CMPUT 654: Mathematical Foundations of Modern AI Systems Fall 2026

Lecture 5: State-space models and system identification

September 15, 2026

Lecturer: Csaba Szepesvári *Note prepared by: Csaba Szepesvári*

Mark Holmstrom prepared a scribe of the lecture. Csaba Szepesvári prepared and expanded the present notes.

About these lecture notes. This lecture asks how a structured hidden state can explain a sequence, how to predict and generate after observing a prefix, and how to fit the model from observations. The main construction first fits observed lag coefficients, then recovers a compact recurrent predictor by matrix factorization. Fitting a dynamical model from data is called *system identification*.

For language, a latent state has a direct modeling interpretation. Spoken or written language can be treated as a partial and noisy observation of an evolving internal process: a speaker can form or retain thoughts that are not expressed, and both the utterance and its reception can affect what happens next. This is a motivating hypothesis, not a claim that the models in this lecture describe human cognition faithfully. At a finer resolution, motor commands, articulation, acoustics, and perception introduce further latent variables; a state model chooses which distinctions to represent. This explains why latent variables are natural for language, while leaving open whether explicitly fitting them improves prediction. A transformer also computes hidden representations, but its conditional likelihood does not usually marginalize over a stochastic latent trajectory. An observable conditional model can therefore be easier to train and more effective even when a latent-process description is scientifically plausible.

Scope and reading guide. The main text follows the classroom sequence: nonlinear state-space models, conditional generation, estimation and identifiability, latent likelihoods and EM, observable prediction, linear models, and the subspace algorithm. It develops the questions about memory, parameter sharing, backpropagation, and attention raised in class. The synthetic comparisons are additions to the lecture. The optional section after this development justifies the algorithm and relates estimation error to prediction, sequence KL, and generation. Cayley–Hamilton was not presented in class; it is retained as an optional noiseless comparison and an exercise. After the take-home messages, the notes continue with endnotes, bibliographical notes, the glossary, exercises, appendices, and references.

Notation. States are $S_t \in \mathcal{S}$, observations are $Z_t \in \mathcal{Z}$, and $Z_{<t} = Z_{1:t-1}$. Finite observation labels are denoted X_t. Vectors are columns. T is trajectory length; d and m are state and observation dimensions. Expectations refer to the model law specified at their use; logarithms are natural. We assume basic matrix algebra, conditional probability, and Gaussian distributions. Lecture 1 supplies the attention terminology; Lecture 4 supplies autoregression, sequence KL, and recovery bounds. We write $[d] = \{1, \dots, d\}$. Superscript asterisks* mark optional material; numbered endnotes contain short refinements.

5.1 State, observations, and the modeling choice

A state-space model specifies what information a process retains and how that information produces observations. Let $\mathcal{S}$ and $\mathcal{Z}$ be the state and observation spaces. The state is *latent* when it is

not directly observed. Latent-variable modeling is much broader than dynamical systems. In a finite mixture model, for example, each observation has an unobserved categorical component label, and its posterior label probabilities act as soft cluster assignments. Missing-data and latent-factor models use the same modeling pattern without temporal dynamics. We use the equations

$$S_0 \sim \pi_\theta, \qquad S_{t+1} = f_\theta(S_t, V_{t+1}), \qquad Z_{t+1} = g_\theta(S_{t+1}, W_{t+1}), \quad t \geq 0. \tag{5.1}$$

Here $(V_t)_{t\geq 1}$ is state noise and $(W_t)_{t\geq 1}$ is observation noise. Each sequence is independent and identically distributed (iid); the two sequences and S_0 are mutually independent. Their laws are part of the model and may depend on θ. We assume measurable update maps; Euclidean spaces and finite sets cover our examples.[1] The notation suppresses observed, time-varying inputs from the environment. They can enter the state update as $S_{t+1} = f_\theta(S_t, U_t, V_{t+1})$, and can also enter the observation map.[2]

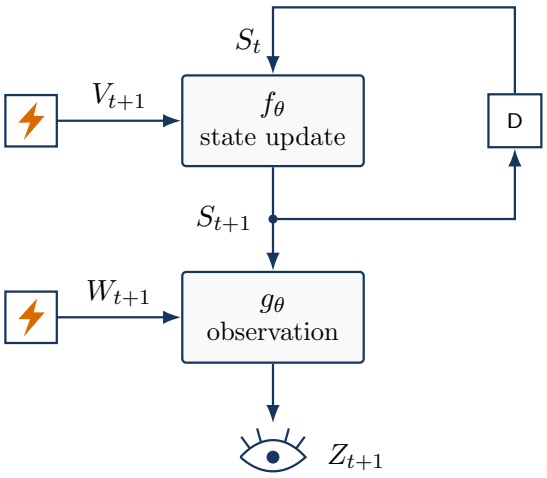


Figure 1: A state-space model. Independent noise sources enter the update and observation maps. The state feeds both the observation map and the next update. The box D is a one-step delay (and one can think of it as also providing the initial value of S_0). Function boxes apply static maps; delay boxes store values between steps. Dots mark branches of the same signal. The eye marks the observed output.

The process (S_t) is Markov: conditional on its current state, its next state does not depend on earlier states or observations. We call the full construction a *hidden Markov model* (HMM), and (Z_t) its *observation process*, or a *hidden Markov process*. The pair process $(S_t, Z_t)_{t\geq 1}$ is Markov. The observation process alone generally is not. We will say *finite HMM* when $\mathcal{S}$ is finite. This terminology also allows a continuous-state HMM.

5.1.1 Where the inductive bias enters

The useful assumption is a small or structured state space and a restricted update rule. As in Lecture 1, an architecture relates predictions at different inputs through shared structure. For physical motion, a few state coordinates can have known meanings. For language, identifying a useful compact state is itself a learning problem.

One structural choice divides the state into components intended to operate at different time scales. A slower component can influence an intermediate component, which can influence a faster component, while all three contribute to the observation.

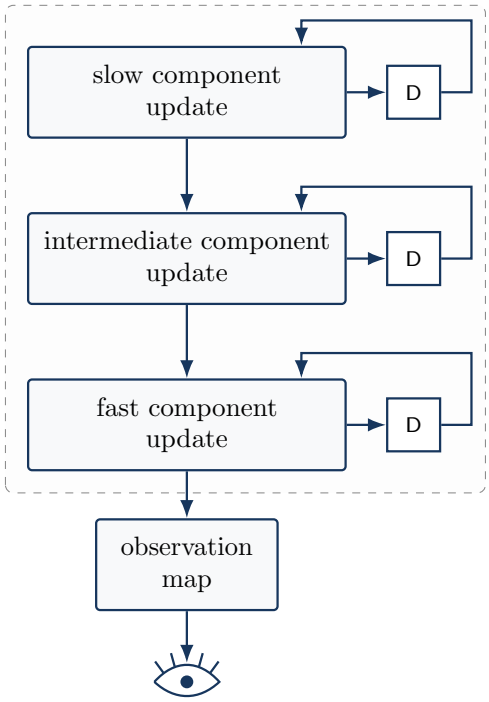


Figure 2: A structured state update with several intended time scales, drawn with the same static-map and delay semantics as Fig. 1. Each component has an explicit self-feedback loop through its own one-step delay D. The only cross-component arrows run from slow to intermediate and from intermediate to fast. Only the fast component enters the observation map. Noise inputs are omitted.

The dependence pattern alone does not make one component slower than another. That distinction requires further restrictions on rates of change, such as different update frequencies, persistence constraints, or noise scales. When these restrictions match the process, they can provide a useful inductive bias. The combined collection of components is still one state. The displayed observation map uses only the fast component. Allowing it to use all three components is a less restrictive alternative.

The representation is a modeling choice. Removing process noise gives

$$S_{t+1} = f_\theta(S_t), \qquad Z_t = g_\theta(S_t, W_t).$$

All state randomness then comes from the initial state. Deterministic dynamics can still describe elaborate processes. Even a scalar linear state update can reproduce any binary process if the random initial state encodes the whole sequence and a nonlinear observation map reads successive digits.[3] If the state and observation maps are both linear and the state is finite-dimensional, noiseless observations instead satisfy a finite recurrence with scalar coefficients; see Section 5.9.6.

Another choice is an *observation-driven model*:

$$R_{t+1} = h_\theta(R_t, Z_t), \qquad Z_t = j_\theta(R_t, W_t), \qquad R_1 \sim \nu_\theta. \tag{5.2}$$

Here R_1 is independent of the iid observation noises (W_t). For fixed parameters and a specified initial state, the observed prefix determines R_t by successive applications of h_θ. If the initial state is random and hidden, uncertainty about it may remain. During generation, random observations feed randomness into future states.

Every process on a finite or Euclidean observation space has an observation-driven representation: take $R_t = Z_{<t}$, let h_θ append the new observation, and let j_θ sample the desired conditional law using fresh uniform noise. The empty history is a fixed initial state, and the history's length records time. Inductive bias comes from restrictions such as a fixed state dimension and linear maps or a prescribed neural-network architecture. Section 5.2 gives another representation whose state is a conditional distribution. For linear Gaussian models this leads to the innovations representation in Section 5.6.3; *innovations* will mean the one-step prediction errors.

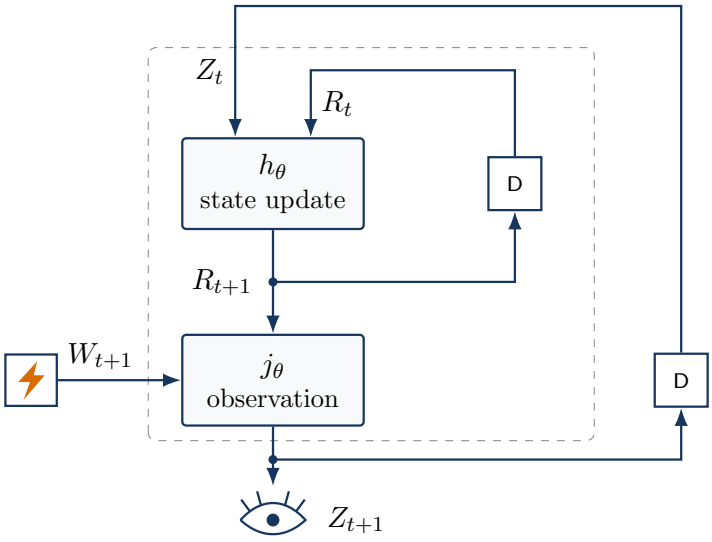


Figure 3: An observation-driven model, drawn in the same layout as Fig. 1. The update uses R_t, Z_t to produce R_{t+1}; the observation map then produces Z_{t+1}. Each D box stores its incoming signal for the next update. The dashed enclosure contains the R_t memory; observation feedback and fresh noise enter from outside.

Viewed from outside the dashed enclosure, observation feedback and fresh noise appear to produce the next observation, as in a first-order autoregressive process. The enclosure, however, contains a delay that stores R_t. The individual function boxes are static maps; their enclosure is a computation with memory. Substituting the state update gives

$$Z_{t+1} = j_\theta\big(h_\theta(R_t, Z_t), W_{t+1}\big).$$

The state R_t remains: Z_t and fresh noise alone need not suffice to generate Z_{t+1} with its conditional law given the full past. The pair (R_t, Z_t) is Markov, while the observation process need not be. Exercise 1 gives an explicit counterexample.

5.1.2 The equations define a probability law

The initial law, noise laws, and maps together determine a unique law Q_θ on state and observation sequences. A *probability kernel* assigns a probability distribution to each input state; the probability of any fixed measurable event must depend measurably on that state. The *transition kernel* is the conditional law of the next state. The *observation kernel*, also called the *emission kernel*, is the conditional law of an observation given its current state. For measurable $B \subseteq \mathcal{S}$ and $D \subseteq \mathcal{Z}$, write them as

$$a_\theta(s, B) = \mathbb{P}(f_\theta(s, V_1) \in B), \qquad b_\theta(s, D) = \mathbb{P}(g_\theta(s, W_1) \in D).$$

Their joint law has the product form

$$Q_\theta(ds_{0:T}, dz_{1:T}) = \pi_\theta(ds_0) \prod_{t=1}^{T} a_\theta(s_{t-1}, ds_t) b_\theta(s_t, dz_t). \tag{5.3}$$

For discrete variables, the kernels give conditional probability masses. When densities exist, they are taken relative to a specified measure, such as Lebesgue measure for real vectors. We write the corresponding masses or densities as $a_\theta(s' \mid s)$ and $b_\theta(z \mid s)$.

The correspondence is not one-to-one. Different randomized maps can implement the same kernels; different hidden representations can give the same observation law.

For a finite HMM, let $S_t \in [d]$ and $X_t \in [N]$, with

$$A_{ij} = Q_\theta(S_t = i \mid S_{t-1} = j), \qquad C_{aj} = Q_\theta(X_t = a \mid S_t = j).$$

Both matrices are *column-stochastic*: their entries are nonnegative and every column sums to one. This convention matches column-vector probability distributions. With $\pi_j = Q_\theta(S_0 = j)$,

$$Q_\theta(s_{0:T}, x_{1:T}) = \pi_{s_0} \prod_{t=1}^{T} A_{s_t s_{t-1}} C_{x_t s_t}. \tag{5.4}$$

Independent uniform random variables and inverse cumulative sums of the columns implement f_θ, g_θ; Exercise 2 asks for the construction. Here X_t is the discrete observation label; later $Z_t = e_{X_t}$ is its *one-hot encoding*, where e_a has entry one in position a and zero in every other position.

5.2 Generation after an observed prefix

Unconditional generation follows the equations: sample S_0, then sample fresh noises at each step. Conditional generation must first account for what the observations reveal about the hidden state.

Let

$$\mu_t^- = Q_\theta(S_t \in \cdot \mid Z_{<t} = z_{<t})$$

be the *predictive state distribution*. To sample a continuation $Z_{t:T}$, first sample $S_t \sim \mu_t^-$, then draw fresh observation noise W_t and set $Z_t = g_\theta(S_t, W_t)$. Continue with $S_{t+1} = f_\theta(S_t, V_{t+1})$ and the remaining observations, using fresh independent noises at each step.

For jointly distributed random elements Y and V, write $\mathcal{L}(Y)$ for the distribution of Y and $\mathcal{L}(Y \mid V = v)$ for the conditional distribution of Y given $V = v$. If $\mathcal{L}(Y)$ is regarded as the *prior distribution* of Y, then $\mathcal{L}(Y \mid V = v)$ is the *posterior distribution* of Y given $V = v$. For the prefix $Z_{1:t} = z_{1:t}$, the *filtering distribution* is $\mu_t = Q_\theta(S_t \in \cdot \mid Z_{1:t} = z_{1:t})$. Bayes' rule gives it and the next prediction:

$$\mu_t(ds) = \frac{b_\theta(z_t \mid s)\mu_t^-(ds)}{\int b_\theta(z_t \mid u)\mu_t^-(du)}, \qquad \mu_{t+1}^-(B) = \int a_\theta(s, B)\mu_t(ds). \tag{5.5}$$

The first formula assumes an observation density or mass and a positive denominator. Conditional laws are specified only almost surely.

For a finite HMM, let $\alpha_t(i) = Q_\theta(S_t = i \mid X_{1:t} = x_{1:t})$ and $\alpha_t^-(i) = Q_\theta(S_t = i \mid X_{<t} = x_{<t})$. Then $\alpha_0 = \pi$ and

$$\alpha_t^- = A\alpha_{t-1}, \qquad \alpha_t = \frac{\operatorname{diag}(C_{x_t,1}, \ldots, C_{x_t,d})\alpha_t^-}{\mathbf{1}^\top \operatorname{diag}(C_{x_t,1}, \ldots, C_{x_t,d})\alpha_t^-}. \tag{5.6}$$

Each step costs $O(d^2)$ for dense A. Equations Eq. (5.6) form the forward filtering recursion. Viterbi instead finds a most likely complete hidden path; smoothing computes posterior state distributions using observations on both sides of the time in question.

For continuous nonlinear models, *particle filtering* approximates the filtering distribution by weighted sampled states: propagate particles through the transition, weight them by observation likelihood, and resample when needed, drawing replacement particles with probabilities proportional to their weights. This algorithm uses evaluable observation likelihoods. *Filter forgetting* means that the effect of the initial filtering distribution diminishes as more observations arrive. It can help keep approximation errors bounded over time. Observation informativeness, dependence across time, dimension, and the particle count also matter; the bibliographical notes give precise sources. A linear Gaussian model permits exact filtering by updating means and covariances; this is *Kalman filtering*.

Filtering also gives an observation-driven representation. Take $R_t = \mu_t^-$, the entire predictive state distribution. Equation (5.5) updates it deterministically after observing z_t; the next observation is sampled by first drawing a state from this distribution and then applying the observation map. The initial R_1 is the fixed law of S_1, even when S_1 is random. For a finite HMM, the predictive law R_t is represented by a probability vector; for a Gaussian LDS, it is represented by a mean and covariance. In general, R_t is a whole probability law, whose update may require difficult integration.

5.3 What can be learned from observations?

If finite hidden states were observed, transition and emission counts would give direct estimates of A, C. For linear models with observed states, least squares estimates the linear maps under standard rank conditions. Hidden states remove these directly observed targets.

We focus on a single observed trajectory $Z_{1:T}$. With independent trajectories, log likelihoods add and the recursions restart at each initial state. These are straightforward algorithmic changes; the statistical information can be substantially different.

We use *mixing* here to mean decreasing statistical dependence between observation blocks as their time separation grows; quantitative conditions will be introduced in Lecture 6. For example, a long mixing trajectory contains repeated information about transitions but only one draw from its initial law. Even observing S_0 exactly would not consistently estimate an unrestricted π_θ from that one draw. Later observations do not supply independent initial states. Mixing and *excitation*—variation in the observations that distinguishes different parameter values—help estimate the dynamics. Lecture 6 returns to learning from dependent data.

5.3.1 Identifiability and the meaning of success

A parameter is *identifiable* if distinct values give distinct observation laws. Relabeling hidden states leaves a finite HMM's observation law unchanged. Unreachable extra states, or suitably duplicated states with identical observable behavior, can leave it unchanged too. Thus neither the state labels nor an unrestricted state count are identifiable.

An invertible change of hidden-state coordinates also preserves the observation law when the update and observation maps are transformed accordingly; Section 5.5 gives the linear case. Beyond this coordinate ambiguity, an observation law need not uniquely separate physical process noise from measurement noise.

Three useful questions are:

1. How accurately does the model predict the next observation?

2. How close are its conditional or joint observation laws, for example in KL, and its generated task statistics?

3. If the model uses the smallest state dimension needed for its predictions, how accurately can its update and observation matrices be recovered, allowing an invertible change of state coordinates?

The third can help answer the first two, but the conversion needs assumptions about sensitivity to data errors and about long-run error propagation.

5.3.2 Likelihood evaluation and optimization

The *observed likelihood* is the mass or density assigned to the observed sequence. It *marginalizes* the hidden trajectory: sums or integrates the joint law over the unobserved states. For the finite HMM,

$$Q_\theta(X_{1:T}) = \sum_{s_{0:T} \in [d]^{T+1}} \pi_{s_0} \prod_{t=1}^{T} A_{s_t s_{t-1}} C_{X_t s_t}. \tag{5.7}$$

There are d^{T+1} terms with this initialization convention. If the initial distribution is specified directly for S_1, there are d^T terms. The forward recursion evaluates the sum in $O(Td^2)$ arithmetic operations; normalization at each step or a log-domain calculation prevents numerical underflow. For continuous states, marginalization is an integral.

We fit θ by minimizing $-\log Q_\theta(Z_{1:T})$. Differentiating the finite-HMM forward algorithm is feasible and is implemented in modern software. Reverse-mode differentiation propagates derivatives backward through the likelihood computation. It has the same order of arithmetic work as likelihood evaluation, with additional storage or recomputation. The parameter optimization is generally nonconvex.

Monte Carlo methods estimate expectations and integrals with random samples. They are also used for likelihoods and gradients in nonlinear state-space models. *Sequential Monte Carlo* methods, including the particle filter above, update these samples as observations arrive.

5.3.3 EM and optimization over hidden trajectories

Expectation–maximization (EM) uses the current parameters $\theta^{(k)}$ to compute a posterior over hidden trajectories, then maximizes the expected *complete-data log likelihood*, the log joint mass or density of states and observations. The posterior expectation is the E-step; maximizing over parameters is the M-step:

$$\begin{aligned} q_k(ds) &= Q_{\theta^{(k)}}(ds_{0:T} \mid Z_{1:T}), \\ \theta^{(k+1)} &\in \arg\max_\theta \mathbb{E}_{S \sim q_k}[\log Q_\theta(S, Z_{1:T})]. \end{aligned} \tag{5.8}$$

The names E-step and M-step describe a conceptual decomposition, not necessarily two separate computational stages. The iteration itself maps $\theta^{(k)}$ to $\theta^{(k+1)}$. An implementation need only compute the posterior expectations under q_k that are required for the parameter update; it need not construct or store q_k explicitly. For a finite HMM, the *forward–backward algorithm* combines forward probability calculations from the past with backward calculations from future observations. It supplies expected transition and emission counts; these replace the observed counts. For a linear Gaussian model, the E-step uses smoothing means *and second moments*, including cross moments at adjacent times.

Exact EM makes the observed log likelihood nondecreasing.[4] Under regularity conditions its limit points are stationary: they satisfy first-order optimality conditions, with zero gradient at an interior point. Finite HMMs and linear Gaussian models admit efficient E- and M-steps. Exercise 5 asks for the monotonicity calculation, and Exercise 6 develops its density form. Exercise 8 then derives the EM update for a linear dynamics matrix under Gaussian transition noise.

Another approach treats the hidden states as optimization variables:

$$\min_{\theta, s_{0:T}} -\log Q_\theta(s_{0:T}, Z_{1:T}). \tag{5.9}$$

Minimizing Eq. (5.9) maximizes a joint mass or density, rather than summing or integrating over states. Multiplying by a prior density on θ gives a joint *maximum a posteriori* (MAP) objective: maximize the posterior density of states and parameters given the observations. Trajectory optimization is central in *simultaneous localization and mapping* (SLAM), where a robot estimates its path and a map from measurements. Its objective is often sparse: each measurement involves only a small subset of the unknown poses and map features. The joint MAP objective can remain nonconvex. For continuous states the density objective also depends on the chosen coordinates and reference measure.

For deterministic linear state dynamics and unit Gaussian observation noise, optimizing the initial state $s = S_1$ gives

$$\min_{A,C,s} \sum_{t=1}^{T} \|Z_t - CA^{t-1}s\|_2^2.$$

Products and powers of the unknown matrices make this a nonconvex parameter-fitting problem.

For the observation-driven model in Eq. (5.2), specified parameters and R_1 determine every R_t by a forward computation on the observed prefix. Likelihood evaluation also needs the conditional mass or density of $j_\theta(R_t, W_t)$. For example, $j_\theta(r, w) = Cr + w$ with Gaussian noise of covariance $\Omega \succ 0$ gives the evaluable density $\mathcal{N}(Cr, \Omega)$. A general observation-sampling map need not provide an efficient way to evaluate its density. When the state updates and log densities are differentiable, backpropagation differentiates their forward computation; Section 5.8 returns to this calculation.

5.4 Fit an observable predictor, then impose state-space structure

The proposed two-stage method first fits a flexible predictor using observed inputs and targets, then fits a compact state-space representation to that predictor. The first stage avoids hidden-state optimization. The second imposes relationships between predictions at different lags.

A finite-window model predicts from $Z_{t-p:t-1}$. If older observations have little additional predictive effect, choosing a sufficiently large p controls approximation error. This is a statement about conditional prediction.

Why fit a large, noisy model first? Many fitted coefficients can be noisy estimates of quantities determined by a few shared parameters. Enforcing those relationships can reduce estimation error. It also introduces bias when the proposed structure is wrong. Exercise 10 makes the variance reduction exact for coefficients constrained to a known low-dimensional family. The subspace algorithm below also estimates the family from data.

Long memory and filtering stability can coexist. The physical state can retain information while new observations correct errors in its estimate. For the linear models below, different matrices govern the physical state and errors in its observation-based estimate. The state may retain or amplify past perturbations even while errors in its estimate decay. Section 5.7 gives a process with permanent memory and an exact compact predictor.

5.5 Linear dynamical systems

A *linear dynamical system* (LDS) has state $S_t \in \mathbb{R}^d$, observations $Z_t \in \mathbb{R}^m$, and equations

$$S_{t+1} = AS_t + V_{t+1}, \qquad Z_{t+1} = CS_{t+1} + W_{t+1}. \tag{5.10}$$

Here $A \in \mathbb{R}^{d\times d}$, $C \in \mathbb{R}^{m\times d}$. The noise laws are specified separately; we state the Gaussian assumption where it is used. The observations are now continuous. KL still means relative entropy between probability laws; when the laws have densities, sums become integrals of log density ratios. It is not defined by treating differential entropy as ordinary discrete entropy. Appendix 5.A.1 gives the precise definition and the Gaussian formula used below.

An invertible coordinate change $S'_t = T_0^{-1}S_t$ replaces the matrices by $A' = T_0^{-1}AT_0$, $C' = CT_0$, and transforms the noise and initial laws accordingly, without changing the observation law. The relation between A and A' is called *similarity*.

The *spectral radius* $\rho(A)$ is the largest absolute value of an eigenvalue of A. We call the linear recurrence *stable* when $\rho(A) < 1$, equivalently $A^t \to 0$: initial-state perturbations decay. For a symmetric matrix Σ, $\Sigma \succ 0$ means $x^\top \Sigma x > 0$ for every nonzero x; this is *positive definiteness*. Replacing $>$ by $\geq$ defines $\Sigma \succeq 0$. We write $\lambda_{\min}(\Sigma)$ for its smallest eigenvalue.

Example 5.5.1 (Position and unobserved velocity)**.** With sampling interval $\Delta > 0$, consider

$$S_t = \begin{pmatrix} P_t \\ U_t \end{pmatrix}, \quad A = \begin{pmatrix} 1 & \Delta \\ 0 & 1 \end{pmatrix}, \quad C = (1, 0), \quad V_{t+1} = \begin{pmatrix} 0 \\ \xi_{t+1} \end{pmatrix}.$$

The model takes $P_{t+1} = P_t + \Delta U_t$, then perturbs velocity. Only position is measured: $Z_t = P_t + W_t$. Differencing measurements estimates velocity, but also differences measurement noise. For independent measurement errors of variance σ_W^2, the noise in $(Z_{t+1} - Z_t)/\Delta$ has variance $2\sigma_W^2/\Delta^2$. A state estimator can combine many measurements instead.

Both eigenvalues of A are one. State perturbations need not decay, and velocity errors accumulate in position. The process-noise covariance is *singular*: a nonzero direction has zero noise variance. Here $\Sigma_V = \operatorname{diag}(0, \sigma_\xi^2)$. This is appropriate for the discrete model just specified.[5]

Adding a small covariance εI makes a singular Gaussian transition density nonsingular and can regularize numerical calculations. This models additional state noise. Full-rank observation noise can give a nonsingular observation law even with singular state-transition covariance. If fitted variances are allowed to collapse around interpolated data, a Gaussian likelihood can become unbounded; an appropriate variance constraint or prior addresses that issue.

We will first justify the algorithm under positive-definite process and measurement covariances and stable A, which permit a short proof. The bibliographical notes describe more general identification results. *Numerical conditioning* concerns the sensitivity of a computed answer to perturbations in its inputs. Near-singular covariances require checking this sensitivity in both prediction and parameter recovery.

5.5.1 Finite HMMs also have linear vector equations

For the finite HMM, the one-hot vectors $e_{S_t} \in \mathbb{R}^d$ and $Z_t = e_{X_t} \in \mathbb{R}^N$ satisfy

$$e_{S_{t+1}} = Ae_{S_t} + V_{t+1}, \qquad Z_{t+1} = Ce_{S_{t+1}} + W_{t+1}.$$

These have the form of Eq. (5.10), with residuals satisfying

$$\mathbb{E}[V_{t+1} \mid \mathcal{G}_t] = 0, \qquad \mathbb{E}[W_{t+1} \mid \mathcal{G}_t, S_{t+1}] = 0, \qquad \mathcal{G}_t = \sigma(S_{0:t}, Z_{1:t}). \tag{5.11}$$

Here $\sigma(S_{0:t}, Z_{1:t})$ denotes the information generated by the listed variables; conditioning on it means conditioning on their history. For example, column j of A is the conditional mean of $e_{S_{t+1}}$ given $S_t = j$; subtraction defines V_{t+1}. These additive residuals generally have state-dependent, non-Gaussian laws. One-hot encoding gives linear conditional means.

More generally, for a finite-valued (*categorical*) observation,

Proposition 5.5.2 (A categorical conditional mean)**.** *With $\mathcal{F}_t = \sigma(X_{1:t})$ and $P_t = \mathbb{E}[e_{X_{t+1}} \mid \mathcal{F}_t]$,*

$$(P_t)_a = Q(X_{t+1} = a \mid X_{1:t}), \qquad Z_{t+1} = P_t + E_{t+1}, \qquad \mathbb{E}[E_{t+1} \mid \mathcal{F}_t] = 0.$$

Proof. Coordinate a of $e_{X_{t+1}}$ is $\mathbb{I}\{X_{t+1} = a\}$. Take its conditional expectation, then subtract. □

A sequence with conditional mean zero given its past is called a *martingale-difference sequence*; the residuals above have this property. Writing $P_t = \phi_t(X_{1:t})$ gives an observation-driven regression representation. The representation becomes a vector autoregression when ϕ_t is linear in a fixed collection of observed lags. Squared loss against $e_{X_{t+1}}$ is the Brier loss of Lecture 3.

5.6 The regression and subspace algorithm

We now give the complete procedure before explaining why it can work. For centered (zero-mean) real observations, choose a history length p, a block count q with $2q \le p$, and a retained state dimension $1 \le r \le qm$. An intercept (constant prediction offset) or a mean model is needed when centering is inappropriate. Vector norms below are Euclidean; matrix norms are induced operator norms unless marked F for Frobenius norm. Explicitly, $\|A\| = \sup_{\|x\|_2=1} \|Ax\|_2$ and $\|A\|_{\mathrm{F}}^2 = \sum_{i,j} A_{ij}^2$.

5.6.1 Fit lag coefficients

Use the observed lag vector

$$\Psi_t = \begin{pmatrix} Z_t \\ Z_{t-1} \\ \vdots \\ Z_{t-p+1} \end{pmatrix}, \qquad \widehat{B} = (\widehat{B}_1, \dots, \widehat{B}_p) \in \arg\min_B \sum_{t=p}^{T-1} \|Z_{t+1} - B\Psi_t\|_2^2. \tag{5.12}$$

Each $B_j \in \mathbb{R}^{m \times m}$ maps one observed lag to the next prediction; $B \in \mathbb{R}^{m \times mp}$. Thus

$$\widehat{Z}_{t+1} = \sum_{j=1}^{p} \widehat{B}_j Z_{t+1-j}. \tag{5.13}$$

This is order-p *linear autoregression*, abbreviated AR(p). The *design matrix* has rows $\Psi_t^\top$, one for each fitted target.

Proposition 5.6.1 (Observable fitting is convex)**.** *For fixed observed data, the objective in Eq. (5.12) is convex in B.*

Proof. Each prediction is linear in B; squared Euclidean distance is convex. □

Minimizing Eq. (5.12) is equivalent to conditional Gaussian maximum likelihood, conditional on the first p observations, for iid *isotropic* Gaussian residuals: their covariance is a fixed multiple of the identity. With a specified covariance $\Omega \succ 0$, the loss is the corresponding quadratic form $u^\top \Omega^{-1} u$. For unrestricted multioutput regression its minimizer is the same ordinary least-squares coefficient matrix when the design has full rank. Other conditional noise densities give other negative-log-likelihood losses.

5.6.2 Arrange, factor, and recover

Form two $qm \times qm$ *block Hankel matrices*:

$$\widehat{\mathcal{H}}_0 = [\widehat{B}_{i+j-1}]_{i,j=1}^{q}, \qquad \widehat{\mathcal{H}}_1 = [\widehat{B}_{i+j}]_{i,j=1}^{q}. \tag{5.14}$$

Each block is $m \times m$; block rows and columns index successive lags. Equal anti-diagonals use the same fitted coefficient. This indexing explains the requirement $2q \le p$.

In a *singular value decomposition* (SVD), a matrix is written as $UDV^\top$, with orthonormal columns in U, V and positive singular values on the diagonal of D. Retain the largest r singular values of $\widehat{\mathcal{H}}_0$, assumed positive, and set

$$\widehat{\mathcal{H}}_{0,r} = U_r D_r V_r^\top = \mathcal{O}\mathcal{R}, \qquad \mathcal{O} = U_r D_r^{1/2}, \quad \mathcal{R} = D_r^{1/2} V_r^\top.$$

The factors have sizes $qm \times r$ and $r \times qm$. The retained matrix $\widehat{\mathcal{H}}_{0,r}$ is the *truncated SVD*, a best rank-r approximation in operator and Frobenius norm. For a matrix with compact SVD $UDV^\top$, its *Moore–Penrose pseudoinverse* is $VD^{-1}U^\top$, denoted by a dagger; only positive singular values are included. Set

$$\overline{F} = \mathcal{O}^\dagger \widehat{\mathcal{H}}_1 \mathcal{R}^\dagger, \qquad \overline{C} = \mathcal{O}_{1:m,:}, \qquad \overline{L} = \mathcal{R}_{:,1:m}. \tag{5.15}$$

Thus $\overline{F}$ updates an r-dimensional predictive state, $\overline{C}$ reads it out, and $\overline{L}$ inserts an observation. A state-update and readout rule implementing a given predictor is called a *realization* of that predictor. This is a regression-based subspace-identification construction.

5.6.3 Estimate innovations and generate

Run the recovered predictor on observed data:

$$\overline{M}_1 = 0, \qquad \widehat{Z}_t = \overline{C}\overline{M}_t, \qquad \overline{M}_{t+1} = \overline{F}\overline{M}_t + \overline{L}Z_t. \tag{5.16}$$

The true *innovation* is the one-step prediction error $E_t = Z_t - \mathbb{E}[Z_t \mid Z_{<t}]$. The fitted residuals $\widehat{E}_t = Z_t - \widehat{Z}_t$ estimate these innovations. Use a *calibration block* $\mathcal{I}$: observations reserved for estimating the residual law, rather than fitting the lag coefficients. For its size n_{cal}, a zero-mean innovations fit uses

$$\widehat{\Omega} = \frac{1}{n_{\text{cal}}} \sum_{t \in \mathcal{I}} \widehat{E}_t \widehat{E}_t^\top. \tag{5.17}$$

Check residual means and temporal correlations; a persistent mean needs a mean model. Discarding an initial transient or fitting the initial predictive state can be appropriate. A *covariance floor* replaces eigenvalues below a chosen positive threshold by that threshold. This makes $\widehat{\Omega}$ positive definite when needed for a density. Using a separate block reduces reuse of fitting noise, although contiguous blocks from one trajectory are not independent.

Set $\overline{A} = \overline{F} + \overline{L}\overline{C}$ and generate

$$\widetilde{M}_1 = 0, \qquad \widetilde{Z}_t = \overline{C}\widetilde{M}_t + \widetilde{E}_t, \qquad \widetilde{M}_{t+1} = \overline{A}\widetilde{M}_t + \overline{L}\widetilde{E}_t, \quad \widetilde{E}_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \widehat{\Omega}). \tag{5.18}$$

Equations Eq. (5.18) define an *innovations realization*. Its state and observation perturbations share the same $\widetilde{E}_t$; they are correlated. In the ideal case it gives the same observation law as the original LDS. A process can have different state-space representations; independent state and observation noises are not needed in this representation. Substituting $\widetilde{E}_t = \widetilde{Z}_t - \overline{C}\widetilde{M}_t$ gives

$$\widetilde{M}_{t+1} = \overline{F}\widetilde{M}_t + \overline{L}\widetilde{Z}_t.$$

The displayed recursion is the observation-driven update in Fig. 3: the observation carries the noise into the next state. The variable $\widetilde{M}_t$ is an observation-based state estimate in suitable coordinates. For the ideal linear Gaussian predictor, $F = A - LC$. Thus A governs generative state dynamics and F governs the propagation of errors when filtering the same observed history. These matrices can have different stability properties.

With exact infinite-history coefficients of a constant-gain Gaussian predictor, the residuals are indeed iid Gaussian. For finite data, truncated regression and estimated matrices, assess residual Gaussianity and temporal correlations on held-out observations. Moreover, *population least squares* minimizes expected squared loss, rather than a finite-data sum. Its finite-window coefficients generally differ from the first p infinite-history coefficients. The omitted tail can correlate with the included lags. Increasing p reduces this bias under suitable filtering stability and design conditioning.

5.6.4 What should be checked?

Select p, q, r using held-out predictive performance and the intended use. Inspect small retained singular values, residual covariance, and both recovered matrices:

$$\rho(\overline{F}) \quad \text{for filtering,} \qquad \rho(\overline{A}) \quad \text{for generation.}$$

Truncated SVD alone does not guarantee either stability condition. Predictive likelihood, generated statistics, and stability diagnostics answer different questions.

The compact model ties all lag coefficients to $\overline{C}\overline{F}^{j-1}\overline{L}$. This is the parameter sharing that the unconstrained first-stage regression omits. It can reduce variance and extend a fitted finite history to a recurrent predictor.

Finite-window AR and variable-length predictors. The first-stage AR(p) fit always uses the same fixed window and learns a separate matrix for each retained lag. The recovered state model instead uses the same three matrices at every time: $M_{t+1} = \overline{F}M_t + \overline{L}Z_t$ and $\widehat{Z}_{t+1} = \overline{C}M_{t+1}$. It therefore defines one rule for prefixes of every length, while its finite-dimensional state summarizes the entire observed prefix. Equivalently, the few matrices generate the whole family of lag coefficients $\overline{C}\overline{F}^{j-1}\overline{L}$.

A causal transformer is closer to the state model than to fixed-window AR in this limited sense. Shared layer parameters define a predictor for variable-length prefixes, and attention can use every token inside the available context rather than assigning one separately learned coefficient to each fixed lag. The mechanisms remain different: a state model compresses the prefix recursively into a fixed-dimensional state, whereas standard attention retains token-level keys and values and revisits them. Its memory and computation grow with the context, and using lengths beyond the architecture's or training distribution's range is not automatic. The later discussion of recurrent attention asks a separate computational question: when can an attention calculation itself be reorganized as a recurrence or parallel scan?

5.7 A controlled example of the memory advantage

A state model can build a state that maintains information from arbitrarily distant past, whereas a fixed observation window discards past information. Consider the scalar Gaussian process

$$\begin{gathered} U \sim \mathcal{N}(0, \tau^2), \qquad S_0 = U, \\ S_{t+1} = S_t, \qquad Z_{t+1} = S_{t+1} + W_{t+1}, \quad t \geq 0, \\ W_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2). \end{gathered} \tag{5.19}$$

with U independent of the noises and $\tau, \sigma > 0$. The state in this process keeps the initial data indefinitely. Suppose the model and its parameters are known and the goal is to predict Z_{n+1} after observing $Z_{1:n}$. The minimum-mean-squared-error predictor is the posterior mean $\mathbb{E}[S_n \mid Z_{1:n}]$. The posterior mean and variance after n observations are

$$m_n = \frac{\sum_{i=1}^n Z_i}{\sigma^2/\tau^2 + n}, \qquad v_n = \frac{\tau^2\sigma^2}{\sigma^2 + n\tau^2}. \tag{5.20}$$

Letting $H_n = \sum_{i=1}^n Z_i$, we obtain m_n from the updates $H_{t+1} = H_t + Z_{t+1}$ and $C_{t+1} = C_t + 1$, initialized with $H_0 = 0$ and $C_0 = \sigma^2/\tau^2$, and return $m_n = H_n/C_n$. Thus the optimal predictor stores two scalars rather than the full history.

Proposition 5.7.1 (Full history versus a fixed window)**.** *For $n \geq p$, the minimum mean squared prediction error for Z_{n+1} using all n observations is σ^2+v_n. Using only the most recent p observations gives minimum error σ^2+v_p, even among nonlinear predictors. The expected KL from the full-history predictive law to the optimal window predictive law is*

$$\frac{1}{2}\log\frac{\sigma^2 + v_p}{\sigma^2 + v_n}.$$

Proof. Gaussian conditioning gives Eq. (5.20). The conditional mean minimizes squared loss; the fresh measurement noise adds σ^2. Any p observations give the same posterior variance v_p, because their joint law is unchanged by reordering them. Let $a = \sigma^2 + v_n$, $b = \sigma^2 + v_p$, and let m_p denote the posterior mean based on the last p observations. Conditional-expectation orthogonality gives

$$\mathbb{E}[(m_n - m_p)^2] = \mathbb{E}[(Z_{n+1} - m_p)^2] - \mathbb{E}[(Z_{n+1} - m_n)^2] = b - a.$$

Applying the Gaussian KL formula in Eq. (5.36) conditionally and averaging therefore leaves only $\frac{1}{2}\log(b/a)$. □

For $\tau = \sigma = 1$, $p = 10$, and $n = 1000$, the excess prediction errors above irreducible noise are $1/11$ and $1/1001$. This comparison uses known model parameters and isolates the effect of history length. Exercise 14 asks you to derive the posterior and prediction formulas and to explain which additional data identify τ^2.

5.7.1 A finite-data comparison of the actual algorithm

A complementary experiment fits both models to the same single training trajectory. The scalar innovations generator is

$$M_{t+1} = 0.98M_t + 0.18E_t, \qquad Z_t = M_t + E_t, \qquad E_t \overset{\text{iid}}{\sim} \mathcal{N}(0,1),$$

started at $M_0 = 0$ and warmed up before the retained data. Its predictor has $F = 0.8$ and coefficients $K_j = 0.18(0.8)^{j-1}$. The lag-Hankel matrix has rank one.

We compare ordinary AR(32) with its rank-one subspace fit ($q = 16, r = 1$); rank and history length are supplied, not selected using test data. Both covariance estimates use a separate 2,000-step calibration continuation. A further 10,000-step continuation evaluates conditional-mean error and Gaussian conditional KL against the exact innovations predictor. Repeating over 50 fixed random seeds measures variation between training trajectories. Results, including generation instability, appear in Table 1. The reproducible script is `scripts/lecture05_synthetic.py`.

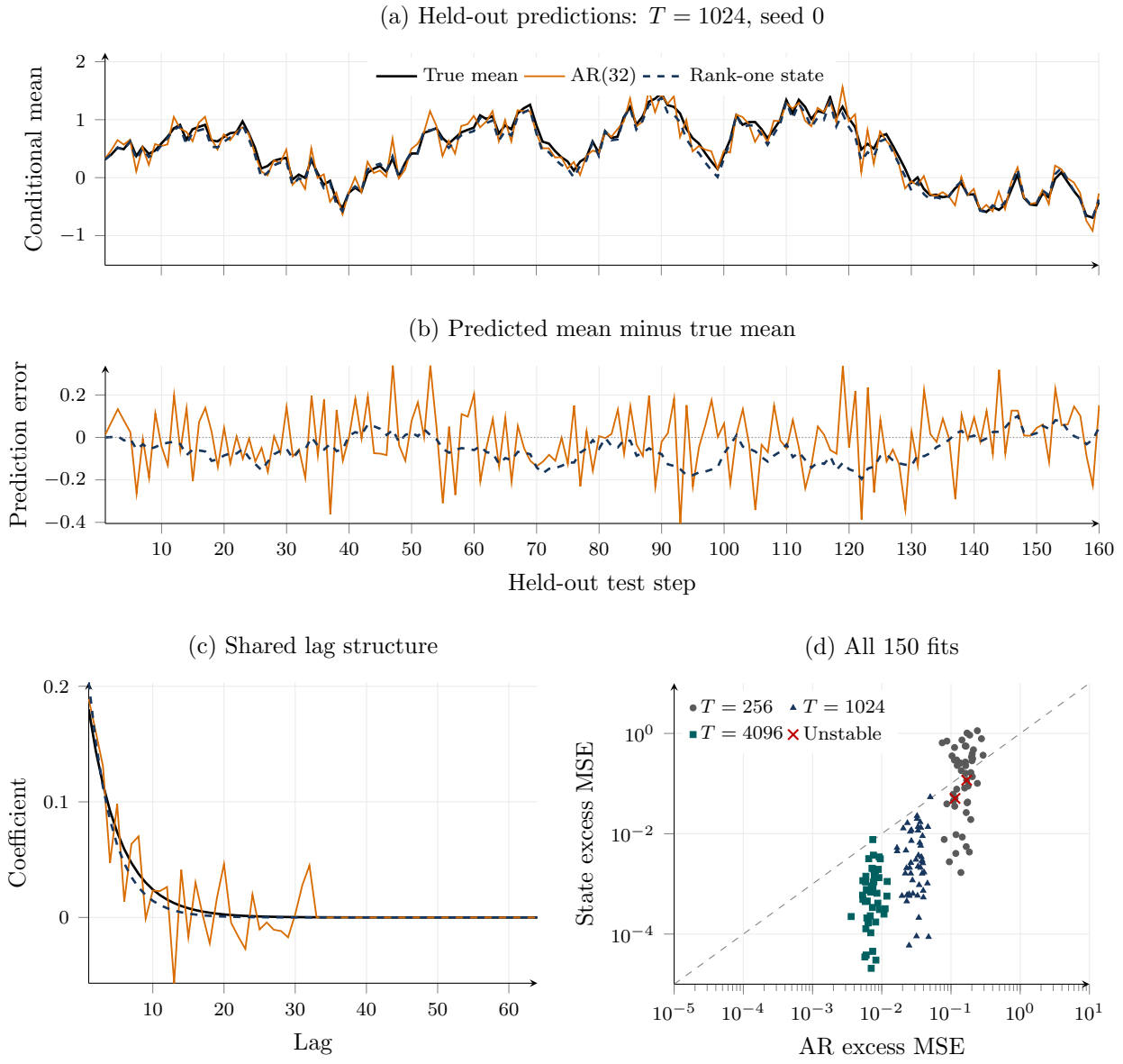


Figure 4: The experiment behind Table 1. (a,b) One-step conditional-mean predictions on the first 160 test steps for the fixed seed 0 and $T = 1024$. Both fitted predictors condition on the same observed history; the true conditional mean is known from the generator. The errors exclude the fresh observation noise. (c) The same fit's lag coefficients: AR(32) is zero beyond lag 32, whereas the recovered state ties all lags to $\overline{C}\overline{F}^{j-1}\overline{L}$. (d) Every seed at all three training sizes. Points below the diagonal favor the state predictor. Crosses mark unstable state-space generators; they are included in the table and plot.

Training T	Model	Excess MSE	KL/step	$\widehat{\Omega}$	Unstable / 50
256	AR(32)	0.1564	0.0727	1.1696	0
256	Rank-one state	0.2693	0.1091	1.2904	2
1024	AR(32)	0.0316	0.0158	1.0371	0
1024	Rank-one state	0.0061	0.0033	1.0104	0
4096	AR(32)	0.0074	0.0039	1.0027	0
4096	Rank-one state	0.0011	0.0008	0.9961	0

Table 1: Means over 50 trajectories. Excess MSE compares the fitted conditional mean with the true mean; irreducible variance is one. Both models estimate $\widehat{\Omega}$ on the calibration block; the true innovation variance is one. KL uses each fitted variance. Unstable means generative spectral radius at least one; such fits are not discarded.

Figure 4 shows one-step conditional-mean predictions and errors, the fitted lag coefficients, and every run's prediction risk. The trajectory is the first test segment from seed 0 at $T = 1024$, selected independently of its performance. Both fitted predictors make one-step conditional predictions using the same observed history. For either predictor, let $\widehat{q}(\cdot \mid Z_{<t})$ be its Gaussian conditional density, with predicted mean $\widehat{\mu}_t$ and calibrated variance $\widehat{\Omega}$. Its per-step test log-loss is

$$-\log \widehat{q}(Z_t \mid Z_{<t}) = \frac{1}{2}\log(2\pi\widehat{\Omega}) + \frac{(Z_t - \widehat{\mu}_t)^2}{2\widehat{\Omega}}.$$

With a fixed positive variance, minimizing this loss over mean predictors is equivalent to minimizing squared error. The AR least-squares fit estimates the mean coefficients; a residual-variance estimate completes the probability model used for scoring and generation. Here both models estimate $\widehat{\Omega}$ on the calibration block. The ideal variance is one for the exact state predictor and approximately $1 + 1.57 \times 10^{-7}$ for AR(32) with coefficients minimizing expected squared error in the long-run limit; see Appendix 5.B. Thus one is the reference value for both methods in Table 1. For either fitted predictor, the data-generating law gives

$$\mathbb{E}[(Z_t - \widehat{\mu}_t)^2] = 1 + \mathbb{E}[(M_t - \widehat{\mu}_t)^2].$$

Calibration includes mean-prediction error; a finite estimate can also fall below one.

The script regenerates the table, plots, and JSON results; the saved results include fitted coefficients, the plotted data, seeds, software versions, and source hashes. Appendix 5.B specifies the data generation, fitting, calibration, evaluation metrics, and construction of all four panels.

At $T = 1024$ and 4096, imposing rank one substantially reduces both mean-prediction error and conditional KL. At $T = 256$, the subspace fit is worse on average and produces two unstable generators. All fitted predictive recursions are stable in this run. The example therefore exhibits both a gain from parameter sharing and a small-sample failure of unconstrained realization.

The comparison is conditional on this low-rank data-generating model. It measures finite-data estimation, whereas Proposition 5.7.1 isolates information lost by a fixed window.

5.7.2 Completing an observed prefix

The two completion experiments answer different questions. The first asks what the two predictors fitted in Section 5.7.1 generate; it includes parameter-estimation and realization error. The second supplies the true constant-state law and asks only what information is lost by restricting prediction to a fixed window. In both experiments, generation starts from an observed prefix and then feeds each model's generated observations back into that model.

Figure 5 uses the same seed 0, $T = 1024$ fits as Fig. 4, starting immediately before its first test observation. The fitted state model has $\overline{A} = \overline{F} + \overline{LC} \approx 0.949$, compared with the true $A = 0.98$. Its conditional mean loses half its magnitude in about 13 steps, compared with 34 for the true process. The AR model's dominant eigenvalue is approximately 0.978. In this run the state fit has smaller one-step prediction error, but its completions lose the prompt's influence too quickly and have too little long-horizon variation. At step 300, the conditional standard deviations are approximately 1.35 for the truth, 1.29 for AR, and 1.20 for the state fit.

The constant-hidden-state process in Eq. (5.19) isolates memory without parameter-estimation error. Take $\tau = \sigma = 1$ and supply a 1,000-observation prompt centered near 2. The full-history model has posterior mean m_n and variance v_n from Eq. (5.20). Its continuation has conditional mean m_n at every future step: the inferred level persists. The resulting continuation law is exactly the true conditional process. Its next-observation variance is $1 + v_n = 1 + 1/1001$, including the remaining uncertainty about the hidden level. The optimal ten-observation predictor instead has residual variance $1 + v_{10} = 12/11$. Sampling from it recursively gives

$$\widetilde{Z}_{t+1} = \frac{1}{11}\sum_{j=0}^{9} \widetilde{Z}_{t-j} + \sqrt{\frac{12}{11}}\,\xi_{t+1}, \qquad \xi_{t+1} \overset{\text{iid}}{\sim} \mathcal{N}(0,1),$$

initialized with the prompt's final ten values. Its coefficients sum to $10/11 < 1$, so its conditional mean eventually returns to the population mean zero. Figure 6 shows the resulting loss of the prompt-specific level. An AR model can propagate information through generated observations for many more steps than its window length; here that influence decays, while the constant-state model retains it.

Corresponding paths in both figures share every fresh standard Gaussian draw. In Fig. 5, this noise drives large fluctuations, yet the paths remain strikingly close: the models respond similarly to it even after the prompt's influence decays. Appendix 5.B.4 gives the code, prompts, noise draws, and protocol.

The figures support two separate conclusions. In Fig. 5, the fitted state model has better one-step prediction but a worse estimate of generative persistence; fitting error can reverse the ranking when the evaluation target changes. In Fig. 6, all probability laws are known. The full-history conditional law is then Bayes optimal, and fixed-window AR has strict approximation error because it discards observations that contain information about U. This error is modest for one-step prediction but qualitatively important for long continuations, where the inferred level is eventually forgotten.

Learning the constant-state law would create a third comparison, not shown in the figure. With many independent training trajectories, each with an independent U, consistent estimators converge to the two displayed predictors. With finitely many trajectories, their approximation error must be compared with parameter-estimation and algorithmic error. One long trajectory does not provide repeated draws from the distribution of U.

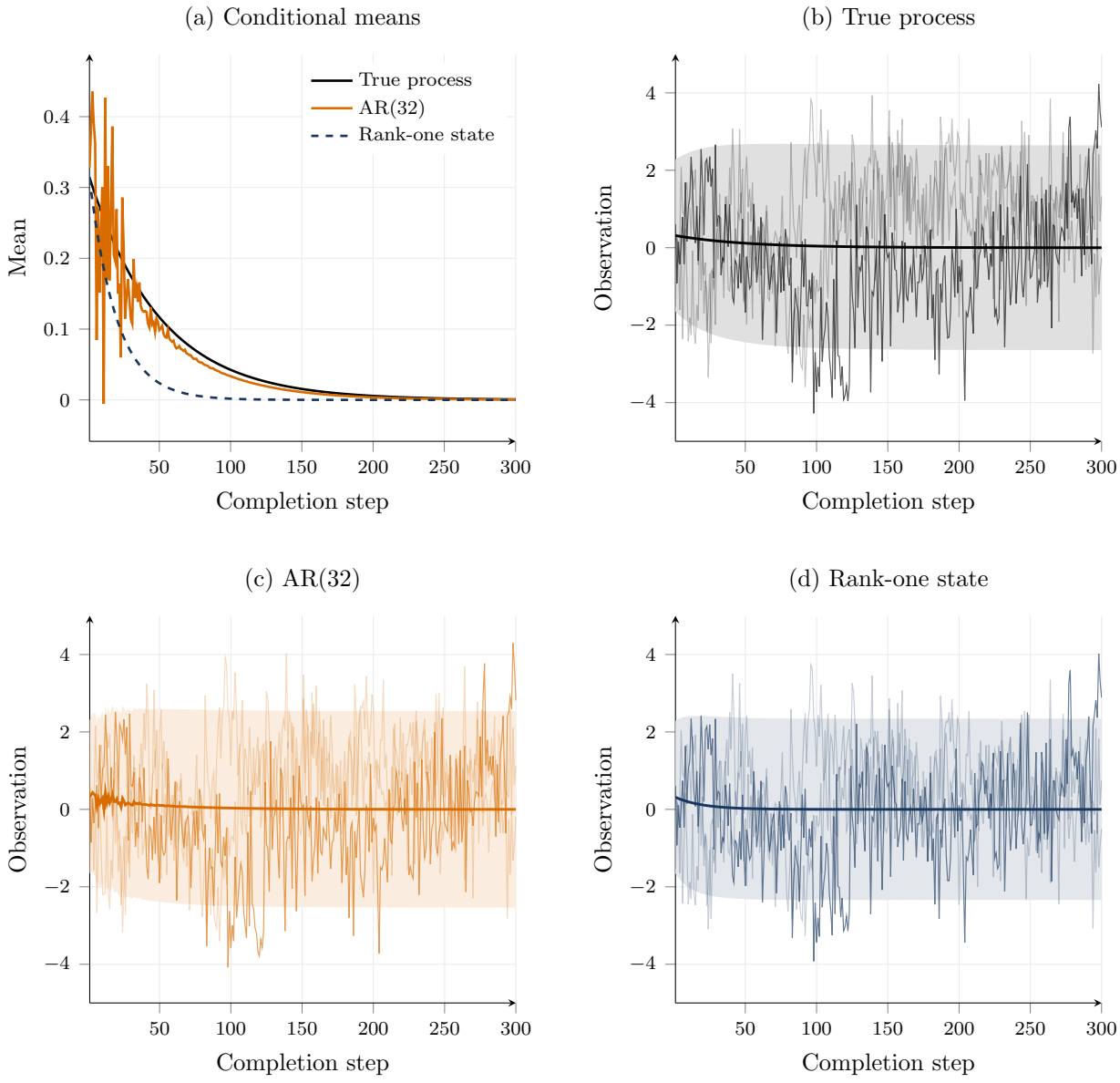


Figure 5: Completions of the scalar Gaussian process $M_{t+1} = 0.98M_t + 0.18E_t$, $Z_t = M_t + E_t$, $E_t \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. AR(32) and the rank-one state model are the seed 0, $T = 1024$ fits from Fig. 4; all models receive the same preceding prefix. (a) Exact conditional means given this prefix. (b–d) Three generated continuations per model; the thick line is the conditional mean and shading is the pointwise 95% prediction interval. Corresponding paths share the same standard Gaussian draw at every future step, scaled by each model's innovation standard deviation. Despite large noise-driven fluctuations, the paths remain strikingly close because the models respond similarly to the shared noise. See Appendix 5.B.4 for the protocol.

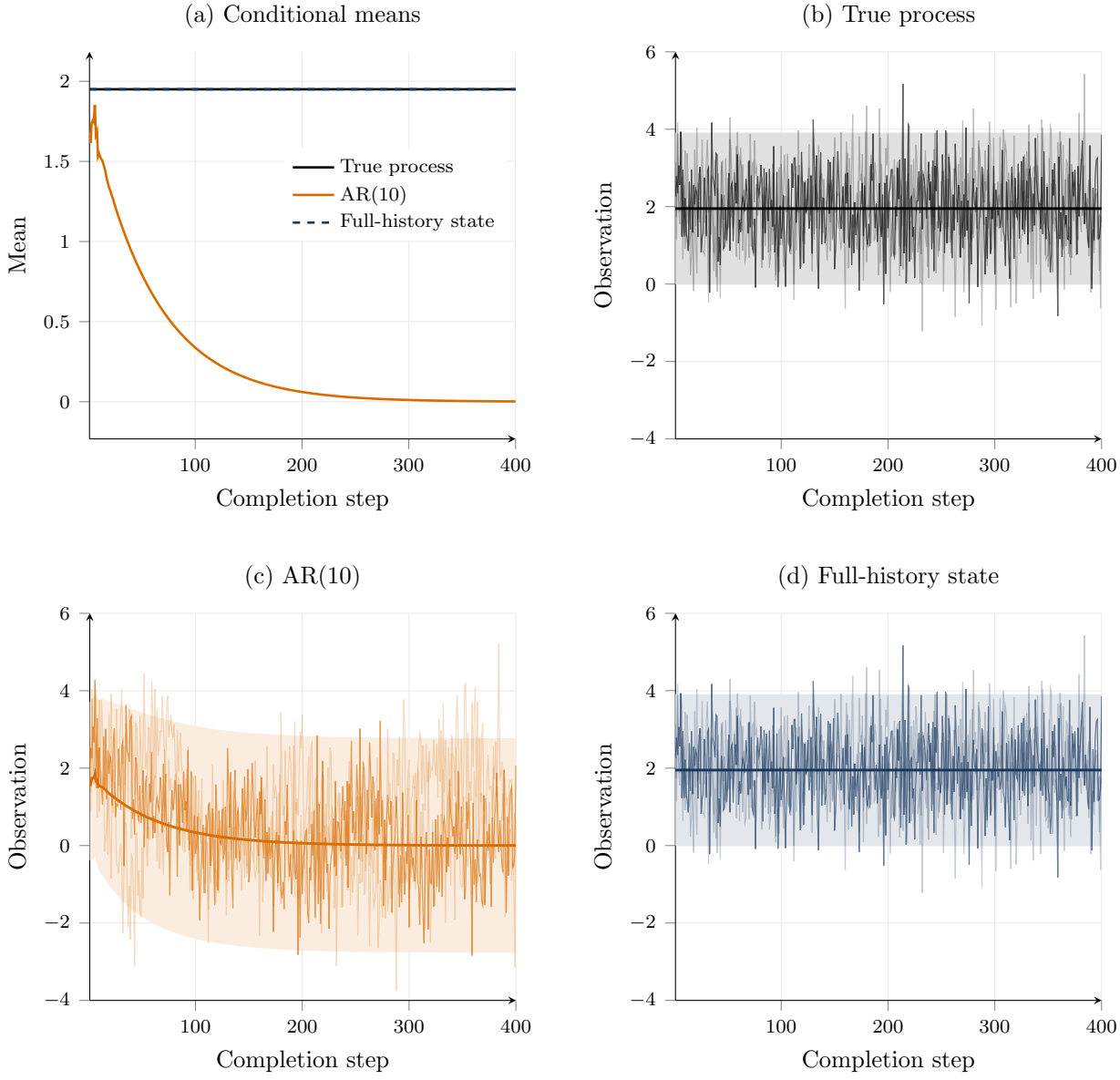


Figure 6: Memory in completions of the constant-hidden-state process $Z_t = U + W_t$, where $U \sim \mathcal{N}(0,1)$ and $W_t \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ are independent. The prescribed 1,000-observation prompt is centered near 2. The panels match Fig. 5: (a) conditional means; (b–d) three continuations, conditional means, and pointwise 95% prediction intervals, all given the original prompt. The full-history state model gives the true conditional law, so their mean curves coincide in (a), and (b,d) have identical paths and bands. AR(10) recursively uses the optimal ten-observation conditional law. Corresponding paths share standard Gaussian draws at every future step. All parameters are known. See Appendix 5.B.4.

What does "simpler" mean? Suppose the constant-state structure in Eq. (5.19) is supplied to the learner. A direct state-model fit then estimates two variances, or an equivalent tied prediction rule. Unrestricted AR(10) estimates ten regression coefficients separately and, for a probabilistic model, a residual variance. For a fixed number of independent training trajectories, our prior expectation is therefore that the correctly specified state fit has the statistical advantage: its parameter sharing reduces estimation variance. This expectation does not follow from the number of time points alone. The distribution of U is identified by variation between trajectories, so one long trajectory is not equivalent to many independent draws of U.

AR is simpler in a different sense. Least squares is a direct convex calculation, whereas fitting a latent model can require nonconvex optimization or a numerically delicate realization step. AR also imposes less of the particular constant-state structure, which can help when that structure is wrong. The rank-one model in Fig. 4 has fewer final parameters than AR(32), yet the generic subspace procedure first estimates AR coefficients and then performs an SVD-based realization; at $T = 256$, that procedure is less reliable. Representation size, optimization difficulty, conditioning, structural commitment, and prediction quality are distinct notions of simplicity. They need not rank two methods in the same order.

Minimum description length (MDL) does not reduce this comparison to counting parameters. It requires a specified class of candidate codes, including how models or parameters and the data are encoded, and it selects the shortest resulting description [Barron et al., 1998]. Familiar parameter-count penalties are special asymptotic approximations for regular models. Latent-variable models are often singular, where such approximations can fail [Watanabe, 2013]. A fully specified MDL procedure may prefer either model; its answer depends on the candidates, code, data, and prediction target. The examples here contradict the slogan that small samples automatically favor the model with fewer parameters. They do not contradict MDL as a precisely specified model-selection method.

5.8 Recurrent networks, backpropagation, and attention

A conventional recurrent predictor computes

$$R_{t+1} = h_\theta(R_t, u_t), \qquad q_\theta(\cdot \mid u_{1:t}) = o_\theta(R_{t+1}).$$

The inputs u_t may be external measurements or previous tokens. With *teacher forcing*, the supplied inputs are the observed past tokens rather than sampled model outputs. The recurrent states are deterministic functions of them, so the output likelihood can be evaluated directly.

Backpropagation through time differentiates this unrolled computation. For a per-step update costing c, its work is $O(Tc)$, up to constant factors for the derivatives; straightforward storage is linear in T times the size of the intermediate values saved per step. *Checkpointing* saves only selected intermediate values and recomputes the others during differentiation. The state-update *Jacobian* is its matrix of partial derivatives with respect to the incoming state. Products of these matrices can make gradients small or large. *Long short-term memory* (LSTM) networks use learned gates to control the retention and updating of state values; gradient propagation depends on the resulting Jacobian products.

Linearity permits additional computational structure. For affine updates with known per-position coefficients,

$$R_{t+1} = A_t R_t + b_t, \qquad (A_2, b_2) \circ (A_1, b_1) = (A_2 A_1, A_2 b_1 + b_2),$$

composition is associative. A *parallel scan* computes all prefix compositions by combining adjacent blocks in parallel. It can reduce sequential depth, the number of operations that must wait for one another. Dense matrix compositions may cost more than a sequential matrix–vector recurrence;

diagonal or other structured transitions make the scan useful. Learning the matrices end to end can still be nonconvex. The coefficients must be computable without first evaluating the unknown preceding recurrent state. Exercise 15 compares these operations.

Observed inputs can also be added to an LDS: $S_{t+1} = AS_t + Bu_t + V_{t+1}$. Lagged inputs become additional regression features. Inputs chosen independently of unobserved system noises are simpler to analyze than inputs correlated with those noises, for example through feedback. Sufficient excitation still matters.

5.8.1 An exact recurrent evaluation of linear attention

Let key and query features $k_t, q_t \in \mathbb{R}^r$, values $v_t \in \mathbb{R}^s$, and gates $\gamma_t \in \mathbb{R}$ be computed from a layer's inputs. The matrix state

$$R_0 = 0, \qquad R_t = \gamma_t R_{t-1} + v_t k_t^\top, \qquad y_t = R_t q_t \tag{5.21}$$

has sr coordinates.

Proposition 5.8.1 (Recurrent and pairwise evaluations)**.** *The same output is*

$$y_t = \sum_{u=1}^{t} v_u (k_u^\top q_t) \prod_{j=u+1}^{t} \gamma_j.$$

Proof. Unroll the affine recursion for R_t and multiply by q_t. The empty product equals one. □

For $\gamma_t = 1$, this is unnormalized *causal linear attention*: it uses only current and past values, with weights $k_u^\top q_t$. If these weights are nonnegative, a normalized version maintains $d_t = d_{t-1} + k_t$ and uses $R_t q_t / (d_t^\top q_t)$, when the denominator is positive. The state and recurrent work per token are $O(sr)$, in addition to feature computation. Directly evaluating all pairwise interactions has quadratic dependence on sequence length.

This identity concerns a finite-feature attention map. Unrestricted softmax attention need not admit such an exact fixed-size summary.[6]

5.8.2 What this says about language models

One can train recurrent-compatible layers from the start, or convert a trained transformer using *distillation*: train the new model to match the original model's predictions or intermediate computations. Mamba-2 develops structured state-space/attention computations with efficient blocked implementations. MOHAWK matches attention-weight matrices, intermediate representations, and predictions in stages. Mamba-in-Llama reuses attention projections and trains hybrid models; retained attention layers still store past keys and values in growing caches.

Hazan and collaborators' spectral filtering takes another route: fit combinations of fixed temporal filters—weighted sums of past observations—whose span approximates a specified family of dynamical predictors. It can avoid identifying a physical transition matrix. Spectral state-space sequence layers extend this idea, and observation spectral filtering addresses prediction from outputs. These methods should be distinguished from recovering (F, C, L) by the Hankel procedure.

A small recurrent state offers economical sequential inference. It must also support the information retrieval, optimization, numerical stability, and accelerator utilization needed by the task. Training-time auxiliary objectives offer another use of state structure: they can require hidden representations to predict future representations at several time scales and then discard the auxiliary hierarchy for generation. This imposes partial state-like constraints while retaining an ordinary

autoregressive generator; the bibliographical notes describe a recent example. The cited papers demonstrate particular constructions and experiments; broad architectural dominance remains an empirical question. The possibility of stronger state-space models is a research motivation.

5.9 Analysis of the autoregression–subspace method*

The following results justify the construction under explicit assumptions. They may be read after the algorithm and classroom discussion. For a random vector Y, its *root mean square* (RMS) norm is $(\mathbb{E}[\|Y\|_2^2])^{1/2}$, also called its L^2 norm.

5.9.1 Gaussian prediction and truncation

For a Gaussian LDS, the conditional distribution can be computed by a recursive predictor. The predictor maintains a state estimate obtained from the observations. Unrolling the recursion gives an observable autoregression. The tail of its history coefficients controls the truncation error.

The observations in this subsection are real vectors. Log loss uses their probability densities, and the probability and KL chain rules apply to these conditional densities.

Theorem 5.9.1 (A constant-gain Gaussian predictor)**.** *Initialize the system at time* 1 *for this theorem. Consider* $S_{t+1} = AS_t + V_{t+1}$, $Z_t = CS_t + W_t$ *for* $t \geq 1$, *with spectral radius* $\rho(A) < 1$, *independent centered Gaussian process and observation noises with covariances* $\Sigma_V \succ 0$ *and* $\Sigma_W \succ 0$. *There are matrices* $\Pi \succ 0$ *and* L *such that, when* $S_1 \sim \mathcal{N}(0, \Pi)$ *independently of the noises, the recursion*

$$M_1 = 0, \qquad M_{t+1} = FM_t + LZ_t, \qquad F = A - LC \tag{5.22}$$

satisfies

$$M_t = \mathbb{E}[S_t \mid Z_{<t}], \qquad Z_t \mid Z_{<t} \sim \mathcal{N}(CM_t, \Omega), \qquad \Omega = C\Pi C^\top + \Sigma_W. \tag{5.23}$$

Moreover, $\rho(F) < 1$ *and the observations have uniformly bounded second moments.*

The recursion is the *Kalman predictor*. The covariance Π is chosen to make the *gain* L, which converts observation error into a state correction, constant over time; Appendix 5.A.2 constructs it and proves the theorem. For an arbitrary Gaussian initialization, the same conditional calculation gives time-dependent gains. Positive noise covariance means positive variance in every nonzero direction. A stable A permits partially observed states and arbitrary C in this result.

Define the *innovations* $E_t = Z_t - CM_t$. Their conditional law is always $\mathcal{N}(0, \Omega)$, so they are independent across time and independent of the previous observations. Consequently, the same observation law can be generated by

$$Z_t = CM_t + E_t, \qquad M_{t+1} = (F + LC)M_t + LE_t = AM_t + LE_t. \tag{5.24}$$

The conditional predictor and its innovations law together specify generation.

Unrolling Eq. (5.22) gives

$$CM_t = \sum_{j=1}^{t-1} K_j Z_{t-j}, \qquad K_j = CF^{j-1}L. \tag{5.25}$$

Because F is stable, these coefficients decay geometrically. The following proposition translates their tail into a sequence-law guarantee.

Proposition 5.9.2 (Truncation controls prediction and sequence KL)**.** *In the setting above, suppose* $\|K_j\| \le G\gamma^{j-1}$ *for* $j \ge 1$*, with* $0 < \gamma < 1$*, and* $\sup_t(\mathbb{E}[\|Z_t\|_2^2])^{1/2} \le B_Z$*. Define*

$$\widehat{Z}_t^{(p)} = \sum_{j=1}^{\min(p,t-1)} K_j Z_{t-j}.$$

Then

$$\left(\mathbb{E}[\|CM_t - \widehat{Z}_t^{(p)}\|_2^2]\right)^{1/2} \le a_p := \frac{GB_Z\gamma^p}{1-\gamma}. \tag{5.26}$$

Let $P_{1:T}$ *be the true observation law and let* $Q_{1:T}^{(p)}$ *use conditional law* $\mathcal{N}(\widehat{Z}_t^{(p)}, \Omega)$*. Then*

$$D(P_{1:T}, Q_{1:T}^{(p)}) \le \frac{Ta_p^2}{2\lambda_{\min}(\Omega)}. \tag{5.27}$$

Proof. The triangle inequality in L^2 bounds the omitted tail by $B_Z \sum_{j>p} G\gamma^{j-1} = a_p$. For Gaussians with the same covariance,

$$D(\mathcal{N}(u,\Omega), \mathcal{N}(v,\Omega)) = \tfrac{1}{2}(u-v)^\top \Omega^{-1}(u-v).$$

Apply this at each observed prefix and use the KL chain rule. □

This explains why an observable autoregression can approximate a noisy state-space model well: the lost remote history has an explicit cost. These bounds use the true coefficients. Regression contributes coefficient-estimation error in addition to this truncation error. Squared loss is convenient for the convex fit and the Gaussian calculation; the state-elimination identity itself is independent of the fitting loss.

5.9.2 Exact recovery from ideal coefficients

The coefficients in Eq. (5.25) come from a small recurrent computation. Their matrix structure reveals that computation. Suppose $K_j = C_\star F_\star^{j-1} L_\star$ has minimal state dimension $d_\star$: no smaller state can realize the same sequence of coefficients. Choose a block count q for which

$$\mathcal{O}_\star = \begin{pmatrix} C_\star \\ C_\star F_\star \\ \vdots \\ C_\star F_\star^{q-1} \end{pmatrix}, \qquad \mathcal{R}_\star = (L_\star, F_\star L_\star, \dots, F_\star^{q-1} L_\star)$$

both have rank $d_\star$. The first maps a state to successive readouts; the second collects how successive observations enter the state. Form the $qm \times qm$ block matrices

$$\mathcal{H}_0 = [K_{i+j-1}]_{i,j=1}^q = \mathcal{O}_\star \mathcal{R}_\star, \qquad \mathcal{H}_1 = [K_{i+j}]_{i,j=1}^q = \mathcal{O}_\star F_\star \mathcal{R}_\star. \tag{5.28}$$

Each block is $m \times m$; block rows and columns index successive lags. A matrix constant along block anti-diagonals is a *block Hankel matrix*. Here $\operatorname{rank}(\mathcal{H}_0) = d_\star$.

Take its rank-$d_\star$ singular value decomposition and factor it:

$$\mathcal{H}_0 = UDV^\top = \mathcal{O}\mathcal{R}, \qquad \mathcal{O} = UD^{1/2}, \quad \mathcal{R} = D^{1/2}V^\top.$$

Apply Eq. (5.15) with the exact matrix $\mathcal{H}_1$ and these factors. Then the matrices satisfy the following recovery property.

Proposition 5.9.3 (Recovery of the prediction map)**.** *The recovered matrices satisfy*

$$\overline{C}\,\overline{F}^{j-1}\overline{L} = K_j \quad \textit{for every } j \geq 1.$$

Thus Eq. (5.16) computes the same predictor on every observed history.

The proof in Appendix 5.A.3 shows that the recovered states differ from minimal predictive coordinates by an invertible change of basis. Their role is to implement prediction; recovering the original physical coordinates is optional.

5.9.3 What finite-data guarantees must control

A finite-data analysis must control four errors: finite-history approximation, estimation of the regression coefficients, matrix realization, and estimation of the innovations law.

The regression error has a useful exact decomposition. Write

$$Z_{t+1} = B^{[p]}\Psi_t + r_t + E_{t+1}, \qquad B^{[p]} = (K_1, \dots, K_p),$$

where r_t is the omitted predictor tail and E_{t+1} is the innovation. If $\sum_t \Psi_t\Psi_t^\top$ is invertible, ordinary least squares gives

$$\widehat{B} - B^{[p]} = \left(\sum_t E_{t+1}\Psi_t^\top + \sum_t r_t\Psi_t^\top\right)\left(\sum_t \Psi_t\Psi_t^\top\right)^{-1}. \tag{5.29}$$

The first numerator sums conditionally mean-zero terms; it is a martingale sum. The second is truncation error correlated with the regressors. The matrix $\sum_t \Psi_t\Psi_t^\top$ is the design's *Gram matrix*. Its inverse expresses *excitation*: insufficient variation in some direction makes that direction hard to estimate. Dependence between overlapping windows must be handled in the probability bounds.

If every coefficient used in a q-block Hankel matrix has operator error at most δ, its matrix error is at most $q\delta$. Indeed, each output block has norm at most $\delta\sum_j \|x_j\|$, and Cauchy–Schwarz bounds the full norm. If the true Hankel matrix has rank r, the best rank-r approximation to its estimate obeys

$$\|\widehat{\mathcal{H}}_{0,r} - \mathcal{H}_0\| \leq 2\|\widehat{\mathcal{H}}_0 - \mathcal{H}_0\|. \tag{5.30}$$

This follows because the discarded singular value is no larger than the perturbation norm. Converting this to matrices requires controlling $\sigma_r(\mathcal{H}_0)$ and the pseudoinverses. Prediction-risk bounds also require the conversions in the next subsection.

There are finite-sample theorems for related stochastic subspace algorithms, including single-trajectory recovery up to similarity at order $T^{-1/2}$ with logarithmic and system-dependent factors. Some allow $\rho(A) \leq 1$. Their assumptions include *observability*: different noiseless initial states can be distinguished by their observations, equivalently $(C^\top, (CA)^\top, \dots, (CA^{d-1})^\top)^\top$ has rank d. They also require *controllability by the process noise*: $(\Sigma_V^{1/2}, A\Sigma_V^{1/2}, \dots, A^{d-1}\Sigma_V^{1/2})$ has rank d, so process noise can affect every state direction over time. Other assumptions include positive measurement covariance, known state dimension, and a *steady-state filtering initialization*, whose prediction-error covariance is fixed by the covariance recursion. Their regression predicts a block of future observations from a block of past observations, rather than the single next observation used here. The bibliographical notes give the exact source and theorem locations. A corresponding rate for the algorithm here requires analysis of Eq. (5.29) and its realization step.

5.9.4 From matrix errors to prediction and sequence laws

The following calculation makes a useful part of the conversion explicit. Compare a true and fitted predictor on the *same observed history*, in aligned state coordinates:

$$M_{t+1} = FM_t + LZ_t, \qquad \overline{M}_{t+1} = \overline{F}\,\overline{M}_t + \overline{L}Z_t, \qquad M_1 = \overline{M}_1 = 0.$$

Suppose $\|\overline{F}\| \le \rho < 1$, the true states and observations have RMS norms bounded by B_M, B_Z, and $\|\overline{F}-F\| \le \delta_F$, $\|\overline{L}-L\| \le \delta_L$, $\|\overline{C}-C\| \le \delta_C$. Then the RMS state discrepancy $e_t = (\mathbb{E}[\|\overline{M}_t - M_t\|^2])^{1/2}$ obeys

$$e_{t+1} \le \rho e_t + \delta_F B_M + \delta_L B_Z.$$

Consequently

$$\sup_t \big(\mathbb{E}[\|\overline{C}\,\overline{M}_t - CM_t\|^2]\big)^{1/2} \le \varepsilon_\mu := \frac{\|\overline{C}\|(\delta_F B_M + \delta_L B_Z)}{1-\rho} + \delta_C B_M. \tag{5.31}$$

Here *contraction* means shrinking distances by a factor strictly below one. A weighted norm $\|x\|_H = (x^\top H x)^{1/2}$, $H \succ 0$, may supply contraction when the ordinary Euclidean norm does not. Alignment and norm conditioning then affect the constants.

For Gaussian conditional laws, mean accuracy and covariance accuracy together give sequence KL.

Proposition 5.9.4 (Conditional errors control Gaussian sequence KL)**.** *Let P and Q have conditional laws $\mathcal{N}(\mu_t, \Omega)$ and $\mathcal{N}(\widehat{\mu}_t, \widehat{\Omega})$, evaluated at the same prefix, with positive-definite constant covariances. Then*

$$D(P_{1:T}, Q_{1:T}) = \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}_P[(\mu_t - \widehat{\mu}_t)^\top \widehat{\Omega}^{-1}(\mu_t - \widehat{\mu}_t)] + \frac{T}{2}\left[\operatorname{tr}(\widehat{\Omega}^{-1}\Omega) - m + \log\frac{\det\widehat{\Omega}}{\det\Omega}\right].$$

If $\mathbb{E}_P[\|\mu_t - \widehat{\mu}_t\|^2] \le \varepsilon_\mu^2$ and all eigenvalues of $\widehat{\Omega}^{-1/2}\Omega\widehat{\Omega}^{-1/2}$ lie in $[1-\eta, 1+\eta]$, $0 \le \eta < 1$, then

$$\frac{1}{T}D(P_{1:T}, Q_{1:T}) \le \frac{\varepsilon_\mu^2}{2\lambda_{\min}(\widehat{\Omega})} + \frac{m\eta^2}{4(1-\eta)^2}. \tag{5.32}$$

Proof. Apply the sequence KL chain rule and the Gaussian KL formula. For $\lambda \in [1-\eta, 1+\eta]$, Taylor's theorem gives $\lambda - 1 - \log\lambda \le (\lambda-1)^2/[2(1-\eta)^2]$. Sum this bound over the covariance eigenvalues. □

This explains which small-noise issue matters for KL: mean errors are measured in units of conditional observation variance. A nearly zero $\lambda_{\min}(\widehat{\Omega})$ can make a small Euclidean error costly. The conditional observation covariance can remain positive definite when the physical process-noise covariance is singular.

For any sequence cost with range length $C_{\max}$, Pinsker gives

$$|\mathbb{E}_P[c] - \mathbb{E}_Q[c]| \le C_{\max}\sqrt{D(P_{1:T}, Q_{1:T})/2}.$$

The sequence KL bound grows with T. To control long-run generated statistics, use stability of the *generative* dynamics and an appropriate *coupling*: a joint construction of the two processes with their prescribed marginal laws. Appendix 5.A.4 gives that comparison. The required condition concerns $\overline{A} = \overline{F} + \overline{LC}$, the matrix governing generation.

5.9.5 Stable autoregression can also control long-run error

Stability can be imposed directly on the observable regression. Here is one sufficient condition that preserves convex fitting.

Proposition 5.9.5 (A stable autoregressive fit)**.** *Let $Z_{t+1} = P_t + E_{t+1}$, where P_t is its conditional mean and the E_t are iid additive innovations independent of the past. Fix fitted coefficient matrices satisfying*

$$\sum_{j=1}^{p} \|\widehat{B}_j\| \le \beta < 1, \qquad \sup_{t \ge p} \left(\mathbb{E}\left[\left\| P_t - \sum_{j=1}^{p} \widehat{B}_j Z_{t+1-j} \right\|_2^2 \right] \right)^{1/2} \le \varepsilon.$$

Start a fitted generator from the same first p observations and use the same innovations in a coupling:

$$\widetilde{Z}_{t+1} = \sum_{j=1}^{p} \widehat{B}_j \widetilde{Z}_{t+1-j} + E_{t+1}.$$

Then

$$\sup_t \big(\mathbb{E}[\|Z_t - \widetilde{Z}_t\|_2^2] \big)^{1/2} \le \frac{\varepsilon}{1-\beta}. \tag{5.33}$$

Proof. Let d_t denote the RMS discrepancy. Inserting the fitted predictor evaluated on the true history and using the L^2 triangle inequality gives $d_{t+1} \le \varepsilon + \sum_j \|\widehat{B}_j\| d_{t+1-j}$. Starting from $d_1 = \cdots = d_p = 0$, induction gives the result. □

The constraint on the sum of operator norms is convex. A statistic c is *L_c-Lipschitz* if $|c(x) - c(y)| \le L_c \|x - y\|$. Such a statistic of one observation, or its time average, then has expectation discrepancy at most $L_c \varepsilon/(1-\beta)$. Appendix 5.A.4 gives the corresponding state-space comparison and a second-moment bound. Thus both state-space modeling and autoregression can use stability. The choice concerns how useful memory is represented, how much is truncated, and what constraints the fitting method preserves.

5.9.6 A noiseless comparison: Cayley–Hamilton

In a noiseless LDS, we can eliminate the hidden state to obtain an exact recurrence involving only observations. The key linear-algebra result is the following.

Theorem 5.9.6 (Cayley–Hamilton)**.** *Let $A \in \mathbb{R}^{d \times d}$ have characteristic polynomial*

$$\chi_A(\lambda) = \det(\lambda I - A) = \lambda^d + a_{d-1}\lambda^{d-1} + \cdots + a_1 \lambda + a_0.$$

Then

$$\chi_A(A) = A^d + a_{d-1}A^{d-1} + \cdots + a_1 A + a_0 I = 0. \tag{5.34}$$

The proof by the adjugate identity is in Appendix 5.A.5.

Corollary 5.9.7 (Eliminating a noiseless hidden state)**.** *If $S_{t+1} = AS_t$ and $Z_t = CS_t$, then*

$$Z_{t+d} + a_{d-1} Z_{t+d-1} + \cdots + a_0 Z_t = 0. \tag{5.35}$$

Proof. The left side is $C\chi_A(A)S_t = 0$. □

The same scalar coefficients $a_0, \ldots, a_{d-1}$ apply to every observation coordinate, even when Z_t is vector-valued. This deterministic linear model is more expressive than it may first appear. Its trajectories are combinations of terms of the form $t^k \lambda^t$, arising from the eigenvalues and Jordan blocks of A. They therefore include exponential growth and decay, polynomial trends, damped or persistent oscillations, periodic motion, and superpositions of oscillations with incommensurate frequencies. Even dynamics that retain information indefinitely have the recurrence in Eq. (5.35).

Nonlinear state updates permit qualitatively different behavior, including multiple attractors and chaos. A precise one-dimensional result illustrates the distinction: if a continuous map from an interval to itself has a point of least period three, then it has points of every positive period and an uncountable scrambled set in the sense of Li and Yorke [Li and Yorke, 1975]. The conclusion requires continuity, an interval state space, and least period three; "period three implies chaos" is not a statement about an arbitrary nonlinear recurrence. The observation map can reveal or hide features of the state dynamics.

With noise, the linear elimination produces additional, generally correlated noise terms. To fit a predictor by regression, we need its conditional mean given the *observations*, as in Section 5.9.1.

5.10 Take-home messages

1. A useful state-space model restricts what is remembered and how it evolves. The observation process need not be Markov.
2. Language may be viewed as a partial observation of an evolving internal process, which motivates latent state. This does not imply that explicitly estimating a stochastic latent trajectory is the best predictive method.
3. Conditional generation first infers a distribution over the current state. Filtering, smoothing, and most likely path decoding answer different questions.
4. Hidden-state likelihood can be evaluated efficiently in important families, while its optimization remains difficult. EM uses posterior averages; joint state optimization uses a different objective.
5. Observable regression supplies a flexible predictor. Hankel factorization imposes shared dynamics and can recover a compact innovations realization. Finite-data success depends on truncation, excitation, rank conditioning, and residual modeling.
6. Fixed-window AR discards older observations. State models and transformers instead share parameters across prefix lengths, but state models compress the prefix into fixed-dimensional recurrent memory while standard attention retains token-level memory that grows with the context.
7. Memory in the physical state and forgetting of filtering errors are compatible. Prediction stability and generation stability must be checked separately.
8. Deterministic recurrent states permit backpropagation through observed prefixes. Structured affine updates can also support parallel training and recurrent evaluation of some attention maps.

Endnotes

1. **General spaces.** The conditional distributions used here exist for finite and Euclidean spaces. More general measurable spaces need additional regularity assumptions.

2. **Input–output models.** Adding an observed input U_t is a small change to the model definition, and filtering can condition on its realized values. Learning can be more difficult when inputs are chosen in response to past outputs or are correlated with unobserved disturbances; causal interpretation then requires assumptions beyond the autonomous model used here. Section 5.8 returns to inputs and feedback for recurrent and linear models.

3. **A linear update with a nonlinear readout.** Let $(B_t)_{t\geq 1}$ have any joint law on binary sequences. Set $S_0 = \sum_{k\geq 1} 3^{-k} B_k$, $S_{t+1} = 3S_t$, and $Z_t = \lfloor S_t \rfloor \bmod 3$ for $t \geq 1$. The fractional tail of S_t is at most $1/2$, and its integer part is $\sum_{k=1}^{t} 3^{t-k} B_k$, so $Z_t = B_t$. The initial law encodes the entire process. Exact generation for arbitrarily long times uses arbitrarily many digits of the initial state.

4. **Gaussian mixtures and clustering.** In a Gaussian mixture, the latent component label turns density estimation into a form of soft clustering. For a spherical mixture, the posterior label probabilities concentrate on the nearest center as the common variance tends to zero, connecting the EM updates to the assignment and center-update steps of k-means. In the two-component known-covariance setting studied by Daskalakis et al. [2017], population EM converges geometrically to the correct means, and their finite-sample procedure has a random-initialization guarantee. Balakrishnan et al. [2017] give local contraction and finite-sample guarantees under suitable initialization for several latent-variable models. These are model-specific results, not consequences of EM monotonicity alone.

5. **When does acceleration also perturb position?** If a random acceleration acts throughout a sampling interval, integrating it generally gives correlated position and velocity perturbations. The zero position-noise convention corresponds to motion at the old velocity followed by a random velocity change at the interval boundary.

6. **A rank obstruction.** Choose scalar queries $q_i = i - 1$ and keys $k_j = \log a_j$ for distinct positive a_j. Then $[\exp(q_i k_j)]_{i,j=1}^{n} = [a_j^{i-1}]_{i,j=1}^{n}$ is an invertible Vandermonde matrix. Row normalization preserves rank. Putting these keys before these queries gives a past-to-future block of causal attention with rank n. A fixed r-feature bilinear score matrix has rank at most r, so it cannot represent all these blocks for $n > r$. This bound concerns finite-feature bilinear representations.

Bibliographical notes

Filtering, smoothing, and likelihood fitting. Doucet and Johansen [2009] give an accessible particle-filtering and smoothing tutorial. Douc et al. [2014] establishes time-uniform control of asymptotic particle-filter variance and error under mixing and observation assumptions, including noncompact state spaces.

Dynamax implements EM and gradient-based marginal-likelihood fitting [Dynamax, 2026]. Thus differentiating a forward recursion is an established computational option. Wu [1983] supplies convergence conditions for EM and explains why monotonicity alone does not establish convergence

to a local optimum. Balakrishnan et al. [2017] give a general framework in which population EM is contractive in a neighborhood of a target optimum; for Gaussian mixtures, mixtures of regressions, and regression with missing covariates, suitable initialization leads finite-sample EM to statistical accuracy. Daskalakis et al. [2017] prove a stronger global geometric-convergence result for the population EM iteration in a two-Gaussian mixture with known covariances, together with a finite-sample result under their initialization scheme. The favorable conclusion is special: Jin et al. [2016] construct well-separated spherical mixtures with at least three components whose population likelihood has bad local maxima and for which random initialization can lead EM to bad critical points with high probability.

Dellaert and Kaess [2017] develops sparse trajectory MAP estimation for robotics. It concerns optimization over states and, in SLAM, map variables, rather than replacing marginal likelihood by an equivalent objective.

Gaussian prediction and realization. Mikusheva [2009] derives Gaussian state prediction. The sufficient-condition proof in Appendix 5.A.2 is included in full. Ho and Kalman [1966] gives the classical realization construction from input-output coefficient matrices; Section 5.9.2 applies that algebra to the predictor's coefficients.

For finite stochastic data, Tsiamis and Pappas [2019], Theorem 1 and Theorem 4 with Remark 4, analyzes future-on-past block regression and an SVD-based realization. Assumptions 1–3 specify known state dimension, observability, controllability by process noise, positive measurement covariance, steady-state filtering initialization, and sufficient block lengths. Together with sample-size and conditioning requirements, the bounds give matrix error of order $T^{-1/2}$ up to logarithmic factors and similarity. The allowed $\rho(A) \leq 1$ includes the boundary case $\rho(A) = 1$. This is a closely related algorithm, not a finite-sample theorem for every one-step AR/SVD variant.

Tsiamis and Pappas [2020] instead bounds *regret*, cumulative squared loss minus that of the Kalman predictor, using online regression. With high probability the bound grows as a power of $\log T$, including systems with $\rho(A) \leq 1$ under the paper's conditions. This directly addresses prediction performance; it need not identify a compact generative realization. The Gaussian KL and coupling calculations here explain additional steps needed when the evaluation criterion is a sequence law or a generated statistic.

Synthetic comparisons and spectral prediction. Synthetic dynamical systems are a standard controlled setting for comparing sequence models. Agarwal et al. [2024], Section 4.1 and Appendix E.1, tests a system with diagonal eigenvalues ± 0.9999 and compares spectral units with learned linear recurrence. These eigenvalues are strictly inside the unit circle, although the appendix calls the system marginally stable. Our constant-state calculation isolates the effect of history length; the AR/SVD experiment compares two fitted predictors.

Hazan et al. [2017] fits a predictor in a fixed spectral feature space and proves guarantees against a specified LDS class. The learner need not recover the comparator's state coordinates. Agarwal et al. [2024] develops spectral sequence layers and an autoregressive extension. Dogariu et al. [2025] extends observable-history prediction using spectral filtering and linear observers in a nonlinear-dynamics setting; the bounds require their structural and conditioning assumptions. These are additional ways to exploit dynamical structure.

Recurrent computation and language models. Pascanu et al. [2013] analyzes vanishing and exploding gradients in recurrent-network training. Katharopoulos et al. [2020] derives recurrent causal linear attention. Dao and Gu [2024], Sections 2 and 5, develops structured state-space duality

and efficient matrix-based algorithms; the finite-feature calculation in Section 5.8.1 is an elementary example.

Bick et al. [2024] introduces MOHAWK's staged transformer-to-SSM distillation. Wang et al. [2024] converts pretrained attention layers and trains hybrid recurrent/attention models. Their measured results concern the models, tasks, and compute settings in those papers.

Shi et al. [2026] introduce Hierarchical Latent Prediction, which adds auxiliary objectives requiring token-level and coarser hidden representations to predict future hidden representations at several temporal scales. These are partial state-like constraints on the shared transformer representation. The auxiliary hierarchy is removed at inference, so generation still uses the ordinary next-token head. The paper reports improvements on its tested coding, multi-step reasoning, and speculative decoding evaluations.

Nonlinear recurrence and chaos. Li and Yorke [1975] prove the theorem summarized after Corollary 5.9.7: for a continuous self-map of an interval, a point of least period three implies points of every positive period and an uncountable scrambled set. This result motivates the standard phrase "period three implies chaos," with the stated setting and Li–Yorke notion of chaos.

Glossary

State-space model; latent state
A model of observations through an evolving, possibly unobserved state. State coordinates specify the information retained over time.

Transition and observation kernels
Conditional probability laws of the next state given the current state, and of the current observation given the current state. The observation kernel is also called the emission kernel; it averages over observation noise.

One-hot encoding; martingale difference
$Z_t = e_{X_t}$ turns token probabilities into vector conditional means. Subtracting a conditional mean produces a residual with conditional expectation zero given the past.

Linear dynamical system; linear autoregression
An LDS has linear state and observation equations. A linear autoregression predicts from a linear combination of observed lags.

Unit-delay box D
A diagram element that stores its incoming signal for one time step. Function boxes are static maps; feedback through a delay gives the diagram memory.

System identification
Estimating dynamical and observation equations from observed data.

Innovation; Kalman predictor
An innovation is the observation minus its conditional-mean prediction. For the Gaussian LDS considered here, the Kalman predictor computes that mean recursively, and innovations are independent Gaussian vectors.

Truncation error
The predictive effect of history omitted by a finite-lag approximation. It is controlled by the tail of the history coefficients.

Hankel factorization
Factoring a matrix of lag coefficients to recover a finite-dimensional recurrent predictor. Its rank expresses the minimal predictive state dimension under the stated rank assumptions.

Filtering; prediction; smoothing
Conditioning a current state on observations through the current time, conditioning a future state on earlier observations, and conditioning a state on observations before and after it.

Observation-driven state
A recurrent state updated deterministically using observations; given parameters and its initial state, the observed prefix determines it.

EM
Alternating posterior inference and expected complete-data likelihood maximization.

Excitation; realization conditioning
Variation in observed regressors and separation of retained singular values from zero; both affect recovery from noisy data.

Backpropagation through time
Differentiation of the unrolled recurrent computation.

5.11 Exercises

The exercises follow the modeling, estimation, and computational questions. The noiseless recurrence in Exercise 11 is optional.

General instructions. Justify every answer, including yes/no answers, even when the question does not explicitly ask for a justification.

Exercise profiles. Filled mountains show difficulty on a four-mountain scale: one is a direct application, two uses a familiar method, three combines ideas, and four is a substantial challenge. Filled cogs show technical load on a three-cog scale: one is light, two is sustained, and three is dense. Filled hourglasses show estimated time: one is under 20 minutes, two is 20–45 minutes, three is 45–90 minutes, and four is more than 90 minutes. Pale pictograms show the unused part of each scale. The work row uses the canonical order proof, calculation, construction, computation, and interpretation; pale icons mark work types not used. ◎ marks the intended learning outcome. Time estimates cover the non-optional parts. Ratings are not prerequisites or grading weights. A superscript asterisk marks an exercise based on supplementary lecture material; it is not a difficulty rating and does not make the exercise optional when assigned.

Exercise 1: Observed feedback does not imply Markov observations.

◎ Construct a Markov state representation and verify that the observed process it generates is not first-order Markov.

Let W_t be iid Bernoulli(1/4), put $Z_{-1} = Z_0 = 0$, and set $Z_t = Z_{t-2} \oplus W_t$, where $\oplus$ is addition modulo two.

(a) Express the model as Eq. (5.2) with $R_t = (Z_{t-2}, Z_{t-1})$. Show that (R_t, Z_t) is Markov.

(b) Compare $Q(Z_3 = 1 \mid Z_1 = 0, Z_2 = 0)$ and $Q(Z_3 = 1 \mid Z_1 = 1, Z_2 = 0)$. Prove that both conditioning events have positive probability and that (Z_t) is not first-order Markov.

Noise maps and HMM estimation

Exercise 2: From noise-driven maps to an HMM law.

◎ Construct finite-state noise maps, derive their induced HMM law, and distinguish a stochastic law from its nonunique noise-map representations.

Use Eqs. (5.1) and (5.3) and the finite HMM matrices A, C defined in Eq. (5.4).

(a) Prove the joint-law product formula by successive conditioning. Write the marginal law of $S_{1:T}, Z_{1:T}$ after integrating out S_0.

(b) For independent uniform $[0, 1]$ noises, give explicit inverse-cumulative-sum maps implementing A and C, including a convention at zero-probability endpoints.

(c) Construct two different noise maps with the same transition kernel. Explain why a specified model gives a unique law but the law does not uniquely specify its noise maps.

Exercise 3: Observed-state HMM estimates.

◎ Derive transition and emission maximum-likelihood estimates and identify the parameters left undetermined by unvisited states.

Suppose the hidden states in the finite HMM of Eq. (5.4) are observed. Define

$$N_{ij} = \sum_{t=1}^{T} \mathbb{I}\{S_{t-1} = j, S_t = i\}, \qquad H_{aj} = \sum_{t=1}^{T} \mathbb{I}\{X_t = a, S_t = j\}.$$

Show that, whenever the denominators are positive, the maximum-likelihood estimates are

$$\widehat{A}_{ij} = \frac{N_{ij}}{\sum_k N_{kj}}, \qquad \widehat{C}_{aj} = \frac{H_{aj}}{\sum_b H_{bj}}.$$

Identify the columns left undetermined when a denominator is zero.

Exercise 4: Latent likelihood and the forward algorithm.

Derive the forward recursion and explain why it replaces enumeration of d^{T+1} hidden trajectories by $O(Td^2)$ computation.

Let d, N, T be positive integers. Let $\pi \in [0,1]^d$ be a probability vector, and let $A \in [0,1]^{d\times d}$ and $C \in [0,1]^{N\times d}$ satisfy

$$\sum_{i=1}^{d} A_{ij} = 1, \qquad \sum_{a=1}^{N} C_{aj} = 1 \quad \text{for every } j \in [d].$$

Let Q be the joint probability mass function of a hidden Markov model on

$$[d]^{T+1} \times [N]^T.$$

The initial-state pmf is π. Given previous state j, the next-state pmf is the column $A_{\cdot j}$. Given current state j, the emission pmf is the column $C_{\cdot j}$. These conditional pmfs do not depend on t. Fix $x_{1:T} \in [N]^T$. For $s_{0:T} \in [d]^{T+1}$, write $Q(s_{0:T}, x_{1:T})$ for the joint mass and $Q(x_{1:T})$ for the marginal mass of $x_{1:T}$.

(a) Derive

$$Q(s_{0:T}, x_{1:T}) = \pi_{s_0} \prod_{t=1}^{T} A_{s_t s_{t-1}} C_{x_t s_t}, \qquad Q(x_{1:T}) = \sum_{s_{0:T} \in [d]^{T+1}} \pi_{s_0} \prod_{t=1}^{T} A_{s_t s_{t-1}} C_{x_t s_t}.$$

Verify that the sum has d^{T+1} terms.

(b) For $t \in [T]$, let Q_t be the joint marginal pmf on $[N]^t \times [d]$. Define $\Lambda_0 = \pi$ and $\Lambda_t \in \mathbb{R}^d$ by $\Lambda_t(j) = Q_t(x_{1:t}, j)$ for $j \in [d]$. Derive a recursion for Λ_t and show that, in vector form, it is

$$\Lambda_t = \operatorname{diag}(C_{x_t,1}, \ldots, C_{x_t,d}) A \Lambda_{t-1}.$$

(c) Let $\mathbf{1} \in \mathbb{R}^d$ be the all-ones vector. Show that $Q(x_{1:T}) = \mathbf{1}^\top \Lambda_T$ and that the recursion evaluates it in $O(Td^2)$ arithmetic operations, rather than summing d^{T+1} terms.

Exercise 5: EM increases likelihood but can stop at a saddle.

Prove EM monotonicity and construct an exact EM iteration that remains at a saddle point of the observed log likelihood.

Let the unobserved variable X and observed variable Y take values in finite sets. Fix an observed value y_{obs}, and assume that $Q_\theta(x, y_{\text{obs}}) > 0$ for every allowed θ, x. Write $\ell(\theta) = \log Q_\theta(y_{\text{obs}})$. Starting from θ_k, one EM iteration consists of the E-step

$$q_k(x) = Q_{\theta_k}(X = x \mid Y = y_{\text{obs}})$$

and the M-step

$$\theta_{k+1} \in \arg\max_\theta \sum_x q_k(x) \log Q_\theta(x, y_{\text{obs}}).$$

A generalized M-step need only make this sum no smaller than its value at θ_k. For part (a), fix θ_0 and abbreviate $q(x) = Q_{\theta_0}(X = x \mid Y = y_{\text{obs}})$.

(a) Prove

$$\ell(\theta) - \ell(\theta_0) = \sum_x q(x) \log \frac{Q_\theta(x, y_{\text{obs}})}{Q_{\theta_0}(x, y_{\text{obs}})} + D(q, Q_\theta(\cdot \mid y_{\text{obs}})).$$

Deduce monotonicity of exact EM and of an M-step that merely increases the expected complete-data log likelihood.

(b) The proof in part (a) depends on q being the full current posterior. Identify the equality that is lost if q is replaced by a point mass at a most likely hidden value. Explain also why monotonicity of the likelihood values does not guarantee convergence to a local maximum.

(c) To exhibit the stationary-point problem from the second subproblem of part (b), consider parameters $(a, b) \in (-1, 1)^2$, with

$$Q_{a,b}(X = x) = \frac{1 + ax}{2}, \qquad Q_{a,b}(Y = 1 \mid X = x) = \frac{1 + bx}{2}, \qquad x \in \{-1, 1\}.$$

Suppose that $Y = 1$ is observed. Show that $\ell(a, b) = \log((1+ab)/2)$. Starting from $(a, b) = (0, 0)$, compute the E- and M-steps and show that exact EM remains at the origin. Use the Hessian of ℓ to show that the origin is a saddle point.

(d) The stationary point in part (c) has an indefinite Hessian. Now suppose that EM converges to an interior stationary point $\theta_\star$. Assume that the observed log likelihood is twice continuously differentiable near $\theta_\star$ and $\nabla^2 \ell(\theta_\star)$ is negative definite. What can be concluded about $\theta_\star$? Justify the conclusion.

HINTS For (a), factor the joint mass as $Q_\theta(x, y_{\text{obs}}) = Q_\theta(x \mid y_{\text{obs}}) Q_\theta(y_{\text{obs}})$. For (c), first compute the posterior of X at the origin. The expected complete-data log likelihood then separates into a function of a and a function of b.

Continuous-state EM

Exercise 6: Joint densities and conditional laws.

Construct a conditional law from a common joint density and verify its disintegration identity.

Let Q_θ be a family of probability laws on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are standard Borel spaces. Fix sigma-finite reference measures P_X and P_Y, put $P = P_X \otimes P_Y$, and assume that $Q_\theta \ll P$. Write

$$q_\theta(x, y) = \frac{dQ_\theta}{dP}(x, y), \qquad m_\theta(y) = \int_{\mathcal{X}} q_\theta(x, y)\, P_X(dx).$$

Let $Q_\theta^{(Y)}$ denote the marginal law of Y.

(a) Prove that $Q_\theta^{(Y)} \ll P_Y$ and

$$\frac{dQ_\theta^{(Y)}}{dP_Y}(y) = m_\theta(y) \quad \text{for } P_Y\text{-almost every } y.$$

(b) For y such that $m_\theta(y) > 0$, define

$$r_\theta(x \mid y) = \frac{q_\theta(x, y)}{m_\theta(y)}.$$

Show that this is a probability density with respect to P_X. For measurable $A \subseteq \mathcal{X}$, set

$$R_\theta(y, A) = \int_A r_\theta(x \mid y)\, P_X(dx).$$

When $m_\theta(y) = 0$, define $R_\theta(y, \cdot)$ to be any fixed probability law on $\mathcal{X}$. Verify that $y \mapsto R_\theta(y, A)$ is measurable for each measurable $A \subseteq \mathcal{X}$, and, for measurable $B \subseteq \mathcal{Y}$, that

$$Q_\theta(X \in A, Y \in B) = \int_B R_\theta(y, A)\, Q_\theta^{(Y)}(dy).$$

Conclude that $R_\theta(y, \cdot)$ is a version of $Q_\theta(X \in \cdot \mid Y = y)$.



Exercise 7: Continuous EM and reference measures.

Prove continuous-EM monotonicity and show why its complete-data objective is invariant under a change of common dominating measure.

Use the setup and notation of Exercise 6. Fix y_{obs} such that, for every parameter under consideration, $m_\theta(y_{\text{obs}}) > 0$ and $q_\theta(x, y_{\text{obs}}) > 0$ for P_X-almost every x. Put $r_k(x) = r_{\theta_k}(x \mid y_{\text{obs}})$. Continuous EM uses

$$\theta_{k+1} \in \arg\max_\theta \int_{\mathcal{X}} r_k(x) \log q_\theta(x, y_{\text{obs}})\, P_X(dx).$$

(a) Prove the density analogue of the identity in Exercise 5 and deduce that this update does not decrease $\log m_\theta(y_{\text{obs}})$.

(b) Explain why the reference measure must be common to all parameter values. More precisely, suppose another common reference measure satisfies $dP' = h\, dP$, where $h > 0$. Show that replacing P by P' changes the expected complete-data log density by a term independent of θ. Explain how counting measures recover the finite-space formulas in Exercise 5.

Gaussian transition EM

Exercise 8: EM for a linear transition with Gaussian noise.

Derive the exact E- and M-steps for a Gaussian transition from the conditional moments of the hidden trajectory.

Use the density form of EM from Exercise 6. Let $S_0, \dots, S_T \in \mathbb{R}^d$. Relative to Lebesgue measure, suppose the transition density is

$$p_A(s_{t+1} \mid s_t) = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{\|s_{t+1} - As_t\|^2}{2\sigma^2}\right), \qquad t = 0, \dots, T-1,$$

where $\sigma^2 > 0$ is fixed. Let p_0 be the initial-state density and let $g_\eta(z_t \mid s_t)$ be the observation density. Assume that neither depends on A. Denote the joint law by $Q_{A,\eta}$. Its complete-data density is

$$q_{A,\eta}(s_{0:T}, z_{1:T}) = p_0(s_0) \prod_{t=0}^{T-1} p_A(s_{t+1} \mid s_t) \prod_{t=1}^{T} g_\eta(z_t \mid s_t).$$

Write its observed-data density as

$$m_{A,\eta}(z_{1:T}) = \int q_{A,\eta}(s_{0:T}, z_{1:T})\, ds_{0:T}.$$

Fix an observed sequence $z_{1:T,\mathrm{obs}}$, and assume that $m_{A_k,\eta_k}(z_{1:T,\mathrm{obs}}) > 0$ at every iteration. At iteration k, let

$$\Pi_{k,\mathrm{obs}}(ds_{0:T}) = Q_{A_k,\eta_k}(ds_{0:T} \mid Z_{1:T} = z_{1:T,\mathrm{obs}})$$

be the conditional law of the entire hidden trajectory given the complete observed sequence. This is often called the *smoothing distribution*. In contrast, a filtering distribution at time t conditions only on observations through time t.

For this fixed iteration and observed sequence, abbreviate the conditional means, covariances, and adjacent-state cross-covariances by

$$\mu_t = \mathbb{E}_{\Pi_{k,\mathrm{obs}}}[S_t], \qquad P_t = \mathbb{E}_{\Pi_{k,\mathrm{obs}}}[(S_t - \mu_t)(S_t - \mu_t)^\top],$$
$$P_{t,t+1} = \mathbb{E}_{\Pi_{k,\mathrm{obs}}}[(S_t - \mu_t)(S_{t+1} - \mu_{t+1})^\top].$$

(a) Derive the density of $\Pi_{k,\mathrm{obs}}$ with respect to Lebesgue measure on the hidden trajectory:

$$\pi_{k,\mathrm{obs}}(s_{0:T}) = \frac{q_{A_k,\eta_k}(s_{0:T}, z_{1:T,\mathrm{obs}})}{m_{A_k,\eta_k}(z_{1:T,\mathrm{obs}})}.$$

Identify the numerator and normalizing constant.

(b) Write one exact EM iteration for (A, η): give its E-step and its M-step as an optimization of the expected complete-data log density under $\Pi_{k,\mathrm{obs}}$. Then show that the part of this M-step objective that depends on A is

$$-\frac{1}{2\sigma^2} \sum_{t=0}^{T-1} \mathbb{E}_{\Pi_{k,\mathrm{obs}}}[\|S_{t+1} - AS_t\|^2].$$

(c) Derive the following expression for each summand:

$$\mathbb{E}_{\Pi_{k,\mathrm{obs}}}[\|S_{t+1} - AS_t\|^2] = \|\mu_{t+1} - A\mu_t\|^2 + \mathrm{tr}(P_{t+1}) - 2\,\mathrm{tr}(AP_{t,t+1}) + \mathrm{tr}(A^\top A P_t).$$

(d) Assume that

$$M_{k,\mathrm{obs}} = \sum_{t=0}^{T-1} \mathbb{E}_{\Pi_{k,\mathrm{obs}}}[S_t S_t^\top]$$

is invertible. Derive the exact M-step update for A, expressed in terms of μ_t, P_t, and $P_{t,t+1}$.

Exercise 9: A posterior-mean approximation to the M-step.

Identify the uncertainty terms lost when the exact conditional distribution is replaced by its posterior-mean trajectory.

Use the setting and notation of Exercise 8. Replace $\Pi_{k,\mathrm{obs}}$, the exact conditional distribution of the hidden trajectory given $z_{1:T,\mathrm{obs}}$, by a point mass at $(\mu_0, \ldots, \mu_T)$. Derive the resulting update for A when its required inverse exists, and identify the conditional covariance terms that this update omits. Explain why this point-trajectory update is not exact EM in general and need not inherit EM's likelihood-monotonicity guarantee.

When $\Pi_{k,\mathrm{obs}}$ has a nonsingular Gaussian density, its mean $(\mu_0, \ldots, \mu_T)$ is also its unique mode. The substitution therefore uses the most likely hidden trajectory.

Exercise 10: Denoising coefficients by parameter sharing.

Quantify how a shared low-dimensional parameterization reduces variance and determine exactly when its bias–variance tradeoff improves prediction.

Let $U \in \mathbb{R}^{p\times r}$ be known with $U^\top U = I_r$. Suppose a preliminary estimate is $\widehat{b} = b + \xi$, where $\xi \sim \mathcal{N}(0, \sigma^2 I_p/n)$. Set $\widetilde{b} = UU^\top \widehat{b}$.

(a) If b lies in the column space of U, compute the expected squared coefficient errors of $\widehat{b}$ and $\widetilde{b}$.

(b) Let an independent test feature Ψ have mean zero and $\mathbb{E}[\Psi\Psi^\top] = I_p$. Compute the corresponding prediction errors for predicting $b^\top \Psi$.

(c) Realize a one-dimensional example using $b_j = \theta\lambda^{j-1}$, $j = 1, \ldots, p$, with known $|\lambda| < 1$. Explain the relationship to a scalar recurrent predictor.

(d) For arbitrary b, derive the bias–variance formula for $\widetilde{b}$. Exactly when is its expected coefficient error smaller than that of $\widehat{b}$?

The independent isotropic coefficient errors are a simplifying model, not an assertion about lagged time-series least squares.

Linear state-space structure

Exercise 11: Observable recurrence.*

Use Cayley–Hamilton to derive an observable recurrence while showing that hidden-state differences persist.

Let $S_{t+1} = AS_t$, where $A = \left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$, and let $Z_t = c^\top S_t$. Use Cayley–Hamilton to show that

$$Z_{t+2} = 2\cos(\theta) Z_{t+1} - Z_t.$$

Show that distinct initial states remain at the same Euclidean distance under these dynamics.

Exercise 12: Predictive-state realization.*

Track initialization, Hankel factorization, and coordinate invariance in a predictive-state realization.

Use $M_{t+1} = FM_t + LZ_t$, $\widehat{Z}_t = CM_t$, and $K_j = CF^{j-1}L$ from Eqs. (5.22) and (5.25).

(a) Unroll the recurrence from a nonzero fixed M_1 and show that

$$CM_t = CF^{t-1}M_1 + \sum_{j=1}^{t-1} K_j Z_{t-j}.$$

(b) For an integer $q \geq 1$, define

$$\mathcal{O} = (C^\top, (CF)^\top, \ldots, (CF^{q-1})^\top)^\top, \qquad \mathcal{R} = (L, FL, \ldots, F^{q-1}L).$$

Verify, including the index shifts, that

$$[K_{i+j-1}]_{i,j=1}^{q} = \mathcal{O}\mathcal{R}, \qquad [K_{i+j}]_{i,j=1}^{q} = \mathcal{O}F\mathcal{R}.$$

(c) Show directly that replacing (F, C, L) by $(T_0^{-1}FT_0, CT_0, T_0^{-1}L)$, with T_0 invertible, preserves every history coefficient.

Exercise 13: Prediction memory and generation.*

Identify why predictor memory can decay while generated-state differences grow.

Use $M_{t+1} = FM_t + LZ_t$, $\widehat{Z}_t = CM_t$, and $K_j = CF^{j-1}L$ from Eqs. (5.22) and (5.25). For scalars $F = 1/2, C = 1, L = 3/4$, show that predictor-memory coefficients decay as

$$K_j = \frac{3}{4}\left(\frac{1}{2}\right)^{j-1},$$

whereas differences between generated states are multiplied by $F + LC = 5/4$. Identify why F governs the first calculation and $F + LC$ governs the second.

Exercise 14: Permanent state and finite-window prediction.

Quantify the prediction loss from a finite context when information is stored in a permanent latent state.

Let $n \geq 1$ be an integer. On a common probability space, let U and $(W_t)_{t\geq 1}$ be real-valued random variables satisfying

$$U \sim \mathcal{N}(0, \tau^2), \qquad Z_t = U + W_t, \qquad W_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2),$$

where U is unobserved, is independent of the noise sequence, and $\tau, \sigma > 0$. The goal is to predict Z_{n+1} given $Z_{1:n} = z_{1:n}$, where $z_{1:n} \in \mathbb{R}^n$. For $k \geq 1$, define

$$v_k = (\tau^{-2} + k\sigma^{-2})^{-1}.$$

(a) Show that

$$\mathcal{L}(U \mid Z_{1:n} = z_{1:n}) = \mathcal{N}(m_n, v_n), \qquad m_n = v_n\sigma^{-2}\sum_{i=1}^{n} z_i.$$

Starting from $a_0 = 0$, derive the running-sum recursion

$$a_k = a_{k-1} + z_k, \qquad m_k = \frac{a_k}{\sigma^2/\tau^2 + k}.$$

(b) Let $1 \le p \le n$. Consider a full-history predictor $h_n(Z_{1:n})$ and a window predictor $h_p(Z_{n-p+1:n})$, where h_n and h_p are arbitrary prediction functions. Show that

$$\min_{h_n} \mathbb{E}[(Z_{n+1} - h_n(Z_{1:n}))^2] = \sigma^2 + v_n$$

and

$$\min_{h_p} \mathbb{E}[(Z_{n+1} - h_p(Z_{n-p+1:n}))^2] = \sigma^2 + v_p.$$

Show that the minima are attained by

$$\widehat{Z}^{\text{full}}_{n+1} = \mathbb{E}[U \mid Z_{1:n}], \qquad \widehat{Z}^{\text{win}}_{n+1} = \mathbb{E}[U \mid Z_{n-p+1:n}].$$

Your result must apply to all predictors based on the indicated observations, including nonlinear predictors. For fixed p, show that the first minimum tends to σ^2, while the second remains $\sigma^2 + v_p$, as $n \to \infty$.

(c) *Optional extension.* Let P_n and $Q_{p,n}$ be the following random probability laws on $\mathbb{R}$:

$$P_n = \mathcal{L}(Z_{n+1} \mid Z_{1:n}), \qquad Q_{p,n} = \mathcal{L}(Z_{n+1} \mid Z_{n-p+1:n}).$$

Identify these two Gaussian predictive laws and show that

$$\mathbb{E}[D(P_n, Q_{p,n})] = \frac{1}{2} \log \frac{\sigma^2 + v_p}{\sigma^2 + v_n},$$

where the expectation is under the stated joint law of $Z_{1:n}$.

(d) *Optional extension.* Suppose now that τ^2 is unknown, let $K \ge 1$ be an integer, and suppose that we observe K independent trajectories

$$Z_{j,t} = U_j + W_{j,t}, \qquad j \in [K], \quad t \in [n],$$

where the U_j are iid $\mathcal{N}(0, \tau^2)$, the $W_{j,t}$ are iid $\mathcal{N}(0, \sigma^2)$, and the two collections are independent. Let $\bar{Z}_{j,n} = n^{-1} \sum_{t=1}^{n} Z_{j,t}$. Show that

$$\widehat{\tau}^2_{K,n} = \frac{1}{K} \sum_{j=1}^{K} \bar{Z}^2_{j,n} - \frac{\sigma^2}{n}$$

is unbiased for τ^2 and, for fixed n, satisfies $\widehat{\tau}^2_{K,n} \to \tau^2$ almost surely as $K \to \infty$. Then set $K = 1$ and show that $\widehat{\tau}^2_{1,n} \to U_1^2$ almost surely as $n \to \infty$. Explain why this limit does not consistently estimate τ^2.

Hints For (b), let X be any random element (e.g., a vector), and let Y, Z be real-valued random variables satisfying $\mathbb{E}[Y^2], \mathbb{E}[Z^2] < \infty$. If Z is a function of X, then

$$\mathbb{E}[(Y - \mathbb{E}[Y \mid X])Z] = 0\,.$$

For (c), recall or derive that, for $a, b \in \mathbb{R}$ and $A, B > 0$,

$$D\big(\mathcal{N}(a, A), \mathcal{N}(b, B)\big) = \frac{1}{2} \left[\log \frac{B}{A} + \frac{A + (a-b)^2}{B} - 1 \right].$$

The identity in the hint for part (b) may also be useful.

Recurrences, scans, and recurrent attention

Exercise 15: Recurrent gradients and nonconvexity.

Relate products of state Jacobians to vanishing or exploding gradients and show that a linear state update can still yield a nonconvex parameter objective.

Consider $R_{t+1} = h_\theta(R_t, u_t)$ with observed inputs.

(a) Express $\partial R_T / \partial R_s$ as a product of state-update Jacobians. For a scalar linear update $R_{t+1} = aR_t$, compare $0 < |a| < 1$ and $|a| > 1$.

(b) Show that linearity in the state does not make learning the transition parameter convex: use $R_0 = 1$, $R_2 = a^2$, and loss $(R_2 - 1)^2$.

Exercise 16: Parallel scans of affine recurrences.

Derive the work, depth, and storage costs of affine scans and identify the advantage of diagonal transitions.

For affine maps, define

$$(A_2, b_2) \circ (A_1, b_1) = (A_2 A_1, A_2 b_1 + b_2).$$

Verify that this operation is associative. Show that a dense d-dimensional sequential recurrence has $O(Td^2)$ work and $O(T)$ sequential depth, while a work-inefficient dense parallel scan has $O(Td^3)$ work and $O(\log T)$ composition depth. Compare their activation storage. Then show that diagonal transitions reduce the scan to $O(Td)$ work, $O(\log T)$ composition depth, and $O(Td)$ storage.

Exercise 17: Recurrent linear attention.

Derive recurrent linear attention and construct the additional state required for normalization.

Starting from

$$R_t = \gamma_t R_{t-1} + v_t k_t^\top, \qquad y_t = R_t q_t,$$

show that

$$y_t = \sum_{u=1}^{t} v_u (k_u^\top q_t) \prod_{j=u+1}^{t} \gamma_j.$$

For $\gamma_t = 1$, derive a recurrent state d_t for which the normalized output has denominator $d_t^\top q_t$. State conditions under which these weights define a normalized weighted average.

5.A Deferred proofs and stability comparisons

5.A.1 KL for continuous laws

For probability laws P and Q on the same measurable space, suppose that P is absolutely continuous with respect to Q, written $P \ll Q$. Then the density ratio dP/dQ exists and

$$D(P,Q) = \int \log\left(\frac{dP}{dQ}\right) dP.$$

If $P \not\ll Q$, set $D(P,Q) = +\infty$. When both laws have densities p, q with respect to Lebesgue measure, this becomes

$$D(P,Q) = \int p(z) \log \frac{p(z)}{q(z)}\, dz.$$

This definition applies directly to discrete, continuous, and mixed laws; it does not require defining differential entropy. It follows the treatment of relative entropy on general measurable spaces in Lattimore and Szepesvári [2020, Chapter 14].

For scalar Gaussian laws with $a, b > 0$, direct integration gives

$$D(\mathcal{N}(\mu,a),\mathcal{N}(\nu,b)) = \frac{1}{2}\left(\log\frac{b}{a} + \frac{a+(\mu-\nu)^2}{b} - 1\right). \tag{5.36}$$

For random conditional laws, expected conditional KL means that this pointwise KL is averaged over the conditioning history.

5.A.2 The constant-gain Gaussian predictor

Theorem 5.9.1 (A constant-gain Gaussian predictor)**.** *Initialize the system at time* 1 *for this theorem. Consider* $S_{t+1} = AS_t + V_{t+1}$, $Z_t = CS_t + W_t$ *for* $t \geq 1$, *with spectral radius* $\rho(A) < 1$, *independent centered Gaussian process and observation noises with covariances* $\Sigma_V \succ 0$ *and* $\Sigma_W \succ 0$. *There are matrices* $\Pi \succ 0$ *and* L *such that, when* $S_1 \sim \mathcal{N}(0,\Pi)$ *independently of the noises, the recursion*

$$M_1 = 0, \qquad M_{t+1} = FM_t + LZ_t, \qquad F = A - LC \tag{5.22}$$

satisfies

$$M_t = \mathbb{E}[S_t \mid Z_{<t}], \qquad Z_t \mid Z_{<t} \sim \mathcal{N}(CM_t, \Omega), \qquad \Omega = C\Pi C^\top + \Sigma_W. \tag{5.23}$$

Moreover, $\rho(F) < 1$ *and the observations have uniformly bounded second moments.*

Proof. We construct the covariance used in Theorem 5.9.1. If $S_t \mid Z_{<t} \sim \mathcal{N}(M_t, \Pi_t)$, Gaussian conditioning and the state equation give

$$\begin{aligned} \Omega_t &= C\Pi_t C^\top + \Sigma_W, \qquad L_t = A\Pi_t C^\top \Omega_t^{-1}, \\ M_{t+1} &= AM_t + L_t(Z_t - CM_t), \\ \Pi_{t+1} &= A\Pi_t A^\top + \Sigma_V - A\Pi_t C^\top \Omega_t^{-1} C\Pi_t A^\top. \end{aligned} \tag{5.37}$$

The covariance update is a deterministic map $\mathcal{T}(\Pi_t)$. To show that it has a fixed point under the stated conditions, use the identity

$$(A-GC)\Pi(A-GC)^\top + \Sigma_V + G\Sigma_W G^\top = \mathcal{T}(\Pi) + (G-L)(C\Pi C^\top + \Sigma_W)(G-L)^\top, \tag{5.38}$$

where $L = A\Pi C^\top (C\Pi C^\top + \Sigma_W)^{-1}$. For any vector, its quadratic form on the left is minimized over G by $G = L$. It follows that $\mathcal{T}$ is monotone in the positive-semidefinite order.

Since $\rho(A) < 1$, the finite matrix

$$B = \sum_{j=0}^{\infty} A^j \Sigma_V (A^\top)^j$$

satisfies $B = ABA^\top + \Sigma_V$. Starting from $\Pi_0 = 0$, iterate $\Pi_{n+1} = \mathcal{T}(\Pi_n)$. Monotonicity gives an increasing sequence, and $\mathcal{T}(B) \preceq ABA^\top + \Sigma_V = B$ bounds it above. It therefore converges to a fixed point Π. Continuity follows from $\Sigma_W \succ 0$, and $\Pi \succeq \Sigma_V \succ 0$.

Set $L = A\Pi C^\top (C\Pi C^\top + \Sigma_W)^{-1}$ and $F = A - LC$. The fixed-point identity becomes

$$\Pi = F\Pi F^\top + \Sigma_V + L\Sigma_W L^\top.$$

Thus $\Pi - F\Pi F^\top \succ 0$; conjugation by $\Pi^{-1/2}$ shows that $\rho(F) < 1$. Initialize $S_1 \sim \mathcal{N}(0, \Pi)$. Induction in Eq. (5.37) now keeps the error covariance equal to Π, proving Eqs. (5.22) and (5.23).

Finally,

$$S_t = A^{t-1} S_1 + \sum_{j=2}^{t} A^{t-j} V_j.$$

The geometric decay of powers of A bounds its second moment uniformly in t; $Z_t = CS_t + W_t$ has the same property. Geometric decay of powers of F also supplies the constants G, γ in Proposition 5.9.2. □

5.A.3 Why the Hankel factors recover the predictor

Proposition 5.9.3 (Recovery of the prediction map)**.** *The recovered matrices satisfy*

$$\overline{C}\,\overline{F}^{j-1}\overline{L} = K_j \quad \textit{for every } j \geq 1.$$

Thus Eq. (5.16) computes the same predictor on every observed history.

Proof. Under the ranks assumed in Section 5.9.2, both $\mathcal{H}_0 = \mathcal{O}_\star \mathcal{R}_\star$ and $\mathcal{H}_0 = \mathcal{O}\mathcal{R}$ are full-rank factorizations through a $d_\star$-dimensional space. Their left factors have the same column space, so $\mathcal{O} = \mathcal{O}_\star T_0$ for an invertible T_0. Multiplying by a left inverse of $\mathcal{O}_\star$ gives $\mathcal{R} = T_0^{-1}\mathcal{R}_\star$. Consequently, Eq. (5.15) yields

$$\overline{F} = T_0^{-1} F_\star T_0, \qquad \overline{C} = C_\star T_0, \qquad \overline{L} = T_0^{-1} L_\star.$$

Substituting these into $\overline{C}\overline{F}^{j-1}\overline{L}$ proves Proposition 5.9.3. Small singular values make inversion sensitive to coefficient errors; the truncated SVD selects which directions to retain. □

5.A.4 State-space error and long-run statistics

Consider two state processes on one probability space:

$$S_{t+1} = g(S_t) + V_{t+1}, \qquad \widetilde{S}_{t+1} = \widehat{g}(\widetilde{S}_t) + \widetilde{V}_{t+1}.$$

Assume that $\widehat{g}$ is ρ-Lipschitz, $0 \leq \rho < 1$, and that the local coupled RMS error is bounded:

$$\left(\mathbb{E}\left[\|g(S_t) + V_{t+1} - \widehat{g}(S_t) - \widetilde{V}_{t+1}\|_2^2\right]\right)^{1/2} \leq \varepsilon.$$

With $e_t = (\mathbb{E}[\|S_t - \widetilde{S}_t\|_2^2])^{1/2}$, inserting $\widehat{g}(S_t) + \widetilde{V}_{t+1}$ between the next states gives $e_{t+1} \le \varepsilon + \rho e_t$. Iterating gives the same geometric-sum bound used in Lecture 4:

$$e_t \le \rho^{t-1} e_1 + \varepsilon \frac{1-\rho^{t-1}}{1-\rho}.$$

For observations $Z_t = r(S_t) + W_t$ and $\widetilde{Z}_t = \widehat{r}(\widetilde{S}_t) + \widetilde{W}_t$, assume that $\widehat{r}$ is L_r-Lipschitz and

$$\left(\mathbb{E}\left[\|r(S_t) + W_t - \widehat{r}(S_t) - \widetilde{W}_t\|_2^2\right]\right)^{1/2} \le b.$$

The observation discrepancy satisfies

$$d_t := (\mathbb{E}[\|Z_t - \widetilde{Z}_t\|_2^2])^{1/2} \le L_r e_t + b.$$

For an L_f-Lipschitz function f,

$$|\mathbb{E}[f(Z_t)] - \mathbb{E}[f(\widetilde{Z}_t)]| \le L_f d_t.$$

If both observation processes have RMS norm at most B_Z, then

$$\left\|\mathbb{E}[Z_t Z_s^\top] - \mathbb{E}[\widetilde{Z}_t \widetilde{Z}_s^\top]\right\|_{\mathrm{F}} \le B_Z(d_t + d_s). \tag{5.39}$$

To see this, expand the matrix difference using $(Z_t - \widetilde{Z}_t)Z_s^\top + \widetilde{Z}_t(Z_s - \widetilde{Z}_s)^\top$, and apply Cauchy–Schwarz. Uniformly small coupled error therefore controls means, Lipschitz costs, and fixed-lag second moments.

For linear Gaussian models in common state coordinates, take $V_t = \Sigma_V^{1/2}\Xi_t$ and $\widetilde{V}_t = \widehat{\Sigma}_V^{1/2}\Xi_t$ with shared standard Gaussian Ξ_t, and couple the observation noises similarly. If the true states have RMS norm at most B_S, sufficient local bounds are

$$\varepsilon \le \|A - \widehat{A}\| B_S + \|\Sigma_V^{1/2} - \widehat{\Sigma}_V^{1/2}\|_{\mathrm{F}},$$
$$b \le \|C - \widehat{C}\| B_S + \|\Sigma_W^{1/2} - \widehat{\Sigma}_W^{1/2}\|_{\mathrm{F}}.$$

The contraction condition is $\|\widehat{A}\| < 1$. A change of coordinates can express contraction in a weighted Euclidean norm. These bounds control errors in observation statistics uniformly over time.

5.A.5 Cayley–Hamilton

Theorem 5.9.6 (Cayley–Hamilton)**.** *Let $A \in \mathbb{R}^{d\times d}$ have characteristic polynomial*

$$\chi_A(\lambda) = \det(\lambda I - A) = \lambda^d + a_{d-1}\lambda^{d-1} + \cdots + a_1\lambda + a_0.$$

Then

$$\chi_A(A) = A^d + a_{d-1}A^{d-1} + \cdots + a_1 A + a_0 I = 0. \tag{5.34}$$

Proof. The *adjugate* is the transpose of the matrix of cofactors. Its defining determinant identity gives

$$(\lambda I - A)\,\mathrm{adj}(\lambda I - A) = \chi_A(\lambda) I.$$

Write the adjugate, a matrix polynomial of degree at most $d-1$, as $\sum_{j=0}^{d-1} \lambda^j D_j$. Comparing coefficients yields

$$D_{d-1} = I, \qquad D_{j-1} - AD_j = a_j I \ (1 \le j \le d-1), \qquad -AD_0 = a_0 I.$$

Multiply the middle identities on the left by A^j, add them to the last identity and to $A^d D_{d-1} = A^d$. All the D_j terms cancel, leaving $\chi_A(A) = 0$. □

5.B Reproducing the synthetic experiment

This appendix specifies the experiment in Table 1 and Fig. 4. The saved code fits the AR and state-space models to the same observations and evaluates them on the same continuation.

5.B.1 Generator, seeds, and data blocks

Use training sizes $T \in \{256, 1024, 4096\}$ and seeds $s \in \{0, \dots, 49\}$. For each pair, initialize NumPy's `default_rng(s)` and draw independent standard normal $E_0, \dots, E_{N-1}$, where $N = T + 14000$. Using the code's zero-based indexing, generate

$$M_0 = 0, \qquad Z_t = M_t + E_t, \qquad M_{t+1} = 0.98M_t + 0.18E_t, \quad 0 \le t < N.$$

Equivalently, $M_{t+1} = 0.8M_t + 0.18Z_t$, so M_t is the exact conditional mean given $Z_{0:t-1}$. The conditional variance of Z_t is one. The data blocks are:

Block	Indices (inclusive)	Length
Warmup	$0, \dots, 1999$	2000
Training	$2000, \dots, 1999 + T$	T
Calibration	$2000 + T, \dots, 3999 + T$	2000
Test	$4000 + T, \dots, 13999 + T$	10000

Only the training block is used to fit lag coefficients. The warmup observations remain available as history when running the fitted predictors. Their states are propagated from time zero through all blocks without resetting at block boundaries. Calibration and test are consecutive parts of the same trajectory. Reusing each seed across training sizes makes those comparisons dependent; the 50 seeds supply the repeated trials at each size.

5.B.2 Fitting and calibration

Fit a zero-intercept AR(32) by ordinary least squares on all $T - 32$ training targets whose 32 preceding observations are also in the training block:

$$(\widehat{B}_1, \dots, \widehat{B}_{32}) \in \operatorname*{argmin}_{b_1,\dots,b_{32}} \sum_{t=2032}^{1999+T} \left(Z_t - \sum_{j=1}^{32} b_j Z_{t-j} \right)^2 .$$

The implementation uses `numpy.linalg.lstsq` with `rcond=None`, with no centering or regularization. Construct the 16×16 matrices

$$(H_0)_{ij} = \widehat{B}_{i+j-1}, \qquad (H_1)_{ij} = \widehat{B}_{i+j}, \qquad 1 \le i, j \le 16.$$

Retain the leading singular triplet (u_1, σ_1, v_1) of H_0, set $\mathcal{O} = \sqrt{\sigma_1} u_1$ and $\mathcal{R} = \sqrt{\sigma_1} v_1^\top$, and compute

$$\overline{F} = \mathcal{O}^\dagger H_1 \mathcal{R}^\dagger, \qquad \overline{C} = \mathcal{O}_{1,1}, \qquad \overline{L} = \mathcal{R}_{1,1}.$$

All three recovered parameters are scalars. The order 32, block size 16, and rank one are fixed before fitting; there is no hyperparameter search or stability correction.

On each observed history, the two fitted means are

$$\widehat{\mu}_t^{\mathrm{AR}} = \sum_{j=1}^{32} \widehat{B}_j Z_{t-j}, \qquad \widehat{\mu}_t^{\mathrm{SS}} = \overline{C}\,\overline{M}_t, \qquad \overline{M}_{t+1} = \overline{F}\,\overline{M}_t + \overline{L} Z_t, \quad \overline{M}_0 = 0.$$

The first 32 AR predictions are set to zero and occur before every evaluation block. Each model separately estimates its scalar innovation variance on the calibration block $\mathcal{I}$:

$$\widehat{\Omega} = \max\left\{10^{-8}, \frac{1}{2000}\sum_{t\in\mathcal{I}}(Z_t - \widehat{\mu}_t)^2\right\}.$$

This is the uncentered residual second moment. Fitted parameters and this variance remain fixed throughout the test block.

The population reference value for the exact state predictor is one. For AR(32), calculate the optimal residual variance under the *stationary law*, whose distributions are unchanged by shifting all time indices. Before observing a finite window, the state variance is $P_0 = 0.18^2/(1-0.98^2) = 9/11$. If the current state has conditional variance P_j, conditioning the Gaussian pair (M_{t+1}, Z_t) gives

$$P_{j+1} = \frac{0.8^2 P_j}{1+P_j}, \qquad \Omega_{\mathrm{AR}(32)} = 1 + P_{32} \approx 1.000000156927612.$$

The conditional mean is linear in the window, so this is also the population least-squares residual variance. The completion script saves this value; its tests independently check it by solving the population regression using the observation autocovariances.

5.B.3 Metrics and the four panels

For either fitted mean, its test excess mean squared error is

$$\mathrm{MSE}_{\mathrm{excess}} = \frac{1}{10000}\sum_{t\in\mathcal{T}}(\widehat{\mu}_t - M_t)^2,$$

where $\mathcal{T}$ is the test block. It measures conditional-mean error and excludes the irreducible observation variance. Averaging the Gaussian conditional KL $D(\mathcal{N}(M_t,1),\mathcal{N}(\widehat{\mu}_t,\widehat{\Omega}))$ over that block gives exactly

$$\mathrm{KL}_{\mathrm{step}} = \frac{1}{2}\left(\frac{1+\mathrm{MSE}_{\mathrm{excess}}}{\widehat{\Omega}} - 1 + \log\widehat{\Omega}\right).$$

The table reports arithmetic means of these metrics over all 50 seeds at each training size, together with the mean calibration estimate $\widehat{\Omega}$. The AR generator is classified as unstable when its companion matrix has spectral radius at least one. For the state generator the criterion is $|\overline{F} + \overline{LC}| \geq 1$; $|\overline{F}|$ is recorded separately for prediction. Every fit is retained in the summaries.

Figure 4 is constructed as follows.

(a) Plot M_t, $\widehat{\mu}_t^{\mathrm{AR}}$, and $\widehat{\mu}_t^{\mathrm{SS}}$ for the first 160 test steps of the preselected seed 0, $T = 1024$ run: absolute indices 5024 through 5183, relabeled 1 through 160. These are conditional predictions on a common observed history, rather than freely generated trajectories.

(b) Plot each fitted mean minus M_t on the same segment.

(c) For the same fitted models, plot lags $j = 1, \ldots, 64$: the true $0.18(0.8)^{j-1}$, the fitted AR coefficients (zero for $j > 32$), and $\overline{C}\overline{F}^{j-1}\overline{L}$.

(d) Plot one point per trial, with AR excess MSE horizontally and state-predictor excess MSE vertically, on logarithmic axes. There are 150 points, distinguished by training size. The diagonal marks equal errors; crosses identify unstable state generators.

5.B.4 Completion experiments

The script `scripts/lecture05_completions.py` reads the saved fitted parameters and covariance estimates and generates Figs. 5 and 6. The plotting script uses the resulting saved JSON without refitting or drawing new randomness.

The fitted models. Use seed 0 and $T = 1024$, as in Fig. 4. The common prompt is the complete prefix $Z_0, \dots, Z_{5023}$: warmup, training, and calibration observations. Run the true and fitted state predictors through this prefix from their zero initial states; initialize the AR model with its final 32 observations. Keep all fitted parameters and calibration variances fixed. Generate three 300-step continuations using the 3×300 array of standard normal draws from NumPy's `default_rng(20260916)`. Every model uses the same row of this array for the corresponding path, scaled by its own innovation standard deviation. Subsequent state updates use the model's generated observations.

The prediction bands are computed analytically. For a scalar innovations model with parameters a, c, ℓ, ω and predictive state r after the prompt, number future observations by $h = 1, 2, \dots$. Under its conditional continuation law,

$$\mathbb{E}[\widetilde{Z}_h] = ca^{h-1}r, \qquad \mathrm{Var}(\widetilde{Z}_h) = \omega \left(1 + (c\ell)^2 \sum_{j=0}^{h-2} a^{2j}\right),$$

with an empty sum at $h = 1$. For an AR model with coefficients b_j, define its response to one unit of innovation by

$$\psi_0 = 1, \qquad \psi_k = \sum_{j=1}^{\min(k,p)} b_j \psi_{k-j}.$$

Its future mean follows the AR recurrence with future innovations set to zero, and its variance at step h is $\omega \sum_{k=0}^{h-1} \psi_k^2$. Each band is the conditional mean plus or minus 1.959963984540054 conditional standard deviations. These are pointwise prediction intervals given the original prompt.

The constant hidden state. The reference model is $U \sim \mathcal{N}(0, 1)$, $Z_t = U + W_t$, with independent iid $W_t \sim \mathcal{N}(0, 1)$. Prescribe the prompt $z_i = 2 + \xi_i$, $1 \le i \le 1000$, using standard normal draws from `default_rng(0)`. The level 2 is fixed by the experimental design; each model's continuation law is conditioned on this supplied prompt. The full-history posterior has $m_n \approx 1.9500$, $v_n = 1/1001$. Every future observation has conditional mean m_n and variance $1 + v_n$ given this prompt.

For coupling with AR innovations, generate the full-history continuation by its exact sequential Gaussian predictor. Starting from $(m, v) = (m_n, v_n)$, at each step draw

$$z = m + \sqrt{1 + v}\,\xi, \qquad m_{\text{next}} = m + \frac{v}{1 + v}(z - m), \qquad v_{\text{next}} = \frac{v}{1 + v}.$$

This sequential construction has the same joint continuation law as drawing U once from its posterior and then adding independent observation noises. The fixed-window model is the AR(10) recurrence in Section 5.7.2, with every coefficient $1/11$ and innovation variance $12/11$. Both use the same 3×400 array of standard normal draws from `default_rng(20260917)`. Their bands are obtained from the exact conditional variances above. All parameters are known in this comparison; neither model is fitted to this one prompt. The full-history state and true-process panels use the same conditional law and the same noise draws, so their means, bands, and paths agree exactly. Both figures order the panels as conditional means, true-process continuations, AR continuations, and state-model continuations.

5.B.5 Saved artifacts and reproduction commands

From the repository root, run:

```
python3 scripts/lecture05_synthetic.py
python3 scripts/lecture05_completions.py
python3 scripts/test_lecture05_synthetic.py
python3 scripts/test_lecture05_completions.py
python3 scripts/build_lecture_notes.py documents/scribes/lecture05.tex
```

The experiment requires Python 3.9 or later and NumPy; the saved run used Python 3.9.6 and NumPy 1.22.0. It writes `documents/figures/lecture05-synthetic.json`, containing all 150 fits and metrics, the displayed trajectory data, configuration, software versions, and SHA-256 hashes of both experiment source files. The adjacent `lecture05-synthetic-results.tex` and `lecture05-synthetic-plots.tex` are generated from these results. PGFPlots renders the figure during the lecture build.

The completion script writes `lecture05-completions.json` in the same directory. It contains both prompts, every shared noise draw, all sampled paths, exact conditional means and variances, prediction bands, source hashes, and the input-data hash. Its generated figures are saved as `lecture05-completion-fit.tex` and `lecture05-completion-memory.tex`. To regenerate only the plots from the saved JSON, run

```
python3 scripts/lecture05_synthetic_plots.py
python3 scripts/lecture05_completion_plots.py
```

The plotting command uses only saved results. Output-directory options are documented in `scripts/lecture05_synthetic.md`. Numerical rounding may depend on NumPy and linear-algebra library versions; the JSON records the environment of the displayed run.

References

Naman Agarwal, Daniel Suo, Xinyi Chen, and Elad Hazan. Spectral State Space Models. 2024. https://arxiv.org/abs/2312.06837v4.

Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. Annals of Statistics, 45(1):77–120, 2017. https://doi.org/10.1214/16-AOS1435.

Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory, 44(6):2743–2760, 1998. https://doi.org/10.1109/18.720554.

Aviv Bick, Kevin Y. Li, Eric P. Xing, J. Zico Kolter, and Albert Gu. Transformers to SSMs: Distilling Quadratic Knowledge to Subquadratic Models. NeurIPS, 2024. https://arxiv.org/abs/2408.10189.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. 2024. https://arxiv.org/abs/2405.21060v1.

Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. Ten Steps of EM Suffice for Mixtures of Two Gaussians. Proceedings of the 2017 Conference on Learning Theory, PMLR 65:704–710, 2017. https://proceedings.mlr.press/v65/daskalakis17b.html.

Frank Dellaert and Michael Kaess. Factor Graphs for Robot Perception. Foundations and Trends in Robotics, 6(1–2):1–139, 2017. https://www.cs.cmu.edu/~kaess/pub/Dellaert17fnt.html.

Evan Dogariu, Anand Brahmbhatt, and Elad Hazan. Universal Learning of Nonlinear Dynamics. 2025. https://arxiv.org/abs/2508.11990.

Randal Douc, Eric Moulines, and Jimmy Olsson. Long-term stability of sequential Monte Carlo methods under verifiable conditions. Annals of Applied Probability, 24(5):1767–1802, 2014. https://arxiv.org/abs/1203.6898.

Arnaud Doucet and Adam M. Johansen. A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. Handbook of Nonlinear Filtering, 2009. Author's tutorial.

Dynamax contributors. State Space Model API: EM and SGD fitting. Documentation, accessed September 15, 2026. https://probml.github.io/dynamax/api.html.

Elad Hazan, Karan Singh, and Cyril Zhang. Learning Linear Dynamical Systems via Spectral Filtering. 2017. https://arxiv.org/abs/1711.00946.

B. L. Ho and Rudolf E. Kalman. Effective construction of linear state-variable models from input/output functions. Regelungstechnik, 14:545–548, 1966. https://doi.org/10.1524/auto.1966.14.112.545.

Chi Jin, Yuchen Zhang, Sivaraman Balakrishnan, Martin J. Wainwright, and Michael I. Jordan. Local Maxima in the Likelihood of Gaussian Mixture Models: Structural Results and Algorithmic Consequences. 2016. https://arxiv.org/abs/1609.00978.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. ICML, 2020. https://proceedings.mlr.press/v119/katharopoulos20a.html.

Tor Lattimore and Csaba Szepesvári. Bandit Algorithms. Cambridge University Press, 2020. https://tor-lattimore.com/downloads/book/book.pdf.

Tien-Yien Li and James A. Yorke. Period Three Implies Chaos. American Mathematical Monthly, 82(10):985–992, 1975. https://doi.org/10.1080/00029890.1975.11994008.

Anna Mikusheva; Paul Schrimpf, scribe. Filtering. State space models. Kalman filter. MIT 14.384, Lecture 21, revised November 24, 2009. MIT OpenCourseWare lecture notes.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. ICML, 2013. https://proceedings.mlr.press/v28/pascanu13.html.

Changyan Shi, Tim Pearce, Manan Tomar, Siddhartha Sen, and John Langford. Hierarchical Latent Prediction for Language Models. 2026. https://arxiv.org/abs/2608.05806.

Anastasios Tsiamis and George J. Pappas. Finite Sample Analysis of Stochastic System Identification. CDC, 2019. https://arxiv.org/abs/1903.09122.

Anastasios Tsiamis and George J. Pappas. Online Learning of the Kalman Filter with Logarithmic Regret. 2020. https://arxiv.org/abs/2002.05141.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. The Mamba in the Llama: Distilling and Accelerating Hybrid Models. NeurIPS, 2024. https://arxiv.org/abs/2408.15237.

Sumio Watanabe. A widely applicable Bayesian information criterion. Journal of Machine Learning Research, 14:867–897, 2013. https://www.jmlr.org/papers/v14/watanabe13a.html.

C. F. Jeff Wu. On the Convergence Properties of the EM Algorithm. Annals of Statistics, 11(1):95–103, 1983. https://doi.org/10.1214/aos/1176346060.