begin{pmatrix} \alpha & \beta & \beta \\ \beta & \alpha & \beta \\ \beta & \beta & \alpha \end{pmatrix}. \tag{43}$$

To summarize, the edge-emitting formulation of the Mess3 HMM is reducible to a node emitting HMM with transition matrix given by Equation 41 and emission matrix given by Equation 43. As $\alpha \to 1$, emissions fully determine the latent state, making it more "observable". As $\nu \to 0$, transitions between latent states occur less frequently, making the latent dynamics more "sticky".

The Birkhoff contraction coefficients of these matrices are $c_T = |\mu - \nu|/(\mu + \nu)$ and $c_E = |\alpha - \beta|/(\alpha + \beta)$, respectively. For the values $\nu = 0.15$ and $\alpha = 0.6$ used by Piotrowski et al. [12], it follows that $c_T = 11/17 \approx .65$ and $c_E = 1/2$. Our theory predicts this to be a more easily approximable HMM, since c_T and c_E are well-separated from 1.

Figure 1 is generated as follows. We construct an RNN with linear latent dynamics and softmax readout:

$$\mathbf{h}_t = F\mathbf{h}_{t-1} + G\mathbf{u}_t \qquad \hat{\mathbf{q}}_t = H \operatorname{softmax}(M\mathbf{h}_t + \mathbf{b}), \tag{44}$$

where $\mathbf{h}_t \in \mathbb{R}^3$ and $\hat{\mathbf{q}}_t \in \mathbb{R}^3_+$. F is constrained to have a spectral radius less than or equal to unity and H to be column stochastic. The network is trained to minimize the next token prediction error,

$$(F_*, G_*, H_*, M_*, \mathbf{b}_*) = \operatorname{argmin}_{F,G,H,M,\mathbf{b}} \mathbb{E}_t[D_{\mathrm{KL}}(\mathbf{q}_t \| \hat{\mathbf{q}}_t)],$$

where $\mathbf{q}_t = ET\mathbf{p}_t$ and $\mathbf{p}_t$ is generated by the forward filter with

$$T = \begin{pmatrix} .70 & .15 & .15 \\ .15 & .70 & .15 \\ .15 & .15 & .70 \end{pmatrix} \qquad E = \begin{pmatrix} .60 & .20 & .20 \\ .20 & .60 & .20 \\ .20 & .20 & .60 \end{pmatrix}. \tag{45}$$

The trained RNN achieves a loss of 8.3×10^{-6}. The learned network parameters of the trained RNN are

$$F_* \approx \begin{pmatrix} 0.7036 & -0.0595 & -0.4989 \\ 0.1900 & 0.4576 & -0.4749 \\ 0.2198 & -0.0580 & 0.0026 \end{pmatrix} \qquad G_* \approx \begin{pmatrix} -0.0689 & 0.1677 & -0.1696 \\ 0.4139 & -0.4166 & -0.1657 \\ -0.1256 & -0.5978 & 0.6211 \end{pmatrix} \tag{46}$$

$$H_* \approx \begin{pmatrix} 0.2587 & 0.4826 & 0.2586 \\ 0.4823 & 0.2587 & 0.2590 \\ 0.2590 & 0.2587 & 0.4824 \end{pmatrix} \qquad M_* \approx \begin{pmatrix} 1.4750 & -0.4297 & -0.3399 \\ -0.4736 & 1.6642 & -0.4237 \\ -0.0671 & -0.3197 & 0.9807 \end{pmatrix} \tag{47}$$

$$\mathbf{b}_* \approx \begin{pmatrix} -0.0531 \\ 0.0915 \\ -0.0181 \end{pmatrix}. \tag{48}$$

Notice that the loss of any optimal tuple $(F_*, G_*, H_*, M_*, \mathbf{b}_*)$ is invariant under

$$F_* \mapsto SF_*S^{-1} \qquad G_* \mapsto SG_* \qquad H_* \mapsto H_*P^{-1} \qquad M_* \mapsto PM_*S^{-1} \qquad \mathbf{b}_* \mapsto P\mathbf{b}_* \tag{49}$$

for any invertible matrix S and permutation matrix P. Our trained network converges to a permuted member of this family: its hidden state encodes the log-belief state in a permuted ordering of the latent states,

$$\log(\hat{\boldsymbol{\rho}}_t) = P(M_*\mathbf{h}_t + \mathbf{b}_*), \qquad P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{50}$$

We undo this permutation, and then remove the remaining freedom by taking S to be the decoder matrix of the unpermuted tuple, PM_*, which is invertible. These "canonical" coordinates ensure the relative errors we report below are invariant to these representational freedoms. The canonical parameters learned by the RNN are

$$F_\star \approx \begin{pmatrix} 0.4319 & 0.0476 & -0.3630 \\ -0.0925 & 0.5500 & -0.3925 \\ -0.0696 & 0.0589 & 0.1819 \end{pmatrix} \qquad G_\star \approx \begin{pmatrix} 0.7747 & -0.5194 & -0.4585 \\ -0.2368 & 0.6295 & -0.3901 \\ -0.2509 & -0.4643 & 0.6735 \end{pmatrix} \tag{51}$$

$$H_\star \approx \begin{pmatrix} 0.4826 & 0.2587 & 0.2586 \\ 0.2587 & 0.4823 & 0.2590 \\ 0.2587 & 0.2590 & 0.4824 \end{pmatrix} \qquad \mathbf{b}_\star \approx \begin{pmatrix} 0.0915 \\ -0.0531 \\ -0.0181 \end{pmatrix}, \tag{52}$$

with $M_\star = I_n$.

Figure 1(left) plots $\log(\mathbf{p}_t) \in \mathbb{R}^3$, in the two-dimensional plane orthogonal to $\mathbf{1}$, since this direction encodes the normalization of $\mathbf{p}_t$. In this projection, $\log(\mathbf{p}_t) = \log(\boldsymbol{\rho}_t)$, representing the ground truth belief state. Figure 1(center) plots the decoded belief state from Equation 50. Finally, Figure 1(right) reproduces Figure 1(d) from Piotrowski et al. [12].

For any near equality $A_\star \approx A$, we define the relative error as

$$\gamma(A_\star) = \frac{\|A - A_\star\|_{\mathrm{Fro}}}{\|A\|_{\mathrm{Fro}}}, \tag{53}$$

where $\|\cdot\|_{\mathrm{Fro}}$ is the Frobenius norm, $A_\star$ is a canonical parameter of the RNN and A is the value of its analog in the solution family introduced in Appendix C. The relative error of $H_\star \approx ET$ is $\gamma(H_\star) = 0.005$, in congruence with Equation 28. Interestingly, despite not being softmax-integrable, the Mess3 HMM studied here appears to be optimally filtered by RNNs that lie quite close to the solution family described in Appendix C. In particular, the trained network approaches the member

$$F_\star \approx T + \mathbf{1}_n\mathbf{k}_1^\top \qquad G_\star \approx \log\left(E^\top\right) + \left(A_\star - I_n\right)\mathbf{b}_\star\mathbf{1}_m^\top + \mathbf{1}_n\mathbf{k}_2^\top, \tag{54}$$

at

$$\mathbf{k}_1 \approx \begin{pmatrix} -0.2434 \\ -0.1145 \\ -0.5245 \end{pmatrix} \qquad \mathbf{k}_2 \approx \begin{pmatrix} 1.3464 \\ 1.1327 \\ 1.1923 \end{pmatrix}, \tag{55}$$

with $\gamma(F_\star) = 0.064$ and $\gamma(G_\star) = 0.018$. These have the form of Equation 114 and Equation 112. Since $M_\star = I_n$, we take $M_\star^\dagger = I_n - \mathbf{1}_n\mathbf{1}_n^\top/n$, which satisfies Equation 59, and $A_\star = M_\star F_\star M_\star^\dagger$.

Note that $M_\star^\dagger M_\star \neq I_n$, so the kernel terms do not vanish: $\left(I_n - M_\star^\dagger M_\star\right) K = \mathbf{1}_n \mathbf{1}_n^\top K/n$ is exactly what supplies $\mathbf{1}_n \mathbf{k}_1^\top$ and $\mathbf{1}_n \mathbf{k}_2^\top$. The $\log(T\mathbf{1}_n)$ term vanishes since T is doubly stochastic. We find that trained RNNs remain close to the solution family over a range of α and ν. However, the solution family is not optimal and the trained RNNs do deviate from it in order to minimize the loss function. An RNN of the form Equation 44 evaluated on the solution family ($F = T$, $G = \log(E^\top)$, $H = ET$, $M = I_n$ and $\mathbf{b} = \mathbf{0}_n$) achieves a loss of 3.7×10^{-5}, about four and a half times that of the trained network.

C Softmax-integrable HMMs

Throughout this section, we compare two dynamical systems:

$$\boldsymbol{\rho}_t = \left(E^\top \mathbf{u}_t\right) \odot \left(T \boldsymbol{\rho}_{t-1}\right) \qquad\qquad \mathbf{p}_t = \frac{\boldsymbol{\rho}_t}{\mathbf{1}_n^\top \boldsymbol{\rho}_t} \tag{56}$$

and

$$\mathbf{h}_t = F\mathbf{h}_{t-1} + G\mathbf{u}_t \qquad\qquad \hat{\mathbf{p}}_t = \mathrm{softmax}(M\mathbf{h}_t + \mathbf{b}) := \frac{\hat{\boldsymbol{\rho}}_t}{\mathbf{1}_n^\top \hat{\boldsymbol{\rho}}_t} \tag{57}$$

The first represents the forward filter derived in Appendix A. In it, $\boldsymbol{\rho}_t \in \mathbb{R}_+^n$ is the unnormalized belief state, $\mathbf{p}_t$ is the belief state, and $E \in \mathbb{R}^{m \times n}, T \in \mathbb{R}^{n \times n}$ are column-stochastic matrices. The input, $\mathbf{u}_t \in \mathcal{U} \subseteq \mathbb{R}^m$, is a one-hot encoding of the observation y_t, $u_{t,i} = \delta_{i y_t}$. Where a proof requires this property, we invoke Assumption 4 explicitly.

The latter equation describes an RNN, where $\mathbf{h}_t \in \mathbb{R}^k$ is the latent state of the network, $\mathbf{u}_t \in \mathcal{U}$ is the same input as above, and $F \in \mathbb{R}^{k \times k}, G \in \mathbb{R}^{k \times m}$ are matrices. The readout of the network is an affine transformation, parameterized by $M \in \mathbb{R}^{n \times k}$ and $\mathbf{b} \in \mathbb{R}^n$, followed by a softmax. We have defined $\hat{\boldsymbol{\rho}}_t := \exp\left(M\mathbf{h}_t + \mathbf{b}\right)$ as the estimated unnormalized belief state. Since M and $\mathbf{b}$ are finite, only strictly positive unnormalized belief states are realizable by finite hidden states, $\mathbf{h}$; unnormalized belief states with zero components can be approximated arbitrarily well.

Note that the readout of the RNN estimates the normalized belief state, not next token probabilities. This section is not yet concerned with next token prediction. Of course, they are related tasks, as next token probabilities are a linear function of the belief state (Equation 28).

Throughout, $\mathbf{1}_d \in \mathbb{R}^d$ denotes the vector of ones in d-dimensions, $\mathbf{0}_d \in \mathbb{R}^d$ the zero vector, and I_d the $d \times d$ identity matrix. We write $\mathbb{R}_+^d$ for the non-negative orthant and $\mathbb{R}_{++}^d$ for the strictly positive orthant. We will also use $\mathbf{v}_d \in \mathbb{R}^d$ to denote an arbitrary vector in d-dimensions. These arbitrary vectors parameterize families of solutions and may be chosen freely. We will denote the entire probability simplex in $\mathbb{R}^d$ as $\Delta^d = \{\, \mathbf{p} \in \mathbb{R}^d : \mathbf{p}^\top \mathbf{1}_d = 1, p_i \geq 0 \,\}$ and its interior, the set of strictly positive vectors, as $\Delta_\circ^d = \{\, \mathbf{p} \in \mathbb{R}^d : \mathbf{p}^\top \mathbf{1}_d = 1, p_i > 0 \,\}$.

Since $\mathbf{p}_t$ is invariant to the magnitude of $\boldsymbol{\rho}_t$, $\mathbf{p}_t = \hat{\mathbf{p}}_t$ implies only that $\boldsymbol{\rho}_t \propto \hat{\boldsymbol{\rho}}_t$ are proportional. Proportionality of beliefs implies that log beliefs are offset by a constant vector:

$$\boldsymbol{\rho} = z\hat{\boldsymbol{\rho}}, \qquad z \in \mathbb{R}_{++} \qquad\Longrightarrow\qquad \log(\boldsymbol{\rho}) = \log(\hat{\boldsymbol{\rho}}) + c\mathbf{1}_n, \qquad c \in \mathbb{R}. \tag{58}$$

Below, we write $\mathbf{x} \simeq \mathbf{y}$, for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, when two vectors are equal up to an offset along the constant vector: $\mathbf{y} = \mathbf{x} + z\mathbf{1}_d$, for some $z \in \mathbb{R}$. We write $\mathbb{R}^d/\mathbf{1}_d$ for the set of vectors in $\mathbb{R}^d$ taken modulo this offset.

We seek to understand when an HMM is softmax-integrable over $\mathcal{P} = \Delta_\circ^n$. Given an HMM with parameters (T, E), we seek a choice of $(k, M, \mathbf{b}, F, G)$ such that Definition 1 holds with $\boldsymbol{\phi} = \mathrm{softmax}$. We will cite the assumptions listed in Section 4 as they are used.

Definition 2. It will be convenient to invert the readout. We call any $M^\dagger \in \mathbb{R}^{k \times n}$ satisfying the following a right inverse of M:

$$MM^\dagger \mathbf{x} \simeq \mathbf{x}, \qquad \forall \mathbf{x} \in \mathbb{R}^n \qquad\qquad M^\dagger \mathbf{1}_n = \mathbf{0}_k. \tag{59}$$

The first condition is what inverts the readout: for any $\mathbf{p} \in \Delta_\circ^n$ and any such $M^\dagger$, taking $\mathbf{h} = M^\dagger\left(\log(\mathbf{p}) - \mathbf{b}\right)$ gives $M\mathbf{h} + \mathbf{b} \simeq \log(\mathbf{p})$ and hence $\mathrm{softmax}(M\mathbf{h} + \mathbf{b}) = \mathbf{p}$. The second says that $M^\dagger\left(\mathbf{x} + z\mathbf{1}_n\right) = M^\dagger \mathbf{x}$ for every $z \in \mathbb{R}$, such that $M^\dagger$ is well defined on $\mathbb{R}^n/\mathbf{1}_n$. This is the natural domain, since the softmax readout cannot distinguish representatives in any case.

Proposition 1. *The readout* $\mathrm{softmax}(M\mathbf{h}+\mathbf{b})$ *is surjective onto* $\Delta_\circ^n$ *if and only if* $M\mathbf{h}+\mathbf{b}$ *ranges over all of* $\mathbb{R}^n/\mathbf{1}_n$*, or equivalently*

$$\mathrm{rank}\left([M \quad \mathbf{1}_n]\right) = n. \tag{60}$$

When Equation 60 is satisfied, $M^\dagger$ *exists, in the sense of Equation 59.*

Proof. Two properties of the softmax are $\mathrm{softmax}(\log(\mathbf{p})) = \mathbf{p}$ for every $\mathbf{p} \in \Delta_\circ^n$, and $\mathrm{softmax}(\mathbf{x} + z\mathbf{1}_n) = \mathrm{softmax}(\mathbf{x})$ for every $\mathbf{x} \in \mathbb{R}^n$ and $z \in \mathbb{R}$. We define $W = [M \quad \mathbf{1}_n] \in \mathbb{R}^{n\times(k+1)}$.

Sufficiency. Suppose Equation 60 holds and let $\mathbf{p}$ be an interior point. Since $\mathbf{p}$ is strictly positive, $\log(\mathbf{p}) \in \mathbb{R}^n$. Because the columns of W span $\mathbb{R}^n$, the system of equations

$$W\begin{bmatrix}\mathbf{h}\\ z\end{bmatrix} = \log(\mathbf{p}) - \mathbf{b} \tag{61}$$

is solvable for $\mathbf{h}$ and z. Evaluated at this solution, $\mathrm{softmax}(M\mathbf{h}+\mathbf{b}) = \mathrm{softmax}(\log(\mathbf{p}) - z\mathbf{1}_n) = \mathrm{softmax}(\log(\mathbf{p})) = \mathbf{p}$. Hence, $\mathrm{softmax}(M\mathbf{h}+\mathbf{b})$ is surjective on the interior of the simplex.

Necessity. Conversely, suppose Equation 60 fails. The columns of W fail to span $\mathbb{R}^n$. Let $V = \mathrm{range}(M) + \mathrm{span}(\mathbf{1}_n) \subset \mathbb{R}^n$, which is a proper subspace. Choose any $\mathbf{w} \in \mathbb{R}^n \setminus V$ and let $\mathbf{p} = \mathrm{softmax}(\mathbf{w}+\mathbf{b})$, which is an interior point. Now suppose, for contradiction, that $\mathrm{softmax}(M\mathbf{h}+\mathbf{b})$ is surjective on the interior. Then, there exists some $\mathbf{h}$ with $\mathrm{softmax}(M\mathbf{h}+\mathbf{b}) = \mathbf{p}$. Writing $z_h = \mathbf{1}_n^\top \exp(M\mathbf{h}+\mathbf{b}) > 0$ and $z_w = \mathbf{1}_n^\top \exp(\mathbf{w}+\mathbf{b}) > 0$, it follows that

$$M\mathbf{h}+\mathbf{b} = \log(\mathbf{p}) + \log(z_h)\mathbf{1}_n, \qquad \mathbf{w}+\mathbf{b} = \log(\mathbf{p}) + \log(z_w)\mathbf{1}_n. \tag{62}$$

Subtracting the second from the first, it follows that $\mathbf{w} = M\mathbf{h} - \log(z_h/z_w)\mathbf{1}_n$ must lie in V, reaching a contradiction. Hence, failure of Equation 60 implies a failure of surjectivity.

Note that $\mathbf{b}$ enters only as a translation and so plays no part in either direction.

Existence. Assume Equation 60 holds, such that W admits a right inverse $WR = I_n$. We take Q to be the first k rows of R, and denote the last row as $\mathbf{q}^\top$. It follows that

$$I_n = WR = MQ + \mathbf{1}_n\mathbf{q}^\top. \tag{63}$$

Let $M^\dagger = Q\left(I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)$. Then $M^\dagger\mathbf{1}_n = Q\left(\mathbf{1}_n - \mathbf{1}_n\right) = \mathbf{0}_k$, and, substituting Equation 63 for MQ,

$$\begin{aligned} MM^\dagger\mathbf{x} &= MQ\left(\mathbf{x} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\mathbf{x}\right) && (64)\\ &= \left(I_n - \mathbf{1}_n\mathbf{q}^\top\right)\left(\mathbf{x} - \frac{1}{n}\left(\mathbf{1}_n^\top\mathbf{x}\right)\mathbf{1}_n\right) && (65)\\ &= \mathbf{x} - \Big(\frac{1}{n}\mathbf{1}_n^\top\mathbf{x} + \mathbf{q}^\top\mathbf{x} - \frac{1}{n}\left(\mathbf{1}_n^\top\mathbf{x}\right)\left(\mathbf{q}^\top\mathbf{1}_n\right)\Big)\mathbf{1}_n && (66)\\ &\simeq \mathbf{x}, && (67)\end{aligned}$$

since the two differ by a multiple of $\mathbf{1}_n$. Both conditions of Equation 59 therefore hold. □

Remark 1. Proposition 1 implies that $\mathrm{rank}(M) \geq n-1$ and hence $k \geq n-1$. This bound is sharp: it is attained by any $M \in \mathbb{R}^{n\times(n-1)}$ whose columns span $\mathbb{R}^n/\mathbf{1}_n$.

Proposition 2. *Let* $M \in \mathbb{R}^{n\times k}$ *be a matrix and* $M^\dagger$ *a right inverse, in the sense of Definition 2. Then, for any* $B \in \mathbb{R}^{n\times d}$*, a matrix* $A \in \mathbb{R}^{k\times d}$ *satisfies*

$$MA\mathbf{x} \simeq B\mathbf{x} \qquad \forall\mathbf{x} \in \mathbb{R}^d \tag{68}$$

if and only if

$$A = M^\dagger B + (I_k - M^\dagger M)C, \tag{69}$$

where $C \in \mathbb{R}^{k\times d}$ *is an arbitrary matrix. A solution always exists.*

Proof. By Equation 59, $MM^\dagger \mathbf{x} - \mathbf{x}$ is a multiple of $\mathbf{1}_n$ for every $\mathbf{x} \in \mathbb{R}^n$. Taking $\mathbf{x}$ over the standard basis and collecting the n multiples into a vector $\mathbf{m} \in \mathbb{R}^n$,

$$MM^\dagger = I_n + \mathbf{1}_n \mathbf{m}^\top, \qquad MM^\dagger M = M + \mathbf{1}_n \left(M^\top \mathbf{m}\right)^\top, \tag{70}$$

the second identity following from the first by right multiplication by M. $MM^\dagger$ is not the identity unless $\mathbf{m} = \mathbf{0}_n$.

Sufficiency. Applying M to both sides of Equation 69, it follows from Equation 70 that

$$MA = MM^\dagger B + \left(M - MM^\dagger M\right) C \tag{71}$$

$$= B + \mathbf{1}_n \mathbf{m}^\top B - \mathbf{1}_n \left(M^\top \mathbf{m}\right)^\top C \tag{72}$$

$$= B + \mathbf{1}_n \left(B^\top \mathbf{m} - C^\top M^\top \mathbf{m}\right)^\top. \tag{73}$$

The two matrices differ by $\mathbf{1}_n$ times a row vector, so $MA\mathbf{x} - B\mathbf{x}$ is a multiple of $\mathbf{1}_n$ for every $\mathbf{x} \in \mathbb{R}^d$ and Equation 68 holds. Taking Equation 69 at $C = 0$ provides a solution for every B.

Necessity. Conversely, suppose A satisfies Equation 68. Taking $\mathbf{x}$ over the standard basis and collecting the d multiples into a vector $\mathbf{w} \in \mathbb{R}^d$ gives $MA = B + \mathbf{1}_n \mathbf{w}^\top$. Left multiplying by $M^\dagger$ and using $M^\dagger \mathbf{1}_n = \mathbf{0}_k$,

$$M^\dagger MA = M^\dagger B + M^\dagger \mathbf{1}_n \mathbf{w}^\top = M^\dagger B. \tag{74}$$

Hence $A - M^\dagger B = A - M^\dagger MA = \left(I_k - M^\dagger M\right) A$, which is Equation 69 with $C = A$. □

Proposition 3. *Let $B \in \mathbb{R}^{n \times d}$ and $A \in \mathbb{R}^{n \times d}$ be any matrices. Then,*

$$A\mathbf{u} \simeq B\mathbf{u} \tag{75}$$

for all one-hot vectors $\mathbf{u} \in \mathbb{R}^d$ implies

$$A\mathbf{x} \simeq B\mathbf{x} \tag{76}$$

for all vectors $\mathbf{x} \in \mathbb{R}^d$.

Proof. Suppose Equation 75. Then $A_{ij} = B_{ij} + z_j$ for all i and j, where each $z_j \in \mathbb{R}$. Hence $A = B + \mathbf{1}_n \mathbf{z}^\top$. It follows that for any $\mathbf{x} \in \mathbb{R}^d$,

$$A\mathbf{x} = B\mathbf{x} + \mathbf{1}_n(\mathbf{z}^\top \mathbf{x}) \tag{77}$$

$$\simeq B\mathbf{x}. \tag{78}$$

□

Lemma 1. *Let $\mathcal{P} = \Delta_\circ^n$. Assume Assumption 2, Assumption 3, and Property I. Then, Property II holds if and only if*

$$MG\mathbf{u} \simeq \log\left(E^\top \mathbf{u}\right) + \log(T\mathbf{1}_n) + (A - I_n)\mathbf{b} \qquad \forall \mathbf{u} \in \mathcal{U} \tag{79}$$

$$MF\mathbf{h} \simeq AM\mathbf{h} \qquad \forall \mathbf{h} \in \mathbb{R}^k \tag{80}$$

where $M^\dagger$ is any right inverse of M (Definition 2) and $A = MFM^\dagger \in \mathbb{R}^{n \times n}$ solves

$$A \log(\boldsymbol{\rho}) \simeq \log\left(T\boldsymbol{\rho}\right) - \log\left(T\mathbf{1}_n\right) \qquad \forall \boldsymbol{\rho} \in \mathbb{R}^n_{++}. \tag{81}$$

Proof. **Necessity.** Assume Property II and let $\hat{\mathbf{p}}_0 = \mathbf{p}_0 \in \mathcal{P}$, such that $\hat{\mathbf{p}}_t = \mathbf{p}_t$ for all time. Then, $\boldsymbol{\rho}_t \propto \hat{\boldsymbol{\rho}}_t$ must be proportional for all $t \geq 0$, or equivalently $\log(\boldsymbol{\rho}_t) \simeq M\mathbf{h}_t + \mathbf{b}$. It follows from Equation 57, Equation 56, $\boldsymbol{\rho}_t \propto \hat{\boldsymbol{\rho}}_t$, and $\hat{\boldsymbol{\rho}}_t = \exp(M\mathbf{h}_t + \mathbf{b})$ that

$$\left[MF\mathbf{h}_{t-1} - \log\left(T\exp(M\mathbf{h}_{t-1} + \mathbf{b})\right)\right] + \left[MG\mathbf{u}_t + \mathbf{b} - \log\left(E^\top \mathbf{u}_t\right)\right] \simeq \mathbf{0}_n, \tag{82}$$

where the constant of proportionality has been absorbed into $\simeq$. Both brackets are finite under Assumption 2 and Assumption 3.

Under Assumption 2 and Assumption 3, every symbol can be emitted from a strictly positive prior, $\mathbb{P}(y_1 = i) = [ET\mathbf{p}_0]_i > 0$ for all i and strictly positive $\mathbf{p}_0$. Every $\mathbf{h}_0$ determines one such strictly

positive prior, and Property I ensures that every strictly positive prior is reached by some $\mathbf{h}_0 \in \mathbb{R}^k$. Then, at time $t = 1$, $\mathbf{h}_0$ and $\mathbf{u}_1$ are jointly unconstrained, and Equation 82 is enforced over the full product space $\mathbb{R}^k \times \mathcal{U}$.

The two brackets depend on arguments that vary independently, so each must be constant under $\simeq$. Hence, there exists a $\mathbf{c} \in \mathbb{R}^n$ such that

$$MF\mathbf{h} - \log\big(T\exp(M\mathbf{h}+\mathbf{b})\big) \simeq \quad \mathbf{c} \qquad \forall \mathbf{h} \in \mathbb{R}^k \tag{83}$$

$$MG\mathbf{u} + \mathbf{b} - \log\big(E^\top \mathbf{u}\big) \simeq -\mathbf{c} \qquad \forall \mathbf{u} \in \mathcal{U}. \tag{84}$$

Consider Equation 83. Property I guarantees that $\mathrm{softmax}(M\mathbf{h}+\mathbf{b})$ is surjective over $\Delta_\circ^n$, such that by Proposition 1, an $M^\dagger$ exists in the sense of Definition 2. For any $\mathbf{h} \in \mathbb{R}^k$, let $\mathbf{h}' = M^\dagger M\mathbf{h}$, satisfying $M\mathbf{h}' = MM^\dagger M\mathbf{h} \simeq M\mathbf{h}$. It follows that $M\mathbf{h}' = M\mathbf{h} + z\mathbf{1}_n$ for some $z \in \mathbb{R}$. Then, $T\exp(M\mathbf{h}'+\mathbf{b}) = e^z T\exp(M\mathbf{h}+\mathbf{b})$ such that $\log\big(T\exp(M\mathbf{h}'+\mathbf{b})\big) \simeq \log\big(T\exp(M\mathbf{h}+\mathbf{b})\big)$.

Subtracting the instances of Equation 83 at $\mathbf{h}$ and at $\mathbf{h}'$, the latter terms and constants cancel under $\simeq$. It follows that

$$\mathbf{0}_n \simeq MF\mathbf{h} - MF\mathbf{h}' = MF\mathbf{h} - MFM^\dagger M\mathbf{h} = MF\mathbf{h} - AM\mathbf{h}, \qquad \forall \mathbf{h} \in \mathbb{R}^k, \tag{85}$$

where we have defined $A := MFM^\dagger$. This gives Equation 80.

For any strictly positive $\boldsymbol{\rho}$, $\mathbf{h} = M^\dagger(\log(\boldsymbol{\rho}) - \mathbf{b})$ will be finite. By Definition 2, $M\mathbf{h}+\mathbf{b} = \log(\boldsymbol{\rho}) + z\mathbf{1}_n$ for some $z \in \mathbb{R}$, such that $\log(T\exp(M\mathbf{h}+\mathbf{b})) \simeq \log(T\boldsymbol{\rho})$. Furthermore, note that $M^\dagger\mathbf{1}_n = \mathbf{0}_k$ gives $A\mathbf{1}_n = MF\left(M^\dagger\mathbf{1}_n\right) = \mathbf{0}_n$, such that Equation 85 reads $MF\mathbf{h} \simeq AM\mathbf{h} = A\left(M\mathbf{h}+\mathbf{b}\right) - A\mathbf{b} = A\log(\boldsymbol{\rho}) - A\mathbf{b}$. Substituting into Equation 83, it follows that

$$A\log(\boldsymbol{\rho}) \simeq \log\big(T\boldsymbol{\rho}\big) + \mathbf{c} + A\mathbf{b} \qquad \forall \boldsymbol{\rho} \in \mathbb{R}^n_{++}. \tag{86}$$

Evaluating at $\boldsymbol{\rho} = \mathbf{1}_n$ gives $\mathbf{c} + A\mathbf{b} \simeq -\log(T\mathbf{1}_n)$, which is finite under Assumption 2. Equation 81 follows.

Next, consider Equation 84. Rearranging and substituting $\mathbf{c} \simeq -\log(T\mathbf{1}_n) - A\mathbf{b}$ from above,

$$MG\mathbf{u} \simeq \log\big(E^\top\mathbf{u}\big) - \mathbf{b} - \mathbf{c} \simeq \log\big(E^\top\mathbf{u}\big) + \log(T\mathbf{1}_n) + (A - I_n)\mathbf{b}, \tag{87}$$

which gives Equation 79.

Sufficiency. Assume Equation 79, Equation 80, and Equation 81. Suppose $\hat{\mathbf{p}}_{t-1} = \mathbf{p}_{t-1}$, such that $\log(\boldsymbol{\rho}_{t-1}) \simeq \log(\hat{\boldsymbol{\rho}}_{t-1}) = M\mathbf{h}_{t-1} + \mathbf{b}$. Then,

$$\log(\hat{\boldsymbol{\rho}}_t) = MF\mathbf{h}_{t-1} + MG\mathbf{u}_t + \mathbf{b} \tag{88}$$

$$\simeq AM\mathbf{h}_{t-1} + \log\big(E^\top\mathbf{u}_t\big) + \log(T\mathbf{1}_n) + (A-I_n)\mathbf{b} + \mathbf{b} \tag{89}$$

$$= A\big(M\mathbf{h}_{t-1}+\mathbf{b}\big) + \log\big(E^\top\mathbf{u}_t\big) + \log(T\mathbf{1}_n) \tag{90}$$

$$= \big[A\log(\hat{\boldsymbol{\rho}}_{t-1}) + \log(T\mathbf{1}_n)\big] + \log\big(E^\top\mathbf{u}_t\big) \tag{91}$$

$$\simeq \log(T\hat{\boldsymbol{\rho}}_{t-1}) + \log\big(E^\top\mathbf{u}_t\big) \tag{92}$$

$$= \log\Big(\big(E^\top\mathbf{u}_t\big) \odot \big(T\hat{\boldsymbol{\rho}}_{t-1}\big)\Big) \tag{93}$$

$$\simeq \log\Big(\big(E^\top\mathbf{u}_t\big) \odot \big(T\boldsymbol{\rho}_{t-1}\big)\Big) \tag{94}$$

$$= \log(\boldsymbol{\rho}_t). \tag{95}$$

The second line invokes Equation 80 and Equation 79, the fifth applies Equation 81 at $\boldsymbol{\rho} = \hat{\boldsymbol{\rho}}_{t-1}$, and the seventh uses $\hat{\boldsymbol{\rho}}_{t-1} \propto \boldsymbol{\rho}_{t-1}$ together with the linearity of T. Hence $\hat{\mathbf{p}}_t = \mathbf{p}_t$. The result follows by induction from $\hat{\mathbf{p}}_0 = \mathbf{p}_0$. □

Remark 2 (Role of the product domain). The separation step in the proof of Lemma 1 relies on Equation 82 holding over the full product $\mathbb{R}^k \times \mathcal{U}$. Two assumptions supply this. Assumption 3 ensures every symbol can be emitted from every strictly positive prior, and Assumption 1, through Property I, ensures every such prior is reached by some $\mathbf{h}_0$. Together they let the pair $(\mathbf{h}_0, \mathbf{u}_1)$ range independently. Consequently, a sum of a function of $\mathbf{h}$ and a function of $\mathbf{u}$ that vanishes identically (under $\simeq$) forces each term to be constant (under $\simeq$). A similar separation argument is used to solve Pexider's equation, $f(x+y) = g(x)+h(y)$ [50, 51]. When either assumption is dropped, the domain is no longer a product and the separation can fail; Section 4.2 gives an example when Assumption 1 is relaxed.

Lemma 2 (Solutions, A)**.** *Assume Assumption 2. Then, there exists an A that solves Equation 81,*

$$A\log(\boldsymbol{\rho}) \simeq \log(T\boldsymbol{\rho}) - \log(T\mathbf{1}_n) \qquad \forall \boldsymbol{\rho} \in \mathbb{R}^n_{++}$$

if and only if T is a permutation matrix or a dense rank-one matrix. In either case the solutions are given by

$$A = \operatorname{diag}(T\mathbf{1}_n)^{-1} T + \mathbf{1}_n \mathbf{v}_n^\top, \tag{96}$$

where $\mathbf{v}_n \in \mathbb{R}^n$ is an arbitrary vector.

Proof. Given Assumption 2, $T\mathbf{1}_n$ is strictly positive and $\operatorname{diag}(T\mathbf{1}_n)$ is invertible. Let $U = \operatorname{diag}(T\mathbf{1}_n)^{-1} T$, such that Equation 81 reads $A\log(\boldsymbol{\rho}) \simeq \log(U\boldsymbol{\rho})$. Note that U is a row-stochastic matrix satisfying $U\mathbf{1}_n = \mathbf{1}_n$. Moreover, since $\operatorname{diag}(T\mathbf{1}_n)^{-1}$ is a strictly positive diagonal matrix, U shares the zero pattern of T.

Necessity. Suppose such an A exists. Then,

$$A\log(\boldsymbol{\rho}) - \log(U\boldsymbol{\rho}) = \beta(\boldsymbol{\rho})\mathbf{1}_n \simeq \mathbf{0}_n \qquad \forall \boldsymbol{\rho} \in \mathbb{R}^n_{++}, \tag{97}$$

for some function $\beta : \mathbb{R}^n_{++} \to \mathbb{R}$. Since U is non-negative with unit row sums, $U\boldsymbol{\rho} > 0$ for every $\boldsymbol{\rho} \in \mathbb{R}^n_{++}$. It follows that, for any i, $\beta(\boldsymbol{\rho}) = \big(A\log(\boldsymbol{\rho}) - \log(U\boldsymbol{\rho})\big)_i$ is a difference of smooth functions and thus a smooth function itself. Since $\boldsymbol{\rho}$ ranges over an open set, we may take derivatives of Equation 97 with respect to $\boldsymbol{\rho}$.

Differentiating the i^{th} component of Equation 97 with respect to ρ_j and evaluating at $\boldsymbol{\rho} = \mathbf{1}_n$, where $U\mathbf{1}_n = \mathbf{1}_n$, gives

$$A_{ij} - U_{ij} = \partial_j \beta(\mathbf{1}_n) \qquad \forall i, j. \tag{98}$$

The right hand side of Equation 98 depends only on the column index j: every solution is of the form

$$A = U + \mathbf{1}_n \mathbf{v}_n^\top. \tag{99}$$

The vector $\mathbf{v}_n$ is arbitrary, impacting only the definition of β. Indeed, whenever a solution $(A, \beta(\boldsymbol{\rho}))$ to Equation 97 exists, the pair $(A + \mathbf{1}_n\mathbf{v}_n^\top,\ \beta(\boldsymbol{\rho}) + \mathbf{v}_n^\top \log(\boldsymbol{\rho}))$ is also a solution, since $\mathbf{1}_n \mathbf{v}_n^\top \log(\boldsymbol{\rho}) \simeq \mathbf{0}_n$. This shift maps the right hand side of Equation 98 from $\partial_j\beta(\mathbf{1}_n)$ to $\partial_j\beta(\mathbf{1}_n) + [\mathbf{v}_n]_j$, which sweeps all of $\mathbb{R}^n$. Hence, under the standing hypothesis that a solution exists, the solution set is exactly Equation 99. It follows that $\mathbf{1}_n$ is an eigenvector of every solution, $A\mathbf{1}_n = (1 + \mathbf{1}_n^\top \mathbf{v}_n)\mathbf{1}_n$.

It remains to determine for which T such an A exists. Let

$$\Delta(\boldsymbol{\rho}) = A\log(\boldsymbol{\rho}) - \log(U\boldsymbol{\rho}) - \beta(\boldsymbol{\rho})\mathbf{1}_n = \mathbf{0}_n. \tag{100}$$

Fix two rows j and k, and subtract the corresponding components:

$$\Delta_j(\boldsymbol{\rho}) - \Delta_k(\boldsymbol{\rho}) = \big(A\log(\boldsymbol{\rho})\big)_j - \big(A\log(\boldsymbol{\rho})\big)_k + \log\left(\frac{(U\boldsymbol{\rho})_k}{(U\boldsymbol{\rho})_j}\right) = 0. \tag{101}$$

We will evaluate this expression at

$$\tilde{\rho}_i = \begin{cases} e^\lambda & i = \ell \\ 1 & \text{else,} \end{cases} \qquad \lambda > 0. \tag{102}$$

Since $\tilde{\boldsymbol{\rho}} \in \mathbb{R}^n_{++}$, $\Delta(\tilde{\boldsymbol{\rho}})$ is well-defined and finite. Note that $\log(\tilde{\rho}_i) = \lambda\delta_{i\ell}$, $[A\log\tilde{\boldsymbol{\rho}}]_i = \lambda A_{i\ell}$ and $[U\tilde{\boldsymbol{\rho}}]_i = 1 + U_{i\ell}(e^\lambda - 1)$, where $\delta_{i\ell}$ is the Kronecker delta. From Equation 99 it follows that $A_{j\ell} - A_{k\ell} = U_{j\ell} - U_{k\ell}$. Hence,

$$U_{j\ell} - U_{k\ell} = \frac{1}{\lambda}\log\big(1 + U_{j\ell}(e^\lambda - 1)\big) - \frac{1}{\lambda}\log\big(1 + U_{k\ell}(e^\lambda - 1)\big) \qquad \forall \lambda > 0. \tag{103}$$

The left hand side of Equation 103 is a fixed number. Each term on the right hand side takes the form $\frac{1}{\lambda}\log\big(1 + U_{i\ell}(e^\lambda - 1)\big)$ and depends on λ. When $U_{i\ell} = 0$, the term vanishes for every λ. When $U_{i\ell} > 0$, the argument of the logarithm is bounded by

$$U_{i\ell}e^\lambda \le 1 + U_{i\ell}(e^\lambda - 1) \le e^\lambda. \tag{104}$$

The lower bound follows from $1 + U_{i\ell}(e^\lambda - 1) - U_{i\ell}e^\lambda = 1 - U_{i\ell} \geq 0$, since $U_{i\ell} \leq 1$. The upper bound follows from $U_{i\ell}(e^\lambda - 1) \leq e^\lambda - 1$, since $U_{i\ell} \leq 1$ and $e^\lambda \geq 1$. Taking logarithms and dividing by λ,

$$1 + \frac{\log(U_{i\ell})}{\lambda} \leq \frac{1}{\lambda} \log\left(1 + U_{i\ell}(e^\lambda - 1)\right) \leq 1. \tag{105}$$

Since $\log(U_{i\ell}) \leq 0$, the term deviates from unity by at most $|\log(U_{i\ell})|/\lambda$.

In short, each term on the right hand side of Equation 103 either equals zero or lies within $|\log(U_{i\ell})|/\lambda$ of unity. Three cases follow. If $U_{j\ell} = U_{k\ell} = 0$, both sides of Equation 103 vanish and no constraint follows. If both entries are positive, $|U_{j\ell} - U_{k\ell}| \leq \big(|\log(U_{j\ell})| + |\log(U_{k\ell})|\big)/\lambda$ for every $\lambda > 0$; a fixed number bounded by an arbitrarily small quantity vanishes, such that $U_{j\ell} = U_{k\ell}$. If exactly one entry is positive, say $U_{j\ell} > 0$ and $U_{k\ell} = 0$ (the opposite case is symmetric), then $|U_{j\ell} - 1| \leq |\log(U_{j\ell})|/\lambda$ for every $\lambda > 0$, such that $U_{j\ell} = 1$. Hence, for every column ℓ and every pair of rows j, k,

$$U_{j\ell}, U_{k\ell} > 0 \implies U_{j\ell} = U_{k\ell}, \qquad U_{j\ell} > 0 \text{ and } U_{k\ell} = 0 \implies U_{j\ell} = 1. \tag{106}$$

Next, we explore two mutually exclusive possibilities.

$$\underset{(a)}{\begin{pmatrix} \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot \\ + & \cdot & \cdot \end{pmatrix}} \implies \underset{(b)}{\begin{pmatrix} \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot \\ 1 & 0 & 0 \end{pmatrix}} \implies \left\{ \underset{(c)}{\begin{pmatrix} + & \cdot & \cdot \\ 0 & \cdot & \cdot \\ 1 & 0 & 0 \end{pmatrix}}, \underset{(d)}{\begin{pmatrix} \cdot & + & \cdot \\ 0 & \cdot & \cdot \\ 1 & 0 & 0 \end{pmatrix}} \right\}$$

Case 1. Suppose U has a zero entry. The argument is illustrated above for $n = 3$, with steps (a)–(d) indicated. A "+" indicates that an entry is positive, "$\cdot$" indicates that, other than being in the interval $[0, 1]$, the entry is unconstrained. (a) Consider a vanishing entry of U (here the $(2, 1)$ entry). Since T is column stochastic, no column of T vanishes, and U shares its zero pattern, it follows that the column of the vanishing entry contains a positive entry in some other row (third row, in the example above). (b) That column now holds a zero and a positive entry, so Equation 106 forces the positive entry to equal unity, and row-stochasticity of U ensures that row is one-hot. Now suppose, for contradiction, that some row of U is not one-hot (the first row, in the example above). Let $U_{r\ell}$ indicate one of its strictly positive entries. The value of $U_{r\ell}$ is strictly less than unity: were $U_{r\ell} = 1$, the unit row sum would zero the remaining entries and the row would be one-hot. Column ℓ either holds (c) the unity of the one-hot row or (d) one of its zeros. In either case Equation 106 forces $U_{r\ell} = 1$ (red pairs in the example above), producing a contradiction. Hence, every row of U is one-hot. Since U and T share a zero pattern, T has exactly one non-zero entry per row, n non-zero elements in the entire matrix. Column stochasticity demands at least one non-zero entry per column, hence exactly one per column, each equal to unity. Hence, T is a permutation matrix.

Case 2. Suppose instead that U, and hence T, is strictly positive. Then every pair of entries in a column is positive, and Equation 106 forces $U_{j\ell} = U_{k\ell}$ for every pair of rows j, k and every column ℓ: every row of U is the same. Unit row sums make $U = \mathbf{1}_n \mathbf{s}^\top$ for a strictly positive probability vector $\mathbf{s}$. Since, by Assumption 2, $\mathbf{r} = T\mathbf{1}_n$ is strictly positive, $T = \mathrm{diag}(\mathbf{r})U = \mathbf{r}\mathbf{s}^\top$ is rank-one and dense. Column stochasticity ensures that $\mathbf{s}$ is proportional to the constant vector: $\mathbf{1}_n^\top T = \mathbf{1}_n^\top = (\mathbf{1}_n^\top \mathbf{r})\mathbf{s}^\top$. Hence, T takes the form $T = \boldsymbol{\pi}\mathbf{1}_n^\top$, for some $\boldsymbol{\pi} = \mathbf{r}/(\mathbf{1}_n^\top \mathbf{r}) \in \mathbb{R}^n_{++}$.

In summary, a solution exists only if T is a permutation matrix or a dense rank-one matrix.

Sufficiency. Suppose T is a permutation matrix. Then $T\mathbf{1}_n = \mathbf{1}_n$ and $U = T$. Since T merely permutes entries, $T\log(\boldsymbol{\rho}) = \log(T\boldsymbol{\rho})$. Hence, $A = U + \mathbf{1}_n \mathbf{v}_n^\top$ gives

$$\begin{aligned} A\log(\boldsymbol{\rho}) &= T\log(\boldsymbol{\rho}) + \big(\mathbf{v}_n^\top \log(\boldsymbol{\rho})\big)\mathbf{1}_n && (107)\\ &\simeq T\log(\boldsymbol{\rho}) && (108)\\ &= \log(T\boldsymbol{\rho}) && (109)\\ &= \log(U\boldsymbol{\rho}). && (110) \end{aligned}$$

Hence, Equation 81 is satisfied.

Suppose instead $T = \boldsymbol{\pi}\mathbf{1}_n^\top$ with $\boldsymbol{\pi}$ strictly positive. Then $U = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ and $A = \mathbf{1}_n\left[\frac{1}{n}\mathbf{1}_n + \mathbf{v}_n\right]^\top$. Both sides of Equation 81 lie along $\mathbf{1}_n$:

$$A\log(\boldsymbol{\rho}) \propto \mathbf{1}_n \simeq \mathbf{0}_n \qquad \log(U\boldsymbol{\rho}) = \log\left(\frac{\mathbf{1}_n^\top\boldsymbol{\rho}}{n}\right)\mathbf{1}_n \simeq \mathbf{0}_n. \tag{111}$$

Hence, $A\log(\boldsymbol{\rho}) \simeq \log(U\boldsymbol{\rho})$ is satisfied. Indeed, if T is a permutation matrix or a dense rank-one matrix, a solution exists. □

Theorem 1 (Solutions, G)**.** *Assume Assumption 2, Assumption 3, Assumption 4, and that $M^\dagger$ exists, in the sense of Definition 2. The family of solutions, G, solving Equation 79 is exactly*

$$G = M^\dagger\left[\log\left(E^\top\right) + \Big(\log(T\mathbf{1}_n) + (MFM^\dagger - I_n)\mathbf{b}\Big)\mathbf{1}_m^\top\right] + (I_k - M^\dagger M)K, \tag{112}$$

where $K \in \mathbb{R}^{k\times m}$ is an arbitrary matrix.

Proof. Under Assumption 4, we may write

$$\log\left(E^\top\mathbf{u}\right) + \log(T\mathbf{1}_n) + (A - I_n)\mathbf{b} = \left[\log\left(E^\top\right) + \Big(\log(T\mathbf{1}_n) + (A - I_n)\mathbf{b}\Big)\mathbf{1}_m^\top\right]\mathbf{u}, \tag{113}$$

for all $\mathbf{u} \in \mathcal{U}$. Since $\mathbf{u}$ enumerates all one-hot vectors in $\mathbb{R}^m$, Equation 79 holds for vectors in $\mathbb{R}^m$, c.f. Proposition 3. The result then follows from Proposition 2. Assumption 3 ensures $\log(E^\top)$ is finite and Assumption 2 ensures $\log(T\mathbf{1}_n)$ is finite. □

Theorem 2 (Solutions, F)**.** *Assume Assumption 2 and that $M^\dagger$ exists, in the sense of Definition 2. Then, there exists an F satisfying Equation 80 whose $A = MFM^\dagger$ solves Equation 81 if and only if T is a permutation matrix or a dense rank-one matrix. In either case, the family of such F is exactly*

$$F = M^\dagger\left[\mathrm{diag}(T\mathbf{1}_n)^{-1}T\right]M + (I_k - M^\dagger M)K, \tag{114}$$

where $K \in \mathbb{R}^{k\times k}$ is an arbitrary matrix.

Proof. By Lemma 2, an A solving Equation 81 exists if and only if T is a permutation matrix or a dense rank-one matrix, in which case $A = U + \mathbf{1}_n\mathbf{v}_n^\top$, for $U = \mathrm{diag}(T\mathbf{1}_n)^{-1}T$.

Necessity. Suppose F satisfies Equation 80 and its $A = MFM^\dagger$ solves Equation 81. Moreover, this A takes the form $A = \mathrm{diag}(T\mathbf{1}_n)^{-1}T + \mathbf{1}_n\mathbf{v}_n^\top$, for some vector $\mathbf{v}_n \in \mathbb{R}^n$. Then, by Lemma 2, T must be a permutation or dense rank-one matrix.

In either case, the solution F takes the following form. Applying Proposition 2 to Equation 80 at $B = AM$ gives $F = M^\dagger AM + (I_k - M^\dagger M)K$. Since $M^\dagger\mathbf{1}_n = \mathbf{0}_k$,

$$F = M^\dagger\left[U + \mathbf{1}_n\mathbf{v}_n^\top\right]M + (I_k - M^\dagger M)K \tag{115}$$

$$= M^\dagger\left[\mathrm{diag}(T\mathbf{1}_n)^{-1}T\right]M + (I_k - M^\dagger M)K, \tag{116}$$

which gives Equation 114.

Sufficiency. Conversely, suppose T is a permutation matrix or a dense rank-one matrix, and let F take the form Equation 114. Multiplying by M on the left and $M^\dagger$ on the right,

$$A = MFM^\dagger = MM^\dagger UMM^\dagger + \left(M - MM^\dagger M\right)KM^\dagger. \tag{117}$$

We reduce each term using Equation 70, together with the row-stochasticity $U\mathbf{1}_n = \mathbf{1}_n$ established in Lemma 2. Indeed, the first term reduces to

$$UMM^\dagger = U\left(I_n + \mathbf{1}_n\mathbf{m}^\top\right) = U + (U\mathbf{1}_n)\,\mathbf{m}^\top = U + \mathbf{1}_n\mathbf{m}^\top, \tag{118}$$

$$MM^\dagger UMM^\dagger = \left(I_n + \mathbf{1}_n\mathbf{m}^\top\right)\left(U + \mathbf{1}_n\mathbf{m}^\top\right) = U + \mathbf{1}_n\Big(\mathbf{m} + U^\top\mathbf{m} + \left(\mathbf{m}^\top\mathbf{1}_n\right)\mathbf{m}\Big)^\top, \tag{119}$$

while the second term becomes

$$\left(M - MM^\dagger M\right)KM^\dagger = -\mathbf{1}_n\left(M^\top\mathbf{m}\right)^\top KM^\dagger = -\mathbf{1}_n\Big(\left(M^\dagger\right)^\top K^\top M^\top\mathbf{m}\Big)^\top. \tag{120}$$

Both corrections are $\mathbf{1}_n$ times a row vector, so $A = U + \mathbf{1}_n\mathbf{w}^\top$ for $\mathbf{w} = \mathbf{m} + U^\top\mathbf{m} + \left(\mathbf{m}^\top\mathbf{1}_n\right)\mathbf{m} - \left(M^\dagger\right)^\top K^\top M^\top\mathbf{m}$. By Lemma 2, this A solves Equation 81.

It remains to verify Equation 80. Applying Proposition 2 to Equation 114 at $B = UM$ gives $MF\mathbf{h} \simeq UM\mathbf{h}$. Since $AM\mathbf{h} = UM\mathbf{h} + \left(\mathbf{w}^\top M\mathbf{h}\right)\mathbf{1}_n \simeq UM\mathbf{h}$, it follows that $MF\mathbf{h} \simeq AM\mathbf{h}$ for every $\mathbf{h} \in \mathbb{R}^k$, and such an F exists. □

D Examples

Below, we provide examples of HMMs whose filters are integrable by one or both readouts. Lemma 2 highlights permutation and rank one, dense matrices as special with respect to softmax-integrability. Indeed, these two regimes are particularly useful for exploring affine-integrability as well.

The forward filter involves both propagating the hidden state and conditioning on evidence. When propagating the belief state involves mixing, the softmax readout struggles, since it must find a linear matrix whose action on log beliefs approximates this mixing. In contrast, the softmax readout excels at conditioning on evidence, since in log-coordinates this step is affine.

An affine-readout appears to struggle in a complementary way. While we do not rigorously enumerate the class of affine-integrable HMMs, we note the following. Consider the Kalman-like form introduced in Equation 34. The mixing step $T\mathbf{p}_{t-1}$ is linear, and therefore simple, but the conditioning step is state-dependent and nonlinear. Affine integrability requires that all state-dependence be carried by the mixing step, such that conditioning is independent of the belief state it acts upon.

The following four examples may be summarized as follows:

	structure	affine-integrable	softmax-integrable
Example 1	T-rank one, dense	yes	yes
Example 2	E-permutation	yes	yes
Example 3	T-permutation	no	yes
Example 4	E-rank one, dense	yes	no

The classifications above each have some caveats, and should be considered heuristic. For example, the structures are not mutually exclusive: an HMM may satisfy several rows at once, occasionally with differing responses as to whether or not the system is integrable. Each row also leaves one of E and T unspecified, and its entries are stated for a *generic* choice of that matrix. The examples below walk through the details.

Example 1 (T-rank one, dense)**.** For any n and m, let $T = \boldsymbol{\pi}\mathbf{1}_n^\top \in \mathbb{R}^{n\times n}$ for some probability vector $\boldsymbol{\pi} \in \mathbb{R}^n_{++}$. Let $E \in \mathbb{R}^{m\times n}$ be any stochastic matrix. Then, the forward recurrence reduces to

$$\boldsymbol{\rho}_t = (E^\top \mathbf{u}_t) \odot (T\boldsymbol{\rho}_{t-1}) = (\mathbf{1}^\top \boldsymbol{\rho}_{t-1})(E^\top \mathbf{u}_t) \odot \boldsymbol{\pi} \propto \operatorname{diag}(\boldsymbol{\pi}) E^\top \mathbf{u}_t \tag{121}$$

such that

$$\mathbf{p}_t = \frac{\boldsymbol{\rho}_t}{\mathbf{1}^\top \boldsymbol{\rho}_t} = \frac{\operatorname{diag}(\boldsymbol{\pi}) E^\top \mathbf{u}_t}{\boldsymbol{\pi}^\top E^\top \mathbf{u}_t} = \operatorname{diag}(\boldsymbol{\pi}) E^\top \operatorname{diag}(E\boldsymbol{\pi})^\dagger \mathbf{u}_t = \Gamma(\boldsymbol{\pi})\mathbf{u}_t, \tag{122}$$

where $\Gamma(\boldsymbol{\pi})$ is the gain matrix defined in Equation 31, evaluated at $\boldsymbol{\pi}$.

In this regime, mixing is maximal: at every time step $T\mathbf{p}_{t-1} = \boldsymbol{\pi}$ clamps the belief state to the stationary vector, erasing the prior in a single step. The softmax readout is therefore never asked to approximate the action of T on log beliefs, and is expected to perform well.

Moreover, the belief state depends only on the most recent observation: there is no opportunity for evidence from two emissions to interfere. As such, the affine readout is expected to perform well.

Indeed, both readouts can ϕ-integrate this system. Let $k = n$, and in both cases set $F = 0$, $M = I_n$ and $\mathbf{b} = \mathbf{0}_n$. Then, this HMM is affine- and softmax-integrable with

$$G = \Gamma(\boldsymbol{\pi}) \quad \text{(affine)}, \qquad\qquad G = \log(\Gamma(\boldsymbol{\pi})) \quad \text{(softmax)}. \tag{123}$$

Notice that when E is strictly positive, so is $\Gamma(\boldsymbol{\pi})$, and $G = \log(\Gamma(\boldsymbol{\pi}))$ will be finite. This is the setting to which Appendix C applies. However, this system remains softmax-integrable, in a generalized sense, when E contains zeros. In this regime, G has components that diverge to negative infinity. The softmax readout regularizes such divergences; components of $\hat{\mathbf{p}}_t$ are exactly zero wherever components of $\mathbf{h}_t = G\mathbf{u}_t$ diverge to negative infinity. Hence, it remains softmax-integrable, but requires an RNN with divergent parameters and hidden state.

Example 2 (E-permutation)**.** With $n = m$, let E be a permutation matrix, $E_{ij} = \delta_{j,\sigma(i)}$, where σ denotes some permutation of $\{1, \ldots, n\}$. Let $T \in \mathbb{R}^{n\times n}$ be any stochastic matrix. Since $\mathbf{u}_t$ is one hot, it follows that $E^\top \mathbf{u}_t = \mathbf{e}_{\sigma(y_t)}$. Hence, the forward recurrence reduces to

$$\boldsymbol{\rho}_t = \mathbf{e}_{\sigma(y_t)} \odot T\boldsymbol{\rho}_{t-1} \propto \mathbf{e}_{\sigma(y_t)} \tag{124}$$

such that

$$\mathbf{p}_t = \mathbf{e}_{\sigma(y_t)} = E^\top \mathbf{u}_t = \Gamma(T\mathbf{p}_{t-1})\mathbf{u}_t, \tag{125}$$

where $\Gamma(T\mathbf{p}_{t-1})$ is the gain matrix defined in Equation 31, which here acts as $E^\top$ on any emission of non-zero probability, from any belief state.

In this regime, mixing occurs, but it never reaches the belief state. At every time step, the observed emission uniquely identifies $\boldsymbol{\rho}_t$ as a one-hot vector, erasing any dependence on the prior. The softmax readout is therefore never asked to approximate the action of T on log beliefs, and is expected to perform well. Notice that the next token probabilities, $\mathbf{q}_t = ET\mathbf{e}_{\sigma(y_t)}$, depend non-trivially on T.

Moreover, the belief state depends only on the most recent observation: there is no opportunity for evidence from two emissions to interfere. As such, the affine readout is expected to perform well.

Indeed, both readouts can ϕ-integrate this system. Let $k = n$, and in both cases set $F = 0$, $M = I_n$ and $\mathbf{b} = \mathbf{0}_n$. Then, this HMM is affine- and softmax-integrable with

$$G = E^\top \quad \text{(affine)}, \qquad\qquad G = \log(E^\top) \quad \text{(softmax)}. \tag{126}$$

By similar logic as in Example 1, this HMM is softmax-integrable in the generalized sense, with G carrying components that diverge to negative infinity where E is zero.

Example 3 (T-permutation)**.** For any n and m, let T be a permutation matrix, $T_{ij} = \delta_{i,\sigma(j)}$, where σ denotes some permutation of $\{1, \ldots, n\}$. Let $E \in \mathbb{R}^{m\times n}$ be any stochastic matrix.

In this regime, there is no mixing, since T is a permutation matrix. It follows that T commutes with the logarithm, $\log(T\boldsymbol{\rho}) = T\log(\boldsymbol{\rho})$, and, as such, the softmax readout is expected to perform well. Indeed it does. Appendix C proves this system is softmax-integrable, for example, by $k = n$, $M = I_n$, $\mathbf{b} = \mathbf{0}_n$, $F = T$ and $G = \log(E^\top)$. By similar logic to Example 1, this HMM remains softmax-integrable when E contains zeros, in a generalized sense.

Since T has no decaying modes, this HMM will, for a generic E, have infinite memory. The belief state will depend non-trivially on all previous emissions, and the gain matrix $\Gamma(T\mathbf{p}_{t-1})$ will remain state dependent. As such, this HMM is not expected to be affine-integrable. However, if E is rank-one, this reduces to Example 4, and the HMM regains affine-integrability. This is not a proof, of course, as we do not enumerate the space of affine-integrable HMMs in this manuscript.

Example 4 (E-rank one, dense)**.** For any n and m, let $E = \boldsymbol{\pi}\mathbf{1}_n^\top \in \mathbb{R}^{m\times n}$ for some probability vector $\boldsymbol{\pi} \in \mathbb{R}^m_{++}$. Let $T \in \mathbb{R}^{n\times n}$ be any stochastic matrix. Then, the forward recurrence reduces to

$$\boldsymbol{\rho}_t = \pi_{y_t}\mathbf{1} \odot T\boldsymbol{\rho}_{t-1}, \tag{127}$$

such that

$$\mathbf{p}_t = \frac{\boldsymbol{\rho}_t}{\mathbf{1}^\top\boldsymbol{\rho}_t} = T\mathbf{p}_{t-1}, \tag{128}$$

which describes pure mixing and no evidence accrual. These hidden dynamics are non-trivial, but unobservable. At every time step, symbols are emitted i.i.d., $\mathbf{q}_t = ET\mathbf{p}_t = \boldsymbol{\pi}$. We can nevertheless discuss integrability.

In this regime, there is only mixing. As such, the softmax readout is expected not to perform well. Indeed, Appendix C proves that this system is only softmax-integrable if T is a permutation or rank-one, dense matrix, which reduces this example to Example 1 or Example 3.

Since this HMM has no evidence accrual, the affine readout is expected to perform well. Indeed, this system is clearly affine-integrable, for example, by $k = n$, $M = I_n$, $\mathbf{b} = \mathbf{0}_n$, $F = T$ and $G = 0_{n\times m}$.

E Hilbert distance and approximation error bounds

In this section, we examine properties of the Hilbert metric and derive dissipation relations on the error of the RNN approximation. We restrict our analysis to strictly positive vectors, $\mathbf{p} \in \mathbb{R}^n_{++}$, meaning every component satisfies $p_i > 0$, and to strictly positive probability vectors, $\mathbf{p} \in \Delta^n_\circ$, which additionally sum to one and so assign nonzero probability to every latent state. The implications of

this restriction are discussed in the main text. For any $\mathbf{p} \in \mathbb{R}^n_{++}$, we use the notation $(\mathbf{p}^{-1})_i = 1/p_i$ to denote its inverse under the Hadamard product.

The Hilbert distance is defined between two such vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n_{++}$ as

$$d(\mathbf{p}, \mathbf{q}) = \max_{i,j} \log \left(\frac{p_i q_j}{p_j q_i} \right), \qquad \forall \mathbf{p}, \mathbf{q} \in \mathbb{R}^n_{++}. \tag{129}$$

The Hilbert distance is non-negative, and finite for all strictly positive inputs. It is normalization invariant

$$d(z\mathbf{p}, z'\mathbf{q}) = d(\mathbf{p}, \mathbf{q}) \qquad \forall z, z' > 0. \tag{130}$$

Like cosine similarity, the Hilbert distance is projective, measuring only the difference of *direction* between two vectors.

Following Seneta [40], we call a nonnegative matrix $T \in \mathbb{R}^{n \times m}_+$ allowable if it has no rows or columns that are identically zero. For any allowable T, the Birkhoff contraction coefficient, c_T, is defined via the relation:

$$d(T\mathbf{p}, T\mathbf{q}) \leq c_T \, d(\mathbf{p}, \mathbf{q}) \qquad c_T = \tanh \left(\frac{1}{4} \Delta_T \right). \tag{131}$$

The argument of the $\tanh$,

$$\Delta_T = \max_{ijkl} \log \frac{T_{ik} T_{jl}}{T_{jk} T_{il}} = \max_{kl} d(T_{:k}, T_{:l}) = \max_{ij} d(T_{i:}, T_{j:}), \tag{132}$$

is referred to as the projective diameter of T. Here, we adopt the convention $\log(x/0) = +\infty$ for $x > 0$, and exclude from the maximum any quadruple for which $T_{ik}T_{jl} = T_{jk}T_{il} = 0$, such that $\Delta_T = \infty$ and $c_T = 1$ are permitted. For usage below, we also define

$$\delta_T := \max_i \left(\max_j \log(T_{ij}) - \min_j \log(T_{ij}) \right) = \max_i d(T_{i:}, \mathbf{1}_m). \tag{133}$$

While Δ_T describes how much the rows of T deviate from each other with respect to the Hilbert distance, δ_T measures how much they deviate from uniform. While $\Delta_T = \Delta_{T^\top}$, due to the symmetry of Equation 132, a similar relation does not hold for δ_T.

Equation 131 was originally established by Birkhoff [39] for square matrices and is stated for rectangular matrices by Cavazos-Cadena [52]. Seneta [40] provides a broad introduction to this coefficient and its usage.

The Birkhoff contraction coefficient is strictly less than 1 whenever T is strictly positive. In particular, if T is column stochastic with $T_{ij} \geq \epsilon > 0$ for all i, j, then $c_T < \frac{1-\epsilon}{1+\epsilon}$. This means that Hilbert distances contract absolutely under the application of any strictly positive T [40]. If T is allowable but contains a zero, then $\Delta_T = \infty$ and $c_T = 1$, so distances no longer need contract. Consistently, the above bound tends to 1 in the limit $\epsilon \to 0$.

Although most proofs in this section follow directly from the definition of the Hilbert distance, and will not require external assumptions, some proofs in Section E.2 will reference the assumptions listed in Section 4 as needed.

E.1 Hilbert Distance Identities

Proposition 4 (Hilbert Metric Invariance to Inversion)**.** *Let $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n_{++}$. Then, the Hilbert distance satisfies*

$$d(\mathbf{p}^{-1}, \mathbf{q}^{-1}) = d(\mathbf{p}, \mathbf{q}).$$

Proof. By inspection

$$d(\mathbf{p}^{-1}, \mathbf{q}^{-1}) = \max_{i,j} \log \left(\frac{(1/p_i)(1/q_j)}{(1/p_j)(1/q_i)} \right) = \max_{i,j} \log \left(\frac{q_i p_j}{q_j p_i} \right) = d(\mathbf{p}, \mathbf{q}). \tag{134}$$

□

Lemma 3 (Hadamard triangle-inequality)**.** *For* $\mathbf{r}, \mathbf{s}, \mathbf{p}, \mathbf{q} \in \mathbb{R}^n_{++}$*:*

$$|d(\mathbf{r}, \mathbf{s}) - d(\mathbf{p}, \mathbf{q})| \leq d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}) \leq d(\mathbf{r}, \mathbf{s}) + d(\mathbf{p}, \mathbf{q}). \tag{135}$$

Proof. Recall that

$$\max_i p_i q_i \leq \left(\max_i p_i\right)\left(\max_j q_j\right) \qquad \min_i p_i q_i \geq \left(\min_i p_i\right)\left(\min_j q_j\right).$$

Therefore,

$$\begin{aligned} d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}) &= \log \frac{\max_i r_i p_i / s_i q_i}{\min_i r_i p_i / s_i q_i} \\ &\leq \log \frac{(\max_i r_i / s_i)\,(\max_i p_i / q_i)}{\min_i r_i p_i / s_i q_i} \\ &\leq \log \frac{(\max_i r_i / s_i)\,(\max_i p_i / q_i)}{(\min_i r_i / s_i)\,(\min_i p_i / q_i)} \\ &= \log \frac{\max_i r_i / s_i}{\min_i r_i / s_i} + \log \frac{\max_i p_i / q_i}{\min_i p_i / q_i} \\ &= d(\mathbf{r}, \mathbf{s}) + d(\mathbf{p}, \mathbf{q}). \end{aligned}$$

This proves the upper bound.

Next, we prove the lower bound, which uses the upper bound in its proof. For any vector, $\mathbf{p} = \mathbf{p} \odot \mathbf{1}_n$ and for any strictly positive vector, $\mathbf{1}_n = \mathbf{p}^{-1} \odot \mathbf{p}$. Hence,

$$\begin{aligned} d(\mathbf{r}, \mathbf{s}) &= d(\mathbf{r} \odot \mathbf{p} \odot \mathbf{p}^{-1}, \mathbf{s} \odot \mathbf{q} \odot \mathbf{q}^{-1}) \\ &\leq d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}) + d(\mathbf{p}^{-1}, \mathbf{q}^{-1}) \\ &= d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}) + d(\mathbf{p}, \mathbf{q}) \end{aligned}$$

and

$$\begin{aligned} d(\mathbf{p}, \mathbf{q}) &= d(\mathbf{p} \odot \mathbf{r} \odot \mathbf{r}^{-1}, \mathbf{q} \odot \mathbf{s} \odot \mathbf{s}^{-1}) \\ &\leq d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}) + d(\mathbf{r}^{-1}, \mathbf{s}^{-1}) \\ &= d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}) + d(\mathbf{r}, \mathbf{s}). \end{aligned}$$

Since both

$$d(\mathbf{r}, \mathbf{s}) - d(\mathbf{p}, \mathbf{q}) \leq d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}) \qquad d(\mathbf{p}, \mathbf{q}) - d(\mathbf{r}, \mathbf{s}) \leq d(\mathbf{r} \odot \mathbf{p}, \mathbf{s} \odot \mathbf{q}),$$

the lower bound follows.

□

Proposition 5 (Quadratic, Reverse Pinsker-type Inequality)**.** *For* $\mathbf{p}, \mathbf{q} \in \Delta^n_\circ$*:*

$$D_{\mathrm{KL}}(\mathbf{p} \| \mathbf{q}) \leq \frac{1}{8}\, d(\mathbf{p}, \mathbf{q})^2.$$

Proof. Define the log ratios $r_i = \log \frac{p_i}{q_i}$ and

$$r_{\max} = \max_i r_i \qquad r_{\min} = \min_i r_i. \tag{136}$$

By definition, $d(\mathbf{p}, \mathbf{q}) = r_{\max} - r_{\min}$. Let R be a random variable, taking on values r_i with probability q_i, and

$$\psi(\lambda) = \log \mathbb{E}[e^{\lambda R}].$$

A few identities follow:

$$\psi(0) = \log \sum_i q_i = \log 1 = 0 \tag{137}$$

$$\psi(1) = \log \sum_i q_i \frac{p_i}{q_i} = \log 1 = 0 \tag{138}$$

$$\psi'(1) = \frac{\sum_i q_i \frac{p_i}{q_i} \log \frac{p_i}{q_i}}{1} = D_{\mathrm{KL}}(\mathbf{p} \| \mathbf{q}). \tag{139}$$

Evaluating ψ at $\lambda = 0$ via an expansion about $\lambda = 1$ gives

$$\psi(0) = \psi(1) - \psi'(1) + \frac{1}{2}\psi''(\theta)$$

for some $\theta \in [0, 1]$. Plugging in,

$$0 = -D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) + \frac{1}{2}\psi''(\theta).$$

Notice that $\psi''(\theta) = \mathrm{Var}_{\mathbf{q}(\theta)}[R]$ is precisely the variance of R under a tilted probability distribution, $q_i(\theta) = q_i e^{\theta r_i} / \sum_j q_j e^{\theta r_j}$. Bounding this value bounds the KL-divergence.

Popoviciu's inequality [53] states that a random variable, X, bounded between m and $M \geq m$, has bounded variance:

$$\mathrm{Var}(X) \leq \frac{(M-m)^2}{4}.$$

This result holds regardless of how X is distributed. Hence, this result holds for R distributed according to $\mathbf{q}(\theta)$. Since R has diameter $d(\mathbf{p}, \mathbf{q})$, it follows that

$$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) \leq \frac{1}{8}\, d(\mathbf{p}, \mathbf{q})^2.$$

□

Proposition 6 (Linear, Reverse Pinsker-type Inequality)**.** *For $\mathbf{p}, \mathbf{q} \in \Delta_\circ^n$:*

$$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) \leq d(\mathbf{p}, \mathbf{q}).$$

Proof. Since

$$\sum_i q_i = 1 = \sum_i p_i,$$

there exists at least one index j such that $p_j \leq q_j$, otherwise we would have $\sum_i p_i > \sum_i q_i$. Hence, $\min_i \frac{p_i}{q_i} \leq 1$. Similarly, there exists k such that $p_k \geq q_k$, otherwise $\sum_i p_i < \sum_i q_i$, and therefore $\max_i \frac{p_i}{q_i} \geq 1$.

It follows that p_i/q_i is uniformly bounded by the Hilbert distance:

$$\frac{p_i}{q_i} = \frac{1}{q_i/p_i} \leq \frac{1}{\min_j q_j/p_j} \leq \frac{\max_k q_k/p_k}{\min_j q_j/p_j} = e^{d(\mathbf{p},\mathbf{q})}.$$

Consequently, in the KL divergence:

$$D_{\mathrm{KL}}(\mathbf{p}||\mathbf{q}) = \sum_i p_i \log \frac{p_i}{q_i} \leq \sum_i p_i \log e^{d(\mathbf{p},\mathbf{q})} = d(\mathbf{p}, \mathbf{q}).$$

□

Lemma 4 (Reverse Pinsker-type Inequality)**.** *For $\mathbf{p}, \mathbf{q} \in \Delta_\circ^n$:*

$$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) \leq \min\left\{ d(\mathbf{p}, \mathbf{q}), \frac{1}{8} d(\mathbf{p}, \mathbf{q})^2 \right\}.$$

Proof. This is a direct consequence of Proposition 5 and Proposition 6. □

Lemma 5. *Let $E \in \mathbb{R}_{++}^{m \times n}$ be any strictly positive, column stochastic matrix. Then*

$$\delta_E \leq 4 \operatorname{arctanh}(c_E) \leq 2\delta_E, \tag{140}$$

where c_E is the Birkhoff contraction coefficient of E and δ_E is given by Equation 133. The upper bound holds for all strictly positive E. The lower bound requires column stochasticity.

Proof. Since $\Delta_E = 4\,\mathrm{arctanh}(c_E)$, c.f. Equation 131, it is sufficient to show that

$$\delta_E \leq \Delta_E \leq 2\delta_E. \tag{141}$$

Upper bound. Each term in the maximum defining Δ_E splits into two row spreads,

$$\log \frac{E_{ik}E_{jl}}{E_{jk}E_{il}} = \big(\log E_{ik} - \log E_{il}\big) + \big(\log E_{jl} - \log E_{jk}\big) \leq 2\delta_E. \tag{142}$$

Each bracket compares two entries lying in a single row of E, and is therefore bounded above by δ_E. Maximizing over i, j, k, l gives $\Delta_E \leq 2\delta_E$. This argument uses only strict positivity.

Lower bound. If $\delta_E = 0$ the bound is immediate, so assume $\delta_E > 0$ and fix indices i, k, l attaining it,

$$\delta_E = \log E_{ik} - \log E_{il}, \qquad \text{such that} \qquad E_{ik} > E_{il}.$$

Because E is column stochastic, columns k and l have equal sums, $\sum_p E_{pk} = 1 = \sum_p E_{pl}$. There must therefore exist a row j with $E_{jk} < E_{jl}$: otherwise $E_{pk} \geq E_{pl}$ for every p, with strict inequality at $p = i$, forcing $\sum_p E_{pk} > \sum_p E_{pl}$. For this j,

$$\Delta_E \geq \log \frac{E_{ik}E_{jl}}{E_{jk}E_{il}} = \delta_E + \log \frac{E_{jl}}{E_{jk}} > \delta_E,$$

which completes the proof. □

E.2 Dissipation Relations

Proposition 7 (Expected belief)**.** *Let T and E be column stochastic matrices. Let $\mathbf{p}_0 \in \Delta_\circ^n$ be a strictly positive probability vector and let $\mathbf{p}_t$ be its evolution under Equation 56. Then,*

$$\mathbb{E}[\mathbf{p}_t \mid \mathbf{p}_0] = T^t \mathbf{p}_0.$$

Proof. Write $p_{t,j} = \mathbb{P}(x_t = j \mid y_{1:t})$ for the components of the belief state, and recall that $\mathbf{p}_0$ is the prior on x_0. The expectation is taken over observation sequences, which are themselves distributed according to the HMM. Summing over every such sequence and applying the product rule,

$$\mathbb{E}[p_{t,j} \mid \mathbf{p}_0] = \sum_{y_{1:t}} \mathbb{P}(y_{1:t})\mathbb{P}(x_t = j \mid y_{1:t}) = \sum_{y_{1:t}} \mathbb{P}(x_t = j, y_{1:t}) = \mathbb{P}(x_t = j), \tag{143}$$

since the sequences $y_{1:t}$ partition the sample space. Furthermore,

$$\mathbb{P}(x_t = i) = \sum_{x_{t-1}} \mathbb{P}(x_t = i, x_{t-1}) = \sum_{x_{t-1}} \mathbb{P}(x_t = i | x_{t-1})\mathbb{P}(x_{t-1}) = \sum_j T_{ij}\mathbb{P}(x_{t-1} = j). \tag{144}$$

Applying this iteratively, it follows that $\mathbb{P}(x_t = j) = [T^t \mathbf{p}_0]_j$. The result follows. □

Remark 3. When T is primitive, it admits a unique stationary density $\boldsymbol{\pi}$, and $T^t \mathbf{p}_0 \to \boldsymbol{\pi}$ for every $\mathbf{p}_0$.

Proposition 8. *Assume Assumption 2 and Assumption 3. Let $\mathbf{p}_0 \in \mathbb{R}^n_{++}$ be strictly positive and let $\mathbf{p}_t$ be its evolution under Equation 56. Then, $\mathbf{p}_t \in \mathbb{R}^n_{++}$ for all finite time. This result is independent of the observed symbol sequence, $y_{1:t}$.*

Proof. Suppose $\mathbf{p}_{t-1} \in \mathbb{R}^n_{++}$ is strictly positive. By Assumption 2, each row of T must have some non-zero element. The corresponding element in $\mathbf{p}_{t-1}$ will also be non-zero, such that $\sum_j T_{ij} p_{t-1,j} > 0$ has at least one non-zero contribution for all i. Hence, $T\mathbf{p}_{t-1}$ is strictly positive. By Assumption 3, $E^\top \mathbf{u}$ is strictly positive as well. Finally, $(E^\top \mathbf{u}_t) \odot (T\mathbf{p}_{t-1})$ will be strictly positive, since the Hadamard product of two strictly positive vectors is strictly positive. We note that the normalizer $\mathbf{u}_t^\top E T \mathbf{p}_{t-1} = (E^\top \mathbf{u}_t) \cdot (T\mathbf{p}_{t-1}) > 0$ is strictly positive as well, for all $\mathbf{u}_t$ and $\mathbf{p}_{t-1}$, since it is a dot product of two strictly positive vectors. Hence, $\mathbf{p}_t = (\mathbf{u}_t^\top E T \mathbf{p}_{t-1})^{-1}(E^\top \mathbf{u}_t) \odot (T\mathbf{p}_{t-1})$ is strictly positive. The result follows by induction from $\mathbf{p}_0$ being strictly positive. □

Proposition 9. *Assume Assumption 2, Assumption 3, and Assumption 4. Let $\mathbf{p}_{t-1} \in \mathbb{R}^n_{++}$ be strictly positive and let $\mathbf{p}_t$ be its evolution under Equation 1. Let $\mathbf{h}_{t-1} \in \mathbb{R}^k$ and let $\mathbf{h}_t$ be its evolution under Equation 2, where $F \in \mathbb{R}^{k\times k}$ and $G \in \mathbb{R}^{k\times m}$ are finite. Let $\hat{\mathbf{p}}_{t-1}, \hat{\mathbf{p}}_t \in \Delta^n_\circ$ be the corresponding readouts, defined in Equation 2, where $M \in \mathbb{R}^{n\times k}$ and $\mathbf{b} \in \mathbb{R}^n$ are finite. Then, the Hilbert distance between $\mathbf{p}_t$ and $\hat{\mathbf{p}}_t$ satisfies a dissipation relation*

$$d(\mathbf{p}_t, \hat{\mathbf{p}}_t) \leq c_T\, d(\mathbf{p}_{t-1}, \hat{\mathbf{p}}_{t-1}) + s(\mathbf{h}_{t-1}, \mathbf{u}_t) \tag{145}$$

where c_T is the Birkhoff contraction coefficient of T and

$$s(\mathbf{h}_{t-1}, \mathbf{u}_t) = d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\boldsymbol{\phi}(M\mathbf{h}_{t-1} + \mathbf{b})\big), \boldsymbol{\phi}(M(F\mathbf{h}_{t-1} + G\mathbf{u}_t) + \mathbf{b})\Big) \tag{146}$$

is the injected error.

Evaluated on the solution family, where $\boldsymbol{\phi} = \mathrm{softmax}$, M satisfies Equation 60 and F and G are given by Equation 6 and Equation 7, respectively, the injected error reduces to

$$s(\mathbf{h}_{t-1}, \mathbf{u}_t) = d\Big(U\hat{\mathbf{p}}_{t-1}, \exp(U\log(\hat{\mathbf{p}}_{t-1}))\Big) := s_{t-1}, \tag{147}$$

where $U = \mathrm{diag}(T\mathbf{1}_n)^{-1}T$ and $\hat{\mathbf{p}}_t = \mathrm{softmax}(M\mathbf{h}_t + \mathbf{b})$. Since the dependence on $\mathbf{u}_t$ vanishes, and $\mathbf{h}_{t-1}$ enters only through $\hat{\mathbf{p}}_{t-1}$, we refer to this error in the main text simply as s_{t-1}.

Proof. Notice that $\hat{\mathbf{p}}_{t-1}$ and $\hat{\mathbf{p}}_t$ are strictly positive, by assumption. For $\boldsymbol{\phi} = \mathrm{softmax}$, this holds automatically: because M, $\mathbf{b}$, and $\mathbf{h}_{t-1}$ are finite, $\hat{\mathbf{p}}_{t-1}$ is strictly positive, and because F, G, and $\mathbf{u}_t$ (by Assumption 4) are finite, $\mathbf{h}_t$ is finite and $\hat{\mathbf{p}}_t$ is strictly positive as well. Moreover, it follows from Assumption 2 and Assumption 3 that $\mathbf{p}_t$ is strictly positive, see Proposition 8.

Then, using the normalization invariance and the triangle inequality of the Hilbert distance, it follows that

$$d(\mathbf{p}_t, \hat{\mathbf{p}}_t) = d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big), \hat{\mathbf{p}}_t\Big) \tag{148}$$

$$\leq d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big), \big(E^\top \mathbf{u}_t\big) \odot \big(T\hat{\mathbf{p}}_{t-1}\big)\Big) + d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\hat{\mathbf{p}}_{t-1}\big), \hat{\mathbf{p}}_t\Big). \tag{149}$$

Consider the first term. It follows from Lemma 3 evaluated at $\mathbf{r} = \mathbf{s}$ that for any $\mathbf{r}, \mathbf{p}, \mathbf{q} \in \mathbb{R}^n_{++}$, $d(\mathbf{r} \odot \mathbf{p}, \mathbf{r} \odot \mathbf{q}) = d(\mathbf{p}, \mathbf{q})$. By Assumption 3, $\mathbf{r} = E^\top \mathbf{u}_t$ is strictly positive, such that

$$d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big), \big(E^\top \mathbf{u}_t\big) \odot \big(T\hat{\mathbf{p}}_{t-1}\big)\Big) = d\big(T\mathbf{p}_{t-1}, T\hat{\mathbf{p}}_{t-1}\big) \tag{150}$$

$$\leq c_T\, d\big(\mathbf{p}_{t-1}, \hat{\mathbf{p}}_{t-1}\big), \tag{151}$$

where in the last line we have used Equation 131.

Now consider the second term. Expanding,

$$d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\hat{\mathbf{p}}_{t-1}\big), \hat{\mathbf{p}}_t\Big) = d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\boldsymbol{\phi}(M\mathbf{h}_{t-1} + \mathbf{b})\big), \boldsymbol{\phi}(M\mathbf{h}_t + \mathbf{b})\Big) \tag{152}$$

$$= d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\boldsymbol{\phi}(M\mathbf{h}_{t-1} + \mathbf{b})\big), \boldsymbol{\phi}(M(F\mathbf{h}_{t-1} + G\mathbf{u}_t) + \mathbf{b})\Big) \tag{153}$$

$$:= s(\mathbf{h}_{t-1}, \mathbf{u}_t). \tag{154}$$

Plugging Equation 151 and Equation 154 into Equation 149 gives Equation 145.

Next, suppose that $\boldsymbol{\phi} = \mathrm{softmax}$, that M satisfies Equation 60, and that F and G are given by Equation 6 and Equation 7, respectively. Then,

$$MF = UM + \mathbf{1}_n \mathbf{v}_1^\top \qquad MG = \log\big(E^\top\big) + \Big(\log(T\mathbf{1}_n) + (U - I_n)\mathbf{b}\Big)\mathbf{1}_m^\top + \mathbf{1}_n \mathbf{v}_2^\top, \tag{155}$$

where $\mathbf{v}_1 \in \mathbb{R}^k$ and $\mathbf{v}_2 \in \mathbb{R}^m$ are vectors arising because $MM^\dagger \neq I_n$ in general (see Equation 70). Notice that the values of these vectors do not matter, since $\mathrm{softmax}((A + \mathbf{1}_n\mathbf{v}^\top)\mathbf{x}) = \mathrm{softmax}(A\mathbf{x})$ for all A, $\mathbf{x}$, and $\mathbf{v}$. It follows that

$$\boldsymbol{\phi}(M(F\mathbf{h}_{t-1} + G\mathbf{u}_t) + \mathbf{b}) = \mathrm{softmax}(U(M\mathbf{h}_{t-1} + \mathbf{b}) + (I_n - U)\mathbf{b} + MG\mathbf{u}_t) \tag{156}$$

$$= \mathrm{softmax}(U\log(\hat{\boldsymbol{\rho}}_{t-1}) + \log(E^\top)\mathbf{u}_t + \log(T\mathbf{1}_n)) \tag{157}$$

$$\propto \exp(U\log(\hat{\boldsymbol{\rho}}_{t-1}) + \log(E^\top)\mathbf{u}_t + \log(T\mathbf{1}_n)) \tag{158}$$

$$= \big(V^\top \mathbf{u}_t\big) \odot \exp(U\log(\hat{\boldsymbol{\rho}}_{t-1})) \tag{159}$$

$$\propto \big(V^\top \mathbf{u}_t\big) \odot \exp(U\log(\hat{\mathbf{p}}_{t-1})), \tag{160}$$

where $V = E\,\mathrm{diag}(T\mathbf{1}_n)$, $\log(\hat{\boldsymbol{\rho}}_t) = M\mathbf{h}_t + \mathbf{b}$ and $\hat{\mathbf{p}}_t = \mathrm{softmax}(M\mathbf{h}_t + \mathbf{b})$. In the third to last line, we have replaced $\mathrm{softmax}$ with $\exp$; since the Hilbert distance is normalization invariant, the normalizer of softmax will be irrelevant. In the last line, we have replaced $\log(\hat{\boldsymbol{\rho}}_{t-1})$ with $\log(\hat{\mathbf{p}}_{t-1})$: since $\log(\hat{\boldsymbol{\rho}}_{t-1}) \simeq \log(\hat{\mathbf{p}}_{t-1})$ and $U\mathbf{1}_n = \mathbf{1}_n$, this replacement results in an overall normalization factor that will be similarly disregarded by the Hilbert distance.

If we examine the other argument, it follows more simply that

$$\big(E^\top \mathbf{u}_t\big) \odot \big(T\boldsymbol{\phi}(M\mathbf{h}_{t-1} + \mathbf{b})\big) = \big(E^\top \mathbf{u}_t\big) \odot \big(T\hat{\mathbf{p}}_{t-1}\big) = \big(V^\top \mathbf{u}_t\big) \odot \big(U\hat{\mathbf{p}}_{t-1}\big). \tag{161}$$

It follows from Assumption 2 and Assumption 3 that $V^\top \mathbf{u}_t$ is strictly positive for all $\mathbf{u}_t$. Hence, applying Lemma 3 again, we find that, evaluated on the solution family,

$$\begin{aligned} s(\mathbf{h}_{t-1}, \mathbf{u}_t) &= d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\boldsymbol{\phi}(M\mathbf{h}_{t-1} + \mathbf{b})\big), \boldsymbol{\phi}(M(F\mathbf{h}_{t-1} + G\mathbf{u}_t) + \mathbf{b})\Big) && (162)\\ &= d\Big(\big(V^\top \mathbf{u}_t\big) \odot \big(U\hat{\mathbf{p}}_{t-1}\big), \big(V^\top \mathbf{u}_t\big) \odot \exp(U\log(\hat{\mathbf{p}}_{t-1}))\Big) && (163)\\ &= d\big(U\hat{\mathbf{p}}_{t-1}, \exp(U\log(\hat{\mathbf{p}}_{t-1}))\big), && (164) \end{aligned}$$

which gives Equation 147. □

Theorem 3 (Pathwise deviation of belief from stationarity)**.** *Assume Assumption 2, Assumption 3, and Assumption 4. Let $\boldsymbol{\pi} \in \Delta_\circ^n$ be stationary, $T\boldsymbol{\pi} = \boldsymbol{\pi}$. Let $\mathbf{p}_0 \in \mathbb{R}^n_{++}$ be strictly positive and let $\mathbf{p}_t$ be its evolution under Equation 56. Then, for all $t \geq 1$,*

$$d(\mathbf{p}_t, \boldsymbol{\pi}) \leq c_T\, d(\mathbf{p}_{t-1}, \boldsymbol{\pi}) + 4\,\mathrm{arctanh}(c_E). \tag{165}$$

This bound holds for every observation sequence $y_{1:t}$.

Proof. Since $\mathbf{p}_t \propto \big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big)$, it follows from the normalization invariance of the Hilbert distance that

$$d(\mathbf{p}_t, \boldsymbol{\pi}) = d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big), \boldsymbol{\pi}\Big). \tag{166}$$

Since $\boldsymbol{\pi} = \boldsymbol{\pi} \odot \mathbf{1}_n$ and $T\boldsymbol{\pi} = \boldsymbol{\pi}$, it follows from Lemma 3 that

$$\begin{aligned} d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big), \boldsymbol{\pi}\Big) &= d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big), \mathbf{1}_n \odot T\boldsymbol{\pi}\Big) && (167)\\ &\leq d(E^\top \mathbf{u}_t, \mathbf{1}_n) + d(T\mathbf{p}_{t-1}, T\boldsymbol{\pi}). && (168) \end{aligned}$$

Assumption 2 and Assumption 3 ensure that all arguments are strictly positive.

By Equation 131, $d(T\mathbf{p}_{t-1}, T\boldsymbol{\pi}) \leq c_T d(\mathbf{p}_{t-1}, \boldsymbol{\pi})$. Given Assumption 4, it follows from Equation 133 and Lemma 5 that $d(E^\top \mathbf{u}_t, \mathbf{1}_n) = d(E_{y_t:}, \mathbf{1}_n) \leq \delta_E \leq 4\,\mathrm{arctanh}(c_E)$. Combining these bounds,

$$\begin{aligned} d(\mathbf{p}_t, \boldsymbol{\pi}) &= d\Big(\big(E^\top \mathbf{u}_t\big) \odot \big(T\mathbf{p}_{t-1}\big), \boldsymbol{\pi}\Big) && (169)\\ &\leq d(T\mathbf{p}_{t-1}, T\boldsymbol{\pi}) + d(E^\top \mathbf{u}_t, \mathbf{1}_n) && (170)\\ &\leq c_T d(\mathbf{p}_{t-1}, \boldsymbol{\pi}) + \delta_E && (171)\\ &\leq c_T d(\mathbf{p}_{t-1}, \boldsymbol{\pi}) + 4\,\mathrm{arctanh}(c_E). && (172) \end{aligned}$$

□

Remark 4. If multiple such $\boldsymbol{\pi}$ exist, Theorem 3 holds for all $\boldsymbol{\pi}$ simultaneously. Since Theorem 3 holds for *any* emission sequence $y_{1:t}$, it should be considered a worst-case bound. Typical sequences may induce smaller deviations from $\boldsymbol{\pi}$ than indicated by this worst-case bound.

Proposition 10. *Assume Assumption 2. Let $T \in \mathbb{R}_+^{n\times n}$ be a column stochastic matrix with a Birkhoff contraction coefficient that is less than unity, $c_T < 1$. Then,*

1. *T is strictly positive.*
2. *The stationary state, $\boldsymbol{\pi} = T\boldsymbol{\pi} \in \Delta^n$, is strictly positive.*
3. *The stationary state is unique.*

Proof. **Item 1.** Suppose that T is not strictly positive, such that $T_{jk} = 0$ for some j and k. By column stochasticity, there exists an i such that $T_{ik} > 0$, and by Assumption 2, there exists an l such that $T_{jl} > 0$. Evaluating Equation 132 at the quadruple (i, j, k, l) gives $\Delta_T \geq \log\left(T_{ik}T_{jl}/(T_{jk}T_{il})\right) = \infty$, since the numerator is strictly positive and the denominator vanishes. An infinite projective diameter corresponds to $c_T = 1$. By contraposition, $c_T < 1$ implies T is strictly positive.

Item 2. Let $\boldsymbol{\pi} \in \Delta^n$ be stationary under T and let T be strictly positive, by Item 1. Then, $\pi_i = \sum_j T_{ij}\pi_j > 0$ for all i, since every $T_{ij} > 0$, every $\pi_j \geq 0$, and at least one $\pi_j > 0$ because $\mathbf{1}_n^\top \boldsymbol{\pi} = 1$. Hence, $\boldsymbol{\pi} \in \Delta_\circ^n$ is strictly positive. Such a $\boldsymbol{\pi}$ exists: since T is column stochastic, it maps the compact, convex set Δ^n continuously into itself, and hence has a fixed point in Δ^n by the Brouwer fixed point theorem.

Item 3. Let $\boldsymbol{\pi}, \boldsymbol{\pi}' \in \Delta^n$ both be stationary under T. By Item 2, both are strictly positive, such that $d(\boldsymbol{\pi}, \boldsymbol{\pi}')$ is finite. By Equation 131, $d(\boldsymbol{\pi}, \boldsymbol{\pi}') = d(T\boldsymbol{\pi}, T\boldsymbol{\pi}') \leq c_T d(\boldsymbol{\pi}, \boldsymbol{\pi}')$. Since $c_T < 1$, this may hold only if $d(\boldsymbol{\pi}, \boldsymbol{\pi}') = 0$, such that $\boldsymbol{\pi}' = z\boldsymbol{\pi}$ for some $z > 0$. Since both sum to one, $z = 1$ and $\boldsymbol{\pi}' = \boldsymbol{\pi}$. □

Theorem 4 (Confinement of belief about stationarity)**.** *Assume Assumption 2, Assumption 3, Assumption 4, and that the Birkhoff contraction coefficient of T is less than unity, $c_T < 1$. Let $\mathbf{p}_0 \in \mathbb{R}^n_{++}$ be strictly positive and let $\mathbf{p}_t$ be its evolution under Equation 56. Then, T admits a unique, strictly positive stationary state, $\boldsymbol{\pi} = T\boldsymbol{\pi}$, and the belief state satisfies, for all $t \geq 0$,*

$$d(\mathbf{p}_t, \boldsymbol{\pi}) \leq c_T^t d(\mathbf{p}_0, \boldsymbol{\pi}) + \frac{4\,\mathrm{arctanh}(c_E)}{1 - c_T}, \tag{173}$$

where c_E is the Birkhoff contraction coefficient of E. Consequently, $\limsup_{t\to\infty} d(\mathbf{p}_t, \boldsymbol{\pi}) \leq 4\,\mathrm{arctanh}(c_E)/(1 - c_T)$.

Proof. By Proposition 10, $c_T < 1$ implies that T admits a unique stationary state $\boldsymbol{\pi} \in \Delta_\circ^n$, which is strictly positive. Writing $d_t = d(\mathbf{p}_t, \boldsymbol{\pi})$ and $\kappa = 4\,\mathrm{arctanh}(c_E)$, Theorem 3 gives $d_t \leq c_T d_{t-1} + \kappa$ for all $t \geq 1$. Here, κ is finite, since E is strictly positive by Assumption 3, such that $c_E < 1$. Unrolling this recursion,

$$d_t \leq c_T^t d_0 + \kappa \sum_{\tau=0}^{t-1} c_T^\tau = c_T^t d_0 + \kappa \frac{1 - c_T^t}{1 - c_T} \leq c_T^t d_0 + \frac{\kappa}{1 - c_T}. \tag{174}$$

The bound holds trivially at $t = 0$, which completes the first claim. Since $\mathbf{p}_0$ and $\boldsymbol{\pi}$ are strictly positive, d_0 is finite, and since $c_T < 1$, $c_T^t d_0 \to 0$ as $t \to \infty$. Taking the limit superior of both sides gives the second claim. □

F Methods

Code for all analyses and dataset generation can be found at https://github.com/AndrewMah44/linear-hmms.git

F.1 Model details

Numerical experiments were performed using a network of the form

$$\mathbf{h}_t = F\mathbf{h}_{t-1} + G\mathbf{u}_t \tag{175}$$

$$\hat{\mathbf{q}}_t = \mathrm{softmax}(A\mathbf{h}_t + \mathbf{d}), \tag{176}$$

with F constrained to have a spectral radius less than or equal to unity. The model components were initialized as follows:

1. Recurrent weights: $F \in \mathbb{R}^{n\times n}$ was initialized as an orthogonal matrix scaled by 0.1.
2. Input embedding: $G \in \mathbb{R}^{n\times m}$ was initialized row-wise from a standard Gaussian distribution and normalized to unit length.
3. Readout weights and bias: $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{d} \in \mathbb{R}^m$ were sampled element-wise uniformly over the range $[-n^{-0.5}, n^{-0.5}]$.

All linear RNNs were implemented using the Equinox library [54].

F.2 Dataset generation

For each HMM, we generated datasets using the Dynamax library. Random HMM datasets consisted of $1,048,576 = 2^{20}$ training sequences, $20,000$ validation sequences, and $50,000$ testing sequences. All sequences consisted of 256 observations. Clustered HMM datasets consisted of $1,048,576 = 2^{20}$ training sequences, $50,000$ validation sequences, and $50,000$ testing sequences. All sequences consisted of 128 observations.

F.3 Model training

Training was performed using the Adam optimizer from the Optax library. RNNs were initialized with a unique random seed. Training began with a learning rate of 0.001. The training data were divided into mini-batches of 64 trials, and the mean categorical cross-entropy loss with weight-decay (10^{-4}) was used to update model parameters. After each epoch, the validation loss was computed. If the validation loss failed to improve upon the previous minimum, the learning rate for the subsequent epoch was reduced by half; otherwise it remained the same. Training was terminated once the validation loss failed to improve for three consecutive epochs. Models were fit on an NVIDIA RTX A6000 (48 GB VRAM, CUDA 12.8, driver 570.172.08). Each training run required between 0.25 and 1 hour on a single RTX A6000 GPU.

F.4 Linear decoding analysis

To assess the learned representation of the linear RNNs, we trained a simple linear decoder to map the RNN hidden-unit activity, $\boldsymbol{h}_t \in \mathbb{R}^{16}$, to the optimal log-belief state, $\log \boldsymbol{\rho}_t \in \mathbb{R}^{16}$ of the form

$$\log(\boldsymbol{\rho}_t) \approx W\boldsymbol{h}_t + \boldsymbol{b}$$

where $W \in \mathbb{R}^{16\times 16}$ and $\boldsymbol{b} \in \mathbb{R}^{16}$ are learnable parameters. W and $\boldsymbol{b}$ were initialized by sampling element-wise uniformly over the range $[-16^{-0.5}, 16^{-0.5}]$. We also trained linear decoders on the hidden-unit activity of randomly initialized linear RNNs (see Section F.1 for initialization details).

Training was performed using the Adam optimizer with a softmax cross-entropy loss on the true belief state. Training ran for 5,000 steps on the entire dataset. Models were fit on an NVIDIA RTX A6000 (48 GB VRAM, CUDA 12.8, driver 570.172.08). Each training run required between 15 and 45 seconds on a single RTX A6000 GPU.

G Additional Data

G.1 Figures

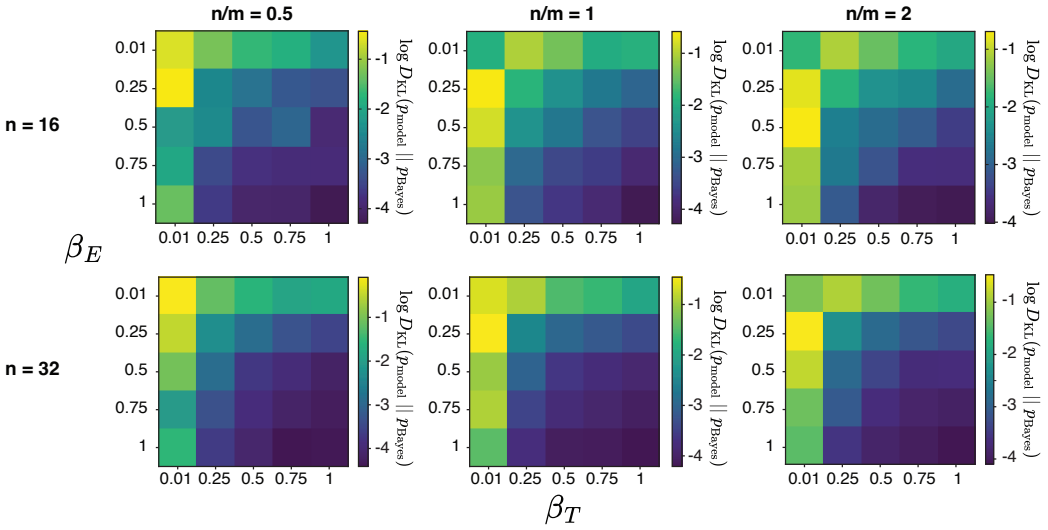


Figure 4: **Linear RNN performance as a function of β_T, β_E is consistent across Random HMM size parameters**. Top row: HMMs with $n = 16$ latent states, Bottom row: HMMs with $n = 32$ latent states. Left Column: HMMs with ratio of latent states to emissions $n/m = 0.5$, Center Column: HMMs with ratio of latent states to emissions $n/m = 1$, Right Column: HMMs with ratio of latent states to emissions $n/m = 2$.

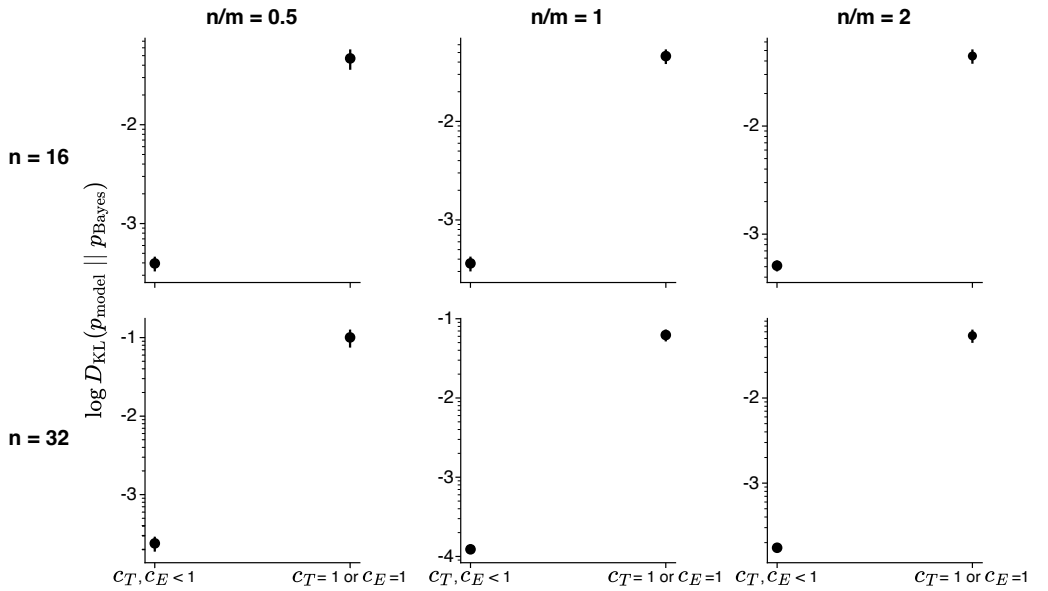


Figure 5: **Linear RNN performance as a function of c_T, c_E is consistent across Random HMM size parameters**. Top row: HMMs with $n = 16$ latent states, Bottom row: HMMs with $n = 32$ latent states. Left Column: HMMs with ratio of latent states to emissions $n/m = 0.5$, Center Column: HMMs with ratio of latent states to emissions $n/m = 1$, Right Column: HMMs with ratio of latent states to emissions $n/m = 2$.

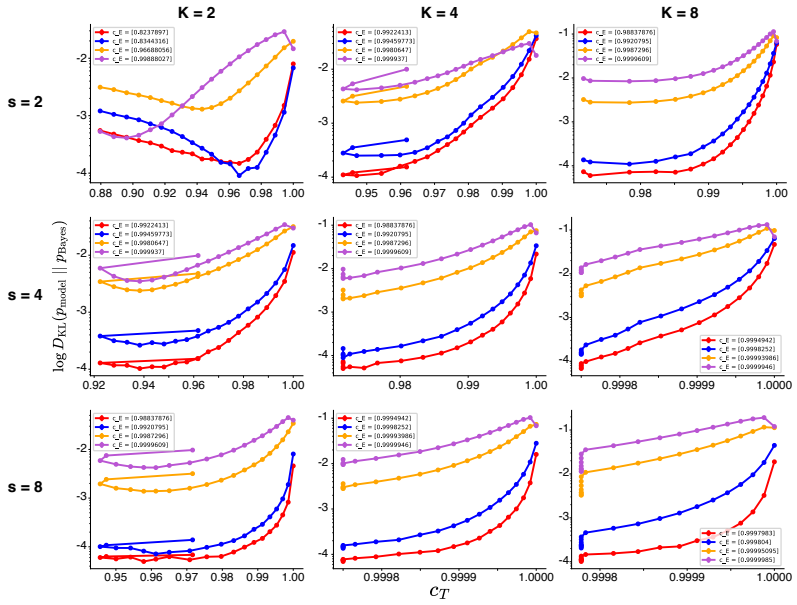


Figure 6: **Linear RNN performance as a function of c_T, c_E is consistent across Clustered HMM size parameters**. Columns are number of clusters ($K = 2, 4, 8$), while rows are number of states per cluster ($s = 2, 4, 8$).

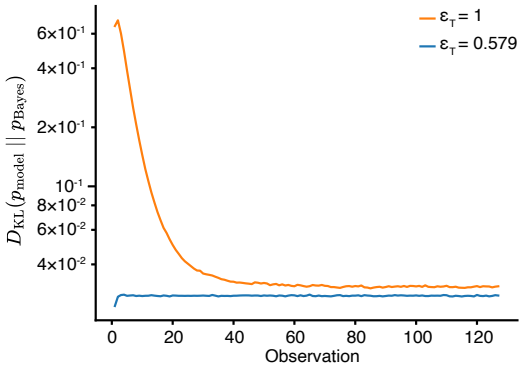


Figure 7: **Linear RNN performance for disconnected clusters as a function of number of observations**.

G.2 Raw Data for $n = 16$

Table 1: Average $D_{KL}(p_{\mathrm{Model}} \,||\, p_{\mathrm{Bayes}})$ for randomly generated HMMs

(β_E, β_T)	Init. 1	Init. 2	Init. 3	Init. 4	Init. 5	**Mean**
$\boldsymbol{n/m = 0.5}$						
$(0.01, 0.01)$	2.275×10^{-1}	2.234×10^{-1}	2.250×10^{-1}	2.256×10^{-1}	2.278×10^{-1}	2.259×10^{-1}
$(0.01, 0.25)$	5.481×10^{-2}	5.495×10^{-2}	5.494×10^{-2}	5.491×10^{-2}	5.501×10^{-2}	5.492×10^{-2}
$(0.01, 0.5)$	2.096×10^{-2}	2.090×10^{-2}	2.096×10^{-2}	2.093×10^{-2}	2.096×10^{-2}	2.094×10^{-2}
$(0.01, 0.75)$	1.444×10^{-2}	1.433×10^{-2}	1.439×10^{-2}	1.436×10^{-2}	1.442×10^{-2}	1.439×10^{-2}
$(0.01, 1)$	4.875×10^{-3}	4.862×10^{-3}	4.890×10^{-3}	4.952×10^{-3}	4.927×10^{-3}	4.901×10^{-3}
$(0.25, 0.01)$	3.641×10^{-1}	3.619×10^{-1}	3.605×10^{-1}	3.638×10^{-1}	3.605×10^{-1}	3.622×10^{-1}
$(0.25, 0.25)$	2.989×10^{-3}	3.003×10^{-3}	2.988×10^{-3}	3.004×10^{-3}	2.986×10^{-3}	2.994×10^{-3}

Continued on next page

(β_E, β_T)	Init. 1	Init. 2	Init. 3	Init. 4	Init. 5	**Mean**
$(0.25, 0.5)$	1.430×10^{-3}	1.434×10^{-3}	1.439×10^{-3}	1.445×10^{-3}	1.430×10^{-3}	1.436×10^{-3}
$(0.25, 0.75)$	5.731×10^{-4}	5.711×10^{-4}	5.725×10^{-4}	5.663×10^{-4}	5.667×10^{-4}	5.699×10^{-4}
$(0.25, 1)$	4.654×10^{-4}	4.615×10^{-4}	4.600×10^{-4}	4.624×10^{-4}	4.665×10^{-4}	4.632×10^{-4}
$(0.5, 0.01)$	6.115×10^{-3}	6.254×10^{-3}	6.642×10^{-3}	6.217×10^{-3}	6.215×10^{-3}	6.289×10^{-3}
$(0.5, 0.25)$	3.421×10^{-3}	3.408×10^{-3}	3.398×10^{-3}	3.417×10^{-3}	3.407×10^{-3}	3.410×10^{-3}
$(0.5, 0.5)$	5.175×10^{-4}	5.123×10^{-4}	5.119×10^{-4}	5.053×10^{-4}	5.067×10^{-4}	5.107×10^{-4}
$(0.5, 0.75)$	9.089×10^{-4}	9.169×10^{-4}	9.007×10^{-4}	9.076×10^{-4}	8.961×10^{-4}	9.061×10^{-4}
$(0.5, 1)$	1.240×10^{-4}	1.200×10^{-4}	1.248×10^{-4}	1.182×10^{-4}	1.183×10^{-4}	1.211×10^{-4}
$(0.75, 0.01)$	9.995×10^{-3}	1.010×10^{-2}	1.011×10^{-2}	1.000×10^{-2}	9.809×10^{-3}	1.000×10^{-2}
$(0.75, 0.25)$	3.539×10^{-4}	3.605×10^{-4}	3.506×10^{-4}	3.754×10^{-4}	3.712×10^{-4}	3.623×10^{-4}
$(0.75, 0.5)$	1.736×10^{-4}	1.826×10^{-4}	1.770×10^{-4}	1.791×10^{-4}	1.720×10^{-4}	1.769×10^{-4}
$(0.75, 0.75)$	1.532×10^{-4}	1.508×10^{-4}	1.489×10^{-4}	1.435×10^{-4}	1.461×10^{-4}	1.485×10^{-4}
$(0.75, 1)$	1.397×10^{-4}	1.351×10^{-4}	1.418×10^{-4}	1.392×10^{-4}	1.364×10^{-4}	1.384×10^{-4}
$(1, 0.01)$	4.274×10^{-2}	4.280×10^{-2}	4.259×10^{-2}	4.272×10^{-2}	4.272×10^{-2}	4.271×10^{-2}
$(1, 0.25)$	2.389×10^{-4}	2.288×10^{-4}	2.219×10^{-4}	2.248×10^{-4}	2.243×10^{-4}	2.277×10^{-4}
$(1, 0.5)$	9.495×10^{-5}	1.006×10^{-4}	9.391×10^{-5}	9.607×10^{-5}	1.049×10^{-4}	9.809×10^{-5}
$(1, 0.75)$	9.158×10^{-5}	8.798×10^{-5}	9.223×10^{-5}	9.103×10^{-5}	9.243×10^{-5}	9.105×10^{-5}
$(1, 1)$	5.231×10^{-5}	5.340×10^{-5}	5.466×10^{-5}	5.108×10^{-5}	4.526×10^{-5}	5.134×10^{-5}
$\boldsymbol{n/m = 1}$						
$(0.01, 0.01)$	1.191×10^{-2}	1.166×10^{-2}	1.173×10^{-2}	1.164×10^{-2}	1.220×10^{-2}	1.183×10^{-2}
$(0.01, 0.25)$	9.161×10^{-2}	9.140×10^{-2}	9.165×10^{-2}	9.090×10^{-2}	9.161×10^{-2}	9.143×10^{-2}
$(0.01, 0.5)$	3.954×10^{-2}	3.941×10^{-2}	3.943×10^{-2}	3.938×10^{-2}	3.949×10^{-2}	3.945×10^{-2}
$(0.01, 0.75)$	1.043×10^{-2}	1.035×10^{-2}	1.021×10^{-2}	1.027×10^{-2}	1.037×10^{-2}	1.033×10^{-2}
$(0.01, 1)$	1.248×10^{-2}	1.242×10^{-2}	1.253×10^{-2}	1.238×10^{-2}	1.244×10^{-2}	1.245×10^{-2}
$(0.25, 0.01)$	2.630×10^{-1}	2.613×10^{-1}	2.629×10^{-1}	2.625×10^{-1}	2.621×10^{-1}	2.623×10^{-1}
$(0.25, 0.25)$	1.374×10^{-2}	1.370×10^{-2}	1.372×10^{-2}	1.376×10^{-2}	1.374×10^{-2}	1.373×10^{-2}
$(0.25, 0.5)$	3.834×10^{-3}	3.836×10^{-3}	3.840×10^{-3}	3.834×10^{-3}	3.834×10^{-3}	3.836×10^{-3}
$(0.25, 0.75)$	1.547×10^{-3}	1.557×10^{-3}	1.546×10^{-3}	1.551×10^{-3}	1.555×10^{-3}	1.551×10^{-3}
$(0.25, 1)$	8.030×10^{-4}	8.030×10^{-4}	8.098×10^{-4}	7.971×10^{-4}	8.037×10^{-4}	8.033×10^{-4}
$(0.5, 0.01)$	1.539×10^{-1}	1.541×10^{-1}	1.541×10^{-1}	1.539×10^{-1}	1.540×10^{-1}	1.540×10^{-1}
$(0.5, 0.25)$	4.066×10^{-3}	4.069×10^{-3}	4.071×10^{-3}	4.066×10^{-3}	4.063×10^{-3}	4.067×10^{-3}
$(0.5, 0.5)$	1.495×10^{-3}	1.500×10^{-3}	1.484×10^{-3}	1.495×10^{-3}	1.494×10^{-3}	1.493×10^{-3}
$(0.5, 0.75)$	4.540×10^{-4}	4.539×10^{-4}	4.505×10^{-4}	4.496×10^{-4}	4.593×10^{-4}	4.535×10^{-4}
$(0.5, 1)$	2.672×10^{-4}	2.710×10^{-4}	2.772×10^{-4}	2.672×10^{-4}	2.614×10^{-4}	2.688×10^{-4}
$(0.75, 0.01)$	5.734×10^{-2}	5.736×10^{-2}	5.728×10^{-2}	5.740×10^{-2}	5.728×10^{-2}	5.733×10^{-2}
$(0.75, 0.25)$	9.346×10^{-4}	9.350×10^{-4}	9.306×10^{-4}	9.286×10^{-4}	9.346×10^{-4}	9.327×10^{-4}
$(0.75, 0.5)$	3.269×10^{-4}	3.354×10^{-4}	3.202×10^{-4}	3.250×10^{-4}	3.363×10^{-4}	3.288×10^{-4}
$(0.75, 0.75)$	2.333×10^{-4}	2.346×10^{-4}	2.365×10^{-4}	2.396×10^{-4}	2.249×10^{-4}	2.338×10^{-4}
$(0.75, 1)$	1.072×10^{-4}	1.159×10^{-4}	1.063×10^{-4}	1.119×10^{-4}	1.083×10^{-4}	1.099×10^{-4}
$(1, 0.01)$	7.137×10^{-2}	7.137×10^{-2}	7.139×10^{-2}	7.145×10^{-2}	7.134×10^{-2}	7.138×10^{-2}
$(1, 0.25)$	4.398×10^{-4}	4.452×10^{-4}	4.524×10^{-4}	4.465×10^{-4}	4.470×10^{-4}	4.462×10^{-4}
$(1, 0.5)$	2.070×10^{-4}	2.025×10^{-4}	2.049×10^{-4}	2.083×10^{-4}	2.047×10^{-4}	2.055×10^{-4}
$(1, 0.75)$	1.135×10^{-4}	1.104×10^{-4}	1.138×10^{-4}	1.221×10^{-4}	1.150×10^{-4}	1.150×10^{-4}
$(1, 1)$	5.305×10^{-5}	5.012×10^{-5}	5.576×10^{-5}	5.498×10^{-5}	5.067×10^{-5}	5.292×10^{-5}
$\boldsymbol{n/m = 2}$						
$(0.01, 0.01)$	1.563×10^{-2}	1.578×10^{-2}	1.566×10^{-2}	1.607×10^{-2}	1.569×10^{-2}	1.577×10^{-2}
$(0.01, 0.25)$	7.952×10^{-2}	7.991×10^{-2}	7.917×10^{-2}	7.979×10^{-2}	7.948×10^{-2}	7.958×10^{-2}
$(0.01, 0.5)$	3.155×10^{-2}	3.134×10^{-2}	3.132×10^{-2}	3.146×10^{-2}	3.139×10^{-2}	3.141×10^{-2}
$(0.01, 0.75)$	1.491×10^{-2}	1.484×10^{-2}	1.478×10^{-2}	1.485×10^{-2}	1.489×10^{-2}	1.485×10^{-2}
$(0.01, 1)$	9.772×10^{-3}	9.805×10^{-3}	9.803×10^{-3}	9.762×10^{-3}	9.848×10^{-3}	9.798×10^{-3}
$(0.25, 0.01)$	1.497×10^{-1}	1.489×10^{-1}	1.493×10^{-1}	1.489×10^{-1}	1.496×10^{-1}	1.493×10^{-1}
$(0.25, 0.25)$	1.375×10^{-2}	1.377×10^{-2}	1.375×10^{-2}	1.376×10^{-2}	1.373×10^{-2}	1.375×10^{-2}

Continued on next page

(β_E, β_T)	Init. 1	Init. 2	Init. 3	Init. 4	Init. 5	**Mean**
$(0.25, 0.5)$	4.619×10^{-3}	4.633×10^{-3}	4.628×10^{-3}	4.631×10^{-3}	4.621×10^{-3}	4.626×10^{-3}
$(0.25, 0.75)$	3.619×10^{-3}	3.622×10^{-3}	3.627×10^{-3}	3.616×10^{-3}	3.625×10^{-3}	3.622×10^{-3}
$(0.25, 1)$	1.456×10^{-3}	1.454×10^{-3}	1.446×10^{-3}	1.455×10^{-3}	1.458×10^{-3}	1.454×10^{-3}
$(0.5, 0.01)$	2.027×10^{-1}	2.032×10^{-1}	2.027×10^{-1}	2.031×10^{-1}	2.032×10^{-1}	2.030×10^{-1}
$(0.5, 0.25)$	2.549×10^{-3}	2.542×10^{-3}	2.539×10^{-3}	2.552×10^{-3}	2.531×10^{-3}	2.542×10^{-3}
$(0.5, 0.5)$	1.428×10^{-3}	1.433×10^{-3}	1.434×10^{-3}	1.427×10^{-3}	1.443×10^{-3}	1.433×10^{-3}
$(0.5, 0.75)$	8.904×10^{-4}	8.930×10^{-4}	9.052×10^{-4}	8.946×10^{-4}	9.087×10^{-4}	8.984×10^{-4}
$(0.5, 1)$	3.745×10^{-4}	3.866×10^{-4}	3.873×10^{-4}	3.814×10^{-4}	3.941×10^{-4}	3.848×10^{-4}
$(0.75, 0.01)$	7.064×10^{-2}	6.991×10^{-2}	6.994×10^{-2}	7.022×10^{-2}	7.069×10^{-2}	7.028×10^{-2}
$(0.75, 0.25)$	2.064×10^{-3}	2.055×10^{-3}	2.055×10^{-3}	2.054×10^{-3}	2.056×10^{-3}	2.057×10^{-3}
$(0.75, 0.5)$	6.862×10^{-4}	6.826×10^{-4}	6.906×10^{-4}	6.902×10^{-4}	6.834×10^{-4}	6.866×10^{-4}
$(0.75, 0.75)$	2.597×10^{-4}	2.680×10^{-4}	2.590×10^{-4}	2.589×10^{-4}	2.636×10^{-4}	2.619×10^{-4}
$(0.75, 1)$	2.546×10^{-4}	2.536×10^{-4}	2.537×10^{-4}	2.537×10^{-4}	2.532×10^{-4}	2.538×10^{-4}
$(1, 0.01)$	6.415×10^{-2}	6.498×10^{-2}	6.468×10^{-2}	6.491×10^{-2}	6.451×10^{-2}	6.465×10^{-2}
$(1, 0.25)$	8.266×10^{-4}	8.361×10^{-4}	8.245×10^{-4}	8.314×10^{-4}	8.322×10^{-4}	8.302×10^{-4}
$(1, 0.5)$	1.556×10^{-4}	1.559×10^{-4}	1.585×10^{-4}	1.615×10^{-4}	1.655×10^{-4}	1.594×10^{-4}
$(1, 0.75)$	1.102×10^{-4}	1.069×10^{-4}	1.035×10^{-4}	1.082×10^{-4}	1.043×10^{-4}	1.066×10^{-4}
$(1, 1)$	9.618×10^{-5}	9.493×10^{-5}	1.005×10^{-4}	9.465×10^{-5}	9.243×10^{-5}	9.575×10^{-5}

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:[Yes].

   Justification: All claims made in the abstract are accurately reflected in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the main text, see Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Proofs are provided in the appendix in full generality, and summaries that should be interpretable by a broad audience are provided in the main text.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full details are provided in the appendix to recreate the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: A link to a GitHub repository for dataset generation and analyses in Section 5 is provided. Details to fully recreate both the models and datasets are also provided in the text.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental details (including dataset generation, hyperparameters, and optimizers) are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Statistical tests are described in the paper, and error-bars are provided when applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are provided in the experimental methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We ensured that the research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer:[N/A].

    Justification: Our results are theoretical in nature and will be of interest to the broader machine learning and physics communities. We do not expect any immediate societal impacts from their dissemination.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer:[N/A].

    Justification: Our work does not present a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All data and models were explicitly generated by the authors. Python packages and the right panel of Figure 1, which were not generated by the authors, are properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: This work uses only standard model classes and synthetic datasets. Sufficient details are provided to ensure reproducibility.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: This work does not use human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer:[N/A].

    Justification: This work does not use human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs do not play a central or novel role in the method developed in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.