HORIZON-INDEPENDENT INFERENCE FROM SPARSE MARKOV OBSERVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study *horizon-independent inference*: maximum-likelihood estimation of monotone degrading Markov systems from sparse n-step observations whose computational cost is independent of the maximum observation horizon N. This is a challenging regime: multi-step probabilities are complex functions of the parameters, making likelihood optimization intractable as N grows under standard formulations. We exploit the monotone bidiagonal structure to derive a *parameterized* diagonalization $\bar{P}(\vec{p}) = U(\vec{p})D(\vec{p})V(\vec{p})$ whose factors are closed-form in the $T-1$ unknown stay-probabilities (T is the sink state), enabling each n-step entry $\bar{P}^n_{x,y}(\vec{p})$ to be evaluated in $O(T)$ time with algebraically exact analytic gradients — reducing total cost from $O(T^2N)$ to $O(T^2 + TH)$, where H is the number of distinct observed tuples. While baselines fail for considered settings beyond $N = 500$, BDS scales to $N = 100{,}000$ in single-digit seconds. Synthetic evaluations show below 1% MAPE with a few thousand observations; real-world results confirm interpretable recovery and competitive predictive accuracy.

1 INTRODUCTION

Managing the lifecycle of degrading assets — knowing how to use them, how intensively, and when to replace them — has become increasingly important across industries dealing with high-value, high-wear products: batteries in electric vehicles, mechanical components in manufacturing, insured fleets, medical devices. In all these settings the underlying aging is stochastic and usage-dependent, and practitioners must *learn* its transition dynamics from data in order to support prediction, lifetime estimation, and operational planning. The data, however, is typically sparse: products are inspected infrequently, successive observations span heterogeneous time intervals, and the maximum horizon N between observations can be large.

This sparsity creates a computational bottleneck. Fitting a Markov transition model requires evaluating n-step transition probabilities $\bar{P}^n_{x,y}(\vec{p})$ as a function of the unknown parameters $\vec{p}$ at every step of the optimizer. Standard approaches fail each in a different way: full $T\times T$ variable models (GSS) are intractable beyond $N = 10$; even smaller band-matrix models either exhaust memory (BSS$^{(1)}$) or suffer from likelihood evaluation costs as N grows (BSS$^{(2)}$); EM produces higher estimation error at small N and scales super-linearly in N at large N. None is applicable in the sparse long-horizon regime $H \ll TN$ that characterises real data with infrequent or delayed observations.

We resolve this bottleneck for *monotone bidiagonal* Markov chains — a natural model whenever states are ordered, transitions are gradual, and recovery is negligible. Many degradation processes assume monotonicity by construction: battery capacity fade, mechanical wear, crack growth, corrosion, and disease staging all progress unidirectionally through ordered states. Our central result (Theorem 4.1) is a novel *parameterized* diagonalization $\bar{P}(\vec{p}) = U(\vec{p})\, D(\vec{p})\, V(\vec{p})$, in which all three factors are explicit closed-form functions of the $T-1$ unknown probabilities, valid as long as the p_i are pairwise distinct. This is categorically different from classical numerical diagonalization of a fixed matrix: because U, D, V are symbolic in $\vec{p}$, any required entry $\bar{P}^n_{x,y}(\vec{p})$ can be evaluated and differentiated in $O(T)$ time independently of n. The resulting BDS algorithm evaluates the full likelihood in $O(T^2 + TH)$, where H counts distinct observed (x, y, n, a) tuples, with no dependence on N (Theorem 4.3). While BSS$^{(1)}$ and BSS$^{(2)}$ are limited (to $N = 20$ and $N = 500$), BDS solves the same problem at, e.g., $N = 100{,}000$ in under 7 seconds (Table 1). Once $\hat{\vec{p}}$ is learned, future state distributions, expected lifetime, and replacement policies follow directly (Section 4.2).

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

The resulting estimator is not merely a faster implementation of Markov-chain fitting: it makes maximum-likelihood learning feasible in a data regime (sparse observations with horizons far larger than the state dimension) in which standard likelihood formulations become prohibitive. More broadly, the approach demonstrates an inference paradigm for sparse long-horizon observations: evaluate only the transition entries appearing in the likelihood at $O(T)$ per entry, rather than propagating the full temporal evolution. We formalize the central algorithmic property that BDS achieves:

Definition 1.1 (Horizon-independent inference)**.** An inference algorithm is *horizon-independent* if its computational complexity depends on the number of observed transition tuples H but not on the maximum observation horizon $N = \max_h n_h$: H controls computation; N does not. This is an arithmetic-complexity statement: p_k^n is unit cost. Numerical precision as a function of n is a separate property, analyzed in Appendix M.

We are not aware of prior methods that achieve horizon-independent likelihood and gradient evaluation for this exact setting — parameter-dependent transition probabilities, sparse selected entries, and arbitrary n-step observations simultaneously. Our contributions are:

- **Parameterized diagonalization (Theorem 4.1).** We derive explicit $\lambda/\kappa/\theta$ product formulas for $U(\vec{p})$, $D(\vec{p})$, $V(\vec{p})$ in the $T-1$ unknown probabilities, i.e. a parameterized spectral representation of the entire bidiagonal family, not a per-iterate numerical decomposition. U and V are precomputed once in $O(T^2)$ and amortized across all H observed tuples; each entry $\bar{P}^n_{x,y}(\vec{p})$ then evaluates in $O(T)$ independently of n, which yields an amortization unavailable to standard numerical matrix-function methods (see Appendix D).
- **Horizon-independent complexity (Theorem 4.3, Corollary 4.4).** The total likelihood and gradient cost is $O(T^2+TH)$, with no dependence on the maximum observation horizon N: two datasets with equal H but arbitrarily different N impose the same asymptotic cost on BDS. Differentiating $\bar{P}^n_{x,y}(\vec{p})$ via the product rule yields exact closed-form gradients at the same cost, without differentiating through n matrix multiplications or finite-differencing.
- **Extensions and empirical validation.** We show three extensions: absorbing failure states, variable reduction (both $O(T^2+TH)$), and belief-distribution observations ($O(T^2)$ per obs.). On synthetic benchmarks BDS is the only tractable method across $N = 1$ to 100,000 among the baselines; on real-world data it recovers interpretable aging dynamics for NASA batteries and outperforms the full GSS model at low sample sizes on XJTU-SY bearings.

Sections 2–6 cover related work, model, algorithm, evaluation, and conclusions; proofs, extensions, and supplementary experiments are in Appendices A–M.

2 Related Work

Estimation of Transition Probabilities in Discrete Markov Chains For discrete-time Markov models, the standard approach identifies the transition matrix P by maximizing the log-likelihood. For 1-step data this is well-studied and various extensions exist (Li et al., 2018). Estimation from n-step data is harder: the entries of P^n become intractable for large state spaces and long horizons. When direct likelihood optimization is infeasible, the EM algorithm (Dempster et al., 1977) is commonly used as an approximation (Craig & Sendi, 2002). However, EM struggles when n is large and observations are scarce relative to the number of parameters.

Three direct baseline methods span this landscape. The general-matrix solver (**GSS**) maximizes the log-likelihood over a full transition matrix P with $T(T-1)$ free variables and matrix powers P^n; it is exact but does not scale with T or N. The banded-matrix solver (**BSS**) restricts to the upper-bidiagonal structure $\bar{P}$ with only $T-1$ variables yet incurs a memory or computational bottleneck. The expectation-maximization approach (**EM**) avoids explicit P^n auxiliary variables via latent transition counts, but converges slowly and has runtimes growing with N. Our BDS approach avoids all three failure modes via closed-form diagonalization of $\bar{P}$ (Theorem 4.1); Appendix D compares BDS against various standard methods — with wall-clock data in Table 4.

Data-Driven Degradation Modeling Degradation processes across diverse domains — batteries, mechanical components, insured assets — have been modelled with Markov chains, but most existing data-driven approaches require large amounts of data, do not support usage dependencies, or assume states are directly observable (Dai et al., 2019; Zhu et al., 2023; Kim et al., 2024). A further complication in practice is that the current degradation state is often not directly observable but only estimable as a distribution — a setting our belief-distribution extension (Section 4.3) supports.

Aging Products and Lifecycle Management Lifecycle management has grown in importance with circular economy models (Guo et al., 2021; Corno & Pozzato, 2020). Exact methods such as linear programming and dynamic programming (Weitzel & Glock, 2018) as well as meta-heuristics (Hossain et al., 2019; Collath et al., 2022) have been applied, but most assume deterministic or highly stylized dynamics (Jafari et al., 2018; Figge et al., 2022), limiting the use of data-driven aging estimates. To the best of our knowledge, no existing approach simultaneously supports large n-step intervals and sparse, heterogeneous observations at the scale we consider.

3 Approach to Estimate Aging Dynamics

3.1 Model Description and Data

We consider T discrete states $s \in \{1, \dots, T\}$, A usage levels $a \in \{1, \dots, A\}$, and K observations, each recording a pre-state $x_1(k)$, a post-state $y(k)$, and an (average) usage level $x_2(k)$ assumed to be constant throughout the usage interval length $n(k)$.

3.2 Problem Description and Markov Chain Model for Steady Aging Dynamics

General Aging Stochastic aging for one period is described by a $T \times T$ transition matrix $P(a)$, which may depend on the usage level a and has up to $T^2 - T$ free parameters per action in general.

Steady Aging When considering short periods of time and steady aging processes (without shocks), the matrix P can become a *band matrix*. Here, (i) as long as periods are short, the new state of an aging product will be close to the old state, (ii) the state does not improve, and (iii) there is a terminal state T. This special class of aging dynamics is supported by real-world data for batteries, see Appendix H. Hence, we focus on band matrices $\bar{P}$ characterized by $T-1$ transition probabilities $\vec{p}(a) = (p_1(a), ..., p_{T-1}(a))$, $a = 1, ..., A$, ($T-1$ variables for all usages a), i.e., we use ($\mathbf{1}_{\{\cdot\}}$ is the indicator function (and rows add up to 1)):

$$\bar{P} = \bar{P}_{i,j}(\vec{p}(a)) = \begin{cases} p_i(a) & i = j < T \\ 1 - p_i(a) & j = i + 1 \leq T \\ \mathbf{1}_{\{i=j=T\}} & else, \end{cases} \tag{1}$$

Our goal is to estimate $\vec{p}(a)$, $a = 1, \dots, A$, from the observed data (Section 3.1), making the n-step probabilities $\bar{P}^n_{i,j}(\vec{p}(a))$ tractable at large N.

Remark 3.1 (Model scope). The bidiagonal structure captures gradual, monotone state deterioration. Occasional catastrophic failures can be represented by allowing direct transitions to an absorbing failure state, i.e. $P_{i,T} > 0$ for all $i < T$. The transient submatrix remains upper-bidiagonal, preserving the spectral structure and $O(T^2{+}TH)$ complexity; the parameter space expands to $2T{-}3$ and $(1{-}p_k)$ is replaced by q_k in the eigenvector λ-products (Proposition C.1, Appendix C).

4 Our Solution Approach

In this section, we propose our solution approach to estimate transition dynamics from few data (Section 4.1). Further, we summarize consequent analyses (Section 4.2) as well as generalizations and extensions of the model (Section 4.3).

4.1 Diagonalization Techniques for Steady Aging

Objective and sufficient statistics. We estimate $\bar{P}$ by maximizing the log-likelihood

$$\max_{\vec{p}(a) \in (0,1)^{T-1},\, a=1,\dots,A} \sum\nolimits_{k=1,\dots,K} \log\left(\bar{P}^{n(k)}_{x_1(k),y(k)}\left(\vec{p}(x_2(k))\right)\right). \tag{2}$$

We reparametrize via $p_i(a) = \sigma(\beta_i(a))$, $\beta_i(a) \in \mathbb{R}$, to keep $p_i \in (0,1)$ unconstrained during optimization. The likelihood depends on the K raw observations only through the counts $c(x, y, n, a) = \#\{k : x_1(k){=}x,\, y(k){=}y,\, n(k){=}n,\, x_2(k){=}a\}$ for $x, y \in \{1, \dots, T\}$, $n \in \{1, \dots, N\}$, $a \in \{1, \dots, A\}$, so it reduces to a sum over the $H \leq K$ distinct (x, y, n, a) tuples, $\sum_k \log \bar{P}^{n(k)}_{x_1(k),y(k)} = \sum_{(x,y,n,a)} c(x, y, n, a) \cdot \log \bar{P}^n_{x,y}(\vec{p}(a))$. Each distinct tuple requires exactly one evaluation of $\bar{P}^n_{x,y}$, so H (not K) is the effective data size that determines computational cost.

The challenge and our approach. Standard formulations compute $\bar{P}^n$ by raising $\bar{P}$ to the n-th power, requiring $O(T^2N)$ memory for n-step auxiliary variables or $O(NT^3)$ per gradient step — both intractable at large N. A natural fix is to diagonalize $\bar{P}$: once $\bar{P} = UDV$, the n-step matrix is UD^nV, costing only $O(T)$ per entry. The difficulty is that we need this factorization as an *explicit function* of the $T-1$ unknown parameters $\vec{p}$ — so that gradients with respect to $\vec{p}$ are available in closed form. A numerical eigenvector decomposition recomputed at each solver iterate costs $O(T^3)$ per step and yields no symbolic derivatives. While eigenvalues of an upper-bidiagonal matrix are its diagonal entries and eigenvectors follow by a simple recurrence, we derive explicit $\lambda/\kappa/\theta$ product formulas for $U(\vec{p})$ and $V(\vec{p})$ simultaneously in $\vec{p}$ — a parameterized representation valid wherever the p_i are pairwise distinct, not a per-iterate numerical decomposition. The key consequence is amortization: U and V are precomputed once in $O(T^2)$ and shared across all H tuples, so each entry $\bar{P}^n_{x,y}(\vec{p}) = \sum_k U_{x,k}\, p^n_{k-1}\, V_{k,y}$ evaluates in $O(T)$ with n as a scalar exponent only.

Theorem 4.1 (Parameterized diagonalization of the monotone bidiagonal family)**.** *Let T be a positive integer, let $p_i \in (0,1)$ for $i = 1,\dots,T-1$, and assume $p_1,\dots,p_{T-1}$ are pairwise distinct. Define $p_0 := 1$. Then the $T \times T$ bidiagonal matrix $\bar{P}(\vec{p})$ of the form (1) can be expressed as*

$$\bar{P}(\vec{p}) = U(\vec{p}) \cdot D(\vec{p}) \cdot V(\vec{p}), \tag{3}$$

where, $i, j = 1, ..., T$,

$$D_{i,j}(\vec{p}) = \begin{cases} 1 & i = j = 1 \\ p_{i-1} & i = j > 1 \\ 0 & else \end{cases} \tag{4}$$

$$U_{i,j}(\vec{p}) = \begin{cases} 1 & j = 1 \vee j = i+1 \\ 0 & j > 1 \wedge i \geq j \\ \frac{\lambda(i,j-2)}{\kappa(i,j-1)} \cdot (-1)^{i+j-1} & else \end{cases} \tag{5}$$

$$V_{i,j}(\vec{p}) = \begin{cases} 1_{\{j=T\}} & i = 1 \\ 1 & i = j+1 \\ 0 & i > j+1 \\ -1 & i = j = T \\ \frac{\lambda(i,T-1)}{\theta(i-1,T-1)} \cdot (-1)^{i+j-1} & 1 < i < j = T \\ \frac{\lambda(i-1,j-1)}{\theta(i-1,j)} \cdot (-1)^{i+j-1} & else \end{cases} \tag{6}$$

with $\lambda(i,j) := \prod_{k=i,\dots,j} (p_k - 1)$, $\kappa(i,j) := \prod_{k=i,\dots,j-1} (p_j - p_k)$, $\theta(i,j) := \prod_{k=i+1,\dots,j} (p_i - p_k)$.

Proof. Empty products equal 1. The full proof is given in Appendix A. □

Remark 4.2 (Applicability and robustness). BDS operates in numerically challenging territory by design: the spectral representation requires pairwise-distinct p_i, yet real data and optimizer trajectories may approach or even target coincident values. Two structural properties govern this territory; a closed-form fix and empirical evidence complete the picture: *(i)* with standard evaluation, exact coincidence produces a $0/0$ return that optimizers treat as a failed step — a natural repulsion keeps iterates away in practice; with the L'Hôpital-based stable evaluation (20) (Appendix M), the limit is resolved analytically from the BDS closed form and the optimizer operates smoothly at any gap; *(ii)* in the well-conditioned interior the likelihood gradient dominates and guides the optimizer toward the MLE — the boundary between regimes is quantified in Appendix M. Table 27 confirms sub-2% estimation error across 20 seeds even when the true parameters coincide exactly.

Complexity. The explicit parameterization of Theorem 4.1 allows the H distinct transition probabilities in the likelihood to be evaluated one at a time, each in $O(T)$, without forming any full n-step matrix. The bidiagonal case (J=1) is the exact closed-form result of this paper; Appendix C shows that the upper-triangular eigenvalue structure and $O(T^2{+}TH)$ complexity extend to fixed J>1, although the telescoping product form of Theorem 4.1 does not.

Theorem 4.3 (Horizon-Independent Likelihood and Gradient Complexity)**.** *Under the assumptions of Theorem 4.1, let H_a denote the number of distinct (x, y, n) triples observed under action a, and let $H := \sum_{a=1}^{A} H_a \leq K$. As shown above, the log-likelihood is a sufficient-statistic sum over distinct tuples; H is therefore the effective data size after aggregation, and $H = K$ only when every observation is unique. The BDS factors $U(\vec{p})$ and $V(\vec{p})$ can be precomputed in $O(T^2)$ time per*

action, after which all H_a transition probabilities required under action a are evaluated in $O(TH_a)$ time, independent of the largest observed horizon N. The total likelihood cost is $O(AT^2 + TH)$; for a fixed number of actions this is $O(T^2 + TH)$. Moreover, the full analytic gradient $\nabla_{\vec{p}} \log \ell$ — all $T-1$ components simultaneously — can be computed at the same asymptotic cost $O(T^2 + TH)$, without differentiating through n successive matrix multiplications or finite-differencing. For the BSS formulation that represents $\bar{P}^1, \ldots, \bar{P}^N$ through recursive matrix multiplication, the corresponding likelihood cost is $O(T^2 N)$ per action, which involves N.

Proof. See Appendix B. □

Corollary 4.4 (Horizon-independent complexity)**.** *For any two datasets $\mathcal{D}$ and $\mathcal{D}'$ with the same state dimension T and the same number of distinct observed transition tuples H, but with arbitrarily different maximum horizons $N = \max_h n_h$ and $N' = \max_h n'_h$, BDS evaluates the log-likelihood and its full gradient via closed-form expressions, at identical asymptotic cost $O(T^2 + TH)$.*

Proof. Immediate from Theorem 4.3: the cost $O(T^2 + TH)$ depends on T and H only. The only operation involving individual observation horizons n_h is the scalar exponentiation $p_{k-1}^{n_h}$ in the spectral sum $\bar{P}_{x,y}^{n_h} = \sum_k U_{xk}\, p_{k-1}^{n_h}\, V_{ky}$; its cost is $O(1)$ per eigenvalue and observation, contributing to the $O(TH)$ term. No loop over $n = 1, \ldots, N$ is performed. □

Remark 4.5 (Analytic gradients). The gradient formula follows from the product rule applied to $\bar{P}_{x,y}^n(\vec{p}) = [U(\vec{p})\, D(\vec{p})^n\, V(\vec{p})]_{x,y}$: $\frac{\partial \bar{P}_{x,y}^n}{\partial p_k} = \left[U'_k D^n V + U\, n D^{n-1} D'_k\, V + U D^n V'_k\right]_{x,y}$. Evaluated naively — forming U'_k, D'_k, V'_k separately for each of the $T-1$ parameters — this would cost $O(T^3)$. Instead, the full gradient vector is obtained in a single pass applying the product rule to the forward computation graph, at cost $O(T^2 + TH)$ matching the forward pass. Appendix B traces this reverse pass stage by stage: the $O(T^2)$ prefix/suffix products for U and V, the $O(T)$ diagonal powers per observation, and the $O(TH)$ likelihood contractions.

4.2 Downstream Inference: State Evolution and Usage-Replacement Policy

Accurate estimates $\hat{\vec{p}}$ unlock several standard downstream quantities directly from the closed-form representation; we summarise the key ones to motivate the value of estimated dynamics.

State Evolution Prediction Given (estimated) usage-dependent transition probabilities $\vec{p}(a)$ predictions regarding future state evolutions require the current state (not a sequence of states as in various time series models). For a closed-loop policy $a(\cdot)$, assemble the policy matrix P^π with rows $P^\pi_{i,\cdot} = \bar{P}_{i,\cdot}(\vec{p}(a(i)))$; the n-step distribution from s is then $e_s^\top (P^\pi)^n$.

Durations and Expected Lifetime By D_s we denote the duration the product remains in a state s, $s = 1, ..., T-1$, under policy $a(s)$. The probability that D_s equals n periods is (geometric distribution), $n \geq 1$, $P(D_s = n) = (1 - p_s(a(s))) \cdot p_s(a(s))^{n-1}$. Hence, the expected duration in state s is $E(D_s) = 1/(1 - p_s(a(s)))$, and the expected lifetime (state 1 to T) is $E(L) = \sum_{s=1}^{T-1} E(D_s)$.

Usage-Replacement Policy The estimated $\hat{\vec{p}}$ plugs directly into a standard infinite-horizon MDP (Puterman, 2014); the bidiagonal structure further yields a closed-form stationary distribution under any control-limit replacement policy (Proposition E.1, Appendix E), enabling direct analytical computation of long-run replacement frequencies and expected cycle lengths.

4.3 Model Extensions and Generalizations

(i) Variable Reduction: Structured Parameter Sharing via Basis Function Models A structured extension reduces the number of estimated parameters: the $T-1$ state-specific transition probabilities $p_i(a)$ need not be estimated as independent free parameters. When the aging rate varies smoothly across states — a natural prior for physical degradation processes — the $\beta_i(a)$ can be expressed through a small set of $B \ll T-1$ basis function coefficients:

$$\beta_i(a) = \alpha_0 f_0(i,a) + \alpha_1 f_1(i,a) + \cdots + \alpha_{B-1} f_{B-1}(i,a), \quad i = 1, \ldots, T-1, \tag{7}$$

and the optimizer acts on the B coefficients $\alpha_b \in \mathbb{R}$ only. This reparametrization encodes the prior that aging rates share a common functional form across states, reducing the parameter dimension and

making the fitted model directly interpretable in terms of the chosen features (e.g. state index, square root of state index). Crucially, basis function reduction composes directly with the parameterized diagonalization: $\bar{P}^n_{x,y}(\vec{p}(\vec{\alpha}))$ remains an $O(T)$ evaluation for any fixed B, so the computational advantage of BDS is fully preserved. Analytic gradients with respect to $\vec{\alpha}$ follow via the chain rule through the sigmoid and Remark 4.5; the gradient cost remains $O(T^2 + TH)$.

(ii) Extension: Absorbing Failure States The bidiagonal structure is preserved when catastrophic failures are included. Allowing direct transitions from any transient state to an absorbing failure state leaves the transient submatrix Q upper-bidiagonal with diagonal $\vec{p}$ and super-diagonal $\vec{q}$. The eigenvalues of Q are again $p_1, \ldots, p_{T-1}$, but the eigenvectors depend on both $\vec{p}$ and $\vec{q}$ — specifically, $(1 - p_k)$ is replaced by q_k in the λ-products of Theorem 4.1 (Proposition C.1, Appendix C). The n-step failure probabilities remain $O(T)$ to evaluate, and the log-space implementation (Appendix M) extends without modification.

(iii) Generalization: Input Data as Belief Distributions Our model extends naturally to settings where pre-states, post-states, or usage levels are observed as probability distributions rather than point values. Each observation k is now a triple of distributions: $\vec{x}_1(k) \in \Delta^T$ (pre-state), $\vec{x}_2(k) \in \Delta^A$ (usage), and $\vec{y}(k) \in \Delta^T$ (post-state). When pre-states, post-states, and usage levels are observed as distributions rather than point values, the log-likelihood objective (2) generalizes to a KL minimization. Given inputs $\vec{x}_1(k)$, $\vec{x}_2(k)$, $\vec{y}(k)$ and interval $n(k)$, the fitted post-state distribution is $\hat{y}(k, j; \vec{p}) = \sum_a x_2(k, a) \sum_i x_1(k, i)\, \bar{P}^{n(k)}_{i,j}(\vec{p}(a))$, $j = 1, \ldots, T$, and we estimate $\vec{p}(a)$ for all a by minimizing the sum of KL divergences over observations (horizon-independence with respect to N is preserved; the per-observation cost rises from $O(T)$ to $O(T^2)$ since each $\hat{y}(k)$ requires a vector–matrix contraction, giving $O(AT^2 + KT^2)$ per gradient call):

$$\min_{\substack{\vec{p}(a) \in (0,1)^{T-1} \\ a = 1, \ldots, A}} \sum_{k=1}^{K} \sum_{a=1}^{A} x_2(k, a) \sum_{j=1}^{T} y(k, j)\, \log\Big(y(k, j) \;/\; \textstyle\sum_{i=1}^{T} x_1(k, i)\, \bar{P}^{n(k)}_{i,j}(\vec{p}(a)) \Big) \,. \tag{8}$$

5 Evaluation

5.1 Baseline Approaches

To estimate dynamics from heterogeneous n-step data, we compare our approach against the following established standard benchmark approaches:

- **GSS**: maximizes (2) for full $T \times T$ matrix P; n-step entries via matrix powers
- **BSS**$^{(1)}$: maximizes (2) for $\bar{P}$; gradient through N matrix powers ($O(AT^3N)$ memory)
- **BSS**$^{(2)}$: maximizes (2) for $\bar{P}$; recomputes $\hat{P}^n$ via matrix products per evaluation ($O(N \cdot T^3)$); because differentiating (2) costs $O(N \cdot T^3)$ per gradient (the same order as the likelihood); thus we use finite-difference gradients ($T-1$ perturbations per step)
- **EM**: iterative lower-bound maximization for $\bar{P}$; $O(N \cdot K \cdot T^2)$ per iteration
- **BDS (ours)**: maximizes (2) for $\bar{P}$; n-step entries via Theorem 4.1 in $O(T)$, indep. of N.

5.2 Scalability of our Approach compared to Standard Methods

Example 5.1. **(Applicability and Runtimes)** To study scalability we considered problems with increasing N (for $T = 50$, $A = 1$, $K = 2\,000$, and P a band matrix as defined in Example 5.2).

Results. Table 1 reports runtimes and MAPE for increasing N. EM is excluded: its MAPE is 8.6% at $N = 1$ and its runtime super-linear in N (0.1 s at $N = 50$, 51.5 s at $N = 1\,000$) because each iteration must precompute all N matrix powers and sweep over $K \cdot T^2$ terms. The remaining baselines each fail at different points in this configuration: GSS DNFs at $N = 10$ due to $O(T^2)$ gradient cost; BSS$^{(1)}$ runs out of memory from $N = 20$ due to forward-mode AD dual-number memory ($O(AT^3N)$); BSS$^{(2)}$ DNFs beyond $N = 500$ as its $O(N)$ matrix-product chain becomes prohibitive. Among the evaluated baselines, BDS is the only method applicable across the full range: runtime is horizon-independent (the $T \times T$ factors U, V do not grow with N, consistent with Corollary 4.4). MAPE stays below 3% throughout, and all solves complete in under 10 s.

Table 1: Scalability comparison for increasing N (Example 5.1). GSS: L-BFGS on the full $T\times T$ stochastic matrix. BSS$^{(1)}$: L-BFGS using n-step auxiliary variables; BSS$^{(2)}$: L-BFGS with the $T-1$ bidiagonal parameters; likelihood via n-step matrix products ($O(N \cdot T^3)$ per evaluation). BDS: L-BFGS with the $T-1$ bidiagonal parameters; likelihood via Theorem 4.3. MAPE: median over 5 independent runs. OOM = out of memory; DNF = did not finish within 300 seconds.

				BDS (ours)	
N	GSS	BSS$^{(1)}$	BSS$^{(2)}$	time	MAPE
5	170.0 s	1.8 s	1.5 s	0.2 s	2.6 %
10	DNF	2.3 s	2.7 s	0.4 s	1.5 %
20	DNF	OOM	5.8 s	0.4 s	1.0 %
50	DNF	OOM	18.8 s	2.2 s	0.5 %
100	DNF	OOM	66.0 s	2.0 s	0.3 %
200	DNF	OOM	102.6 s	7.0 s	0.2 %
500	DNF	OOM	DNF	4.5 s	0.2 %
1 000	DNF	OOM	DNF	7.9 s	0.1 %
2 000	DNF	OOM	DNF	9.6 s	0.1 %
5 000	DNF	OOM	DNF	5.3 s	1.5 %
10 000	DNF	OOM	DNF	5.1 s	0.5 %
20 000	DNF	OOM	DNF	7.3 s	0.6 %
50 000	DNF	OOM	DNF	6.2 s	1.7 %
100 000	DNF	OOM	DNF	6.2 s	2.6 %

Horizon-independent complexity. The experiment reveals a property stronger than scalability in N. Consider two datasets of equal size K, state-space dimension T, and distinct-tuple count H, differing only in their observation horizon: Dataset A with $n \in \{1, \dots, 10^2\}$ and Dataset B with $n \in \{10^5, \dots, 10^5 + 10^2 - 1\}$. BDS imposes the same asymptotic likelihood cost on both, because $\bar{P}^n_{x,y}(\vec{p})$ is evaluated in $O(T)$ time for any n and the total cost $O(T^2 + TH)$ contains no N (Corollary 4.4). The complexity is governed by the observed information H, not by the interval $[1, N]$ in which it falls. Figure 1 isolates this directly: holding $T = 20$ and $K = 500$ fixed and varying N over seven orders of magnitude, BDS evaluation time is flat (≈ 0.1 ms) while BSS$^{(2)}$ grows linearly and is terminated at $N = 10^5$. Numerical limitations, if any, are governed by floating-point precision in p^n_k—a function of eigenvalue separation and hop distance, not of N itself.

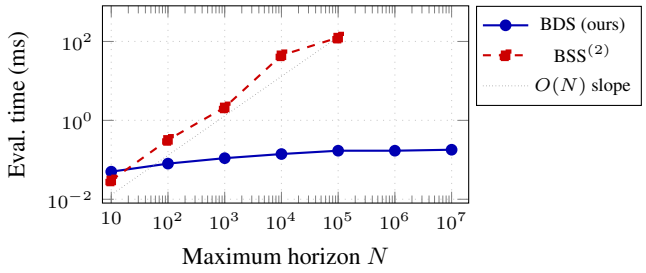

Figure 1: Likelihood evaluation time vs. maximum horizon N (log–log), with $T = 20$, $K = 500$ observations, $H \approx 500$ distinct tuples. BDS (solid) is flat across seven orders of magnitude; BSS$^{(2)}$ (dashed) scales linearly and becomes prohibitive beyond $N = 10^5$. The dotted line marks $O(N)$.

5.3 Results for Synthetic Data (Match and Mismatch Settings)

To test the accuracy of our methodology, we consider an exemplary aging matrix $\bar{P}$ (ground truth) to be revealed. We first consider the case of observable states and actions, cf. Section 3.1. and solve (2) using BDS, which dominates BSS in solve time and EM in both runtime and accuracy. We use Julia/L-BFGS on a notebook with Intel Core Ultra 5 225U, 32 GB RAM.

Example 5.2. **(Accuracy against Synthetic Ground Truth and Impact of Model Parameters)** We consider the Base Case: $A = 1$, $T = 20$, $N = 20$, $G = 1$, with observable pre-states x_1 and actions x_2 using discrete uniform random realizations, i.e. $U(1, T-1)$ and $U(1, A)$; for the interval lengths n, we use $U(1, \lfloor G \cdot N \rfloor)$. Further, for reproducibility, we consider the (true) exemplary probabilities $p_i(a) = 1 - \frac{1}{2} \cdot \frac{a+i}{A+T} \cdot \frac{1}{N^{0.5}}$ and simulate corresponding post-states y for $K = 1\,000$ observations. We

use 10 repeated runs to evaluate average results and their standard deviations. The stay-probabilities $\vec{p}$ are state- and action-dependent: higher usage a and older states accelerate degradation Li et al. (2023), and the values are calibrated to N so that no state drains to zero probability within the observation window.

Results. Accuracy improves monotonically with N (Table 7, Appendix F): larger horizons provide more informative observations at no additional computational cost, consistent with Corollary 4.4. Appendix F provides further sensitivity tables for N, K, T, G, and A.

Example 5.3. **(Robustness to Model Misspecification)** We inject two structural violations into the bidiagonal data-generating process: (i) recovery transitions at rate r (states decrease by one with probability r), and (ii) wider jump bandwidths $J \in \{1, 2, 3\}$ (up to J steps per period). BDS is fit without modification in both cases. For full details and reproducibility, see Appendix G.

Results. Recovery contamination degrades accuracy gracefully: MAPE rises from 0.9% at $r{=}0$ to 1.7% at $r{=}10\%$, and at the NASA-observed rate of 3.6% the mismatch adds only $\approx$0.3 percentage points (Table 10, Appendix G). Wider jump widths are a harder violation: MAPE reaches 41% at $J{=}2$ and 56% at $J{=}3$ (Table 11), confirming that the $J{=}1$ assumption is load-bearing and that the $J{>}1$ extension (Appendix C) should be used when multi-step jumps within a period are plausible.

5.4 Results for Real-world Data

Example 5.4. **(Real-World Battery Data)** We use $A = 1$, $T = 10$, $N = 20$, and NASA battery data with observable states, cf. x_1 and y. For reproducibility, see Appendix H.

Results. The results for using (i) a general aging matrix P and (ii) a band matrix $\bar{P}$ almost coincide; they are summarized in Appendix H.1 (Tables 13 and 14). Both models yield nearly identical estimates, and the fitted general model is strongly concentrated on the bidiagonal structure, confirming that the monotone upper-bidiagonal approximation is reasonable for this dataset.

Example 5.5. **(Real-World Bearing Data)** We use $A = 1$, $T = 10$, $N_{\text{max}} = 30$, and the XJTU-SY bearing run-to-failure dataset Wang et al. (2020) (15 bearings, three load conditions). RMS vibration per epoch is discretized into T states. For full details and reproducibility, see Appendix I.

Results. BDS leads at $K{=}50$ (3.25 vs. 4.60 nats), confirming that the bidiagonal prior helps under genuine data scarcity. From $K{=}200$ onward, GSS overtakes BDS. A condition-stratified analysis (Table 16) confirms this is a genuine structural limitation — GSS leads within every individual condition — rather than a pooling artifact. Full results are in Appendix I (Tables 16, 18).

5.5 Results for Extensions

We evaluate three extensions from Section 4.3: (i) structured parameter-sharing, where the $T{-}1$ free parameters are replaced by $B \ll T{-}1$ basis-function coefficients; (ii) absorbing failure states, where direct transitions from any transient state to a terminal state are permitted; and (iii) belief-distribution inputs, where states and actions are observed only as distributions, using objective (8).

Example 5.6. **(i) Variable Reduction.** We consider the base case Example 5.2 with $B = 3$ features. As described at the end of Section 4.3, within $p_i(a) = \exp(\beta_i(a))/(1 + \exp(\beta_i(a))) = \sigma(\beta_i(a)) \in (0, 1)$, $i = 1, ..., T-1$, $a = 1, ..., A$, we define $\beta_i(a) = \alpha_0 \cdot f_0(i, a) + \alpha_1 \cdot f_1(i, a) + \alpha_2 \cdot f_2(i, a)$ and use the features $f_0(i, 1) := 1$, $f_1(i, 1) := i$, and $f_2(i, 1) := i^{0.5}$ with $B = 3$ associated variables $\alpha_0, \alpha_1, \alpha_2 \in \mathbb{R}$. Further, we vary the number of states T.

Results. Table 19 (Appendix J) demonstrates the structured parameter-sharing contribution. Instead of $T{-}1$ free parameters, BDS now uses only $B = 3$ coefficients $(\alpha_0, \alpha_1, \alpha_2)$ to describe all p_i, $i = 1, \ldots, T{-}1$. Two effects are visible: (i) the number of variables is reduced from $T{-}1$ to $B = 3$, so the optimizer sees a fixed-dimensional surface regardless of T; (ii) estimation error stays at or below 0.4% MAPE across the full range $T = 5, \ldots, 100$ — including at $T = 100$, where the basis regularization improves over the unstructured counterpart. This confirms that basis function reparametrization is a practically effective way to reduce the parameter dimension while maintaining or improving estimation accuracy. When actions are metric (e.g. a usage intensity or frequency), the same reparametrization applies along the action dimension, reducing the $(T{-}1) \cdot A$ parameter space to B coefficients regardless of A.

Example 5.7. **(ii) Absorbing States.** We extend the base case Example 5.2 ($A = 1$, $T = 20$, $N = 20$) with an absorbing failure state. Each transient state i transitions to the failure state with probability $r_i = \gamma \cdot (i/T) \cdot (1 - p_i)$, with $\gamma = 0.5$, so that roughly 20% of observations end at the failure sink. BDS jointly estimates $(p_1, \ldots, p_{T-1})$ and $(q_1, \ldots, q_{T-2})$ a total of $2T - 3 = 37$ parameters via automatic differentiation (Proposition C.1). We vary $K \in \{500, 1\,000, 2\,000\}$.

Results. Table 20 (Appendix K) shows accuracy, solve time, and convergence across all K. MAPE_p at $K = 1\,000$ (0.008 ± 0.001) matches the base case without absorbing states, confirming that the additional failure parameters do not degrade estimation of the staying probabilities. MAPE_q is larger, reflecting weaker statistical signal for the advance parameters: q_i values are small and failure observations provide most of the information. All runs converge, and accuracy improves monotonically with K: MAPE_q falls from 0.31 at $K = 500$ to 0.15 at $K = 2\,000$.

Example 5.8. **(iii) Belief-Distribution Data.** We consider the Base Case of Example 5.2 but now we use distributions for given states and actions, i.e., we use $\vec{x}_1$, $\vec{x}_2$, and $\vec{y}$ as data, which are distributed uniformly with $W = 1$ (i.e. $(1/3, 1/3, 1/3)$ for differences $\{-1, 0, 1\}$) around random 1-dimensional realizations as used in Example 5.2. Further, we let $K = T \cdot A \cdot 50$ to account for different numbers of variables, cf. $p_i(a)$.

Results. Appendix L summarizes the full parameter sensitivity for Example 5.8: accuracy vs. T (Table 21), N (Table 22, mirrors Table 7), K (Table 23, plateau near 0.5% from $K = 5{,}000$), and A (Table 24, accuracy stable near 0.8% across all A). Overall, compared to Example 5.2, the accuracies are slightly worse, which is due to the stochasticity in the data; the estimation of the different $\vec{p}(a)$ decouples across actions in (8).

6 Discussion and Conclusion

We studied MLE of monotone upper-bidiagonal Markov aging processes from sparse, heterogeneous n-step observations — a setting where the evaluated competing methods either time out, exhaust memory, or degrade substantially at large N (Table 1). The central contribution is Theorem 4.1: a parameterized diagonalization $\bar{P}(\vec{p}) = U(\vec{p})D(\vec{p})V(\vec{p})$ in which all three factors are explicit closed-form functions of the unknown probabilities $\vec{p}$. The key consequence (Theorem 4.3) is *horizon-independent* inference: likelihood evaluation and exact analytic gradients both cost $O(T^2 + TH)$ with no dependence on the maximum observation horizon N. BDS reaches $N = 100{,}000$ in under 7 s (Table 1). In further experiments, BDS achieves below 1% MAPE at $K = 1{,}000$ observations, and improves monotonically with both K and N. Two extensions preserve $O(T^2 + TH)$: absorbing failure states ($2T-3$ parameters) and basis-function reparametrization (below 0.4% MAPE up to $T = 100$ with $B = 3$ coefficients). Belief-distribution observations preserve horizon-independence but raise the per-observation cost to $O(T^2)$, yielding $O(AT^2 + KT^2)$ per gradient call.

Two real-world datasets show that the role of the structural prior depends on the domain. On NASA battery B0006, BDS matches a fully general $T \times T$ fitted matrix to within 0.02 per entry, and leads GSS at all sample sizes (6.13 vs. 16.89 nats at $K = 50$, 7.61 vs. 13.57 at full data, chronological split) — the bidiagonal prior is structurally correct for monotone degradation. On XJTU-SY bearings (15 run-to-failure trajectories across three load conditions), BDS again leads at $K = 50$ (3.25 vs. 4.60 nats), but GSS overtakes from $K = 200$ onward and reaches 1.11 vs. 2.41 nats at full data. The reversal reflects heterogeneous, non-monotone wear rates pooled across conditions and bearings: the bidiagonal prior is too rigid once the unconstrained model has sufficient data.

Limitations and Future Work The bidiagonal ($J = 1$) case is the exact result of this paper; the $J > 1$ upper-triangular eigenvalue structure provides a natural starting point for analogous closed-form expressions (Proposition C.2, Appendix C). The monotone single-step assumption absorbs 3.6% recovery transitions in the NASA data without material accuracy loss, but settings where recovery is systematic require a generalized model. Pooled heterogeneous sub-populations, as in the bearing experiment, motivate mixture or hierarchical BDS extensions — one component per operating condition. Bayesian estimation via conjugate Beta priors (Jafari et al., 2019) would add calibrated uncertainty over degradation rates. On the numerical side, the L'Hôpital-based stable evaluation (20) (Appendix M) resolves coincident-eigenvalue cases exactly; extending this to the large-hop tail via a hybrid fallback would complete the numerical coverage for any Δp and any h.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and minor implementation support, including assistance with code-related tasks during development. We have not used generative AI tools to generate the scientific contributions, theoretical results, proofs, experimental findings, or interpretations presented in this work. We have reviewed and verified all AI-assisted text and implementation changes, including checking the correctness of any AI-assisted code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We do not identify any ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

We provide (anonymous) source code folder containing all implementations, the experimental configurations, and scripts used to reproduce all reported results.

Experimental setups — solvers, hardware, data generation — are described in Section 5; full parameter sensitivity tables and real-world datasets are described in the Appendix.

All results were produced on a consumer laptop (Intel Core Ultra 5 225U, 32 GB RAM, single-threaded using Julia).

REFERENCES

Nils Collath, Benedikt Tepe, Stefan Englberger, Andreas Jossen, and Holger Hesse. Aging aware operation of lithium-ion battery energy storage systems: A review. *Journal of Energy Storage*, 55: 105634, 2022. ISSN 2352-152X. doi: https://doi.org/10.1016/j.est.2022.105634. URL https://www.sciencedirect.com/science/article/pii/S2352152X2201622X.

Matteo Corno and Gabriele Pozzato. Active adaptive battery aging management for electric vehicles. *IEEE Transactions on Vehicular Technology*, 69(1):258–269, 2020. doi: 10.1109/TVT.2019.2940033.

Bruce A Craig and Peter P Sendi. Estimation of the transition matrix of a discrete-time Markov chain. *Health economics*, 11(1):33–42, 2002.

Houde Dai, Guangcai Zhao, Mingqiang Lin, Ji Wu, and Gengfeng Zheng. A novel estimation method for the state of health of lithium-ion battery using prior knowledge-based neural network and markov chain. *IEEE Transactions on Industrial Electronics*, 66(10):7706–7716, 2019. doi: 10.1109/TIE.2018.2880703.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Frank Figge et al. Does the circular economy fuel the throwaway society? The role of opportunity costs for products that lose value over time. *Journal of Cleaner Production*, 368:133207, 2022.

Jia Guo, Yaqi Li, Kjeld Pedersen, and Daniel-Ioan Stroe. Lithium-ion battery operation, degradation, and aging mechanism in electric vehicles: An overview. *Energies*, 14(17), 2021. ISSN 1996-1073. doi: 10.3390/en14175220. URL https://www.mdpi.com/1996-1073/14/17/5220.

Md Alamgir Hossain, Hemanshu Roy Pota, Stefano Squartini, Forhad Zaman, and Josep M. Guerrero. Energy scheduling of community microgrid with battery cost using particle swarm optimisation. *Applied Energy*, 254:113723, 2019. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2019.113723. URL https://www.sciencedirect.com/science/article/pii/S0306261919314102.

Mehdi Jafari, Khalid Khan, and Lucia Gauchia. Deterministic models of li-ion battery aging: It is a matter of scale. *Journal of Energy Storage*, 20:67–77, 2018. ISSN 2352-152X. doi: https://doi.org/10.1016/j.est.2018.09.002. URL https://www.sciencedirect.com/science/article/pii/S2352152X18303098.

Mehdi Jafari, Laura E. Brown, and Lucia Gauchia. Hierarchical bayesian model for probabilistic analysis of electric vehicle battery degradation. *IEEE Transactions on Transportation Electrification*, 5(4):1254–1267, 2019. doi: 10.1109/TTE.2019.2956350.

Jun-Hyeong Kim, Eunji Kwak, Jinho Jeong, and Ki-Yong Oh. Physics-informed markov chain model to identify degradation pathways of lithium-ion cells. *IEEE Transactions on Transportation Electrification*, 10(2):3468–3481, 2024. doi: 10.1109/TTE.2023.3302433.

Rui Li, Liying Bao, Lai Chen, Cheng Zha, Jingyang Dong, Nan Qi, Rui Tang, Yun Lu, Meng Wang, Rong Huang, Kang Yan, Yuefeng Su, and Feng Wu. Accelerated aging of lithium-ion batteries: Bridging battery aging analysis and operational lifetime prediction. *Science Bulletin*, 68(23):3055–3079, 2023. ISSN 2095-9273. doi: https://doi.org/10.1016/j.scib.2023.10.029. URL https://www.sciencedirect.com/science/article/pii/S2095927323007296.

Xudong Li, Mengdi Wang, and Anru Zhang. Estimation of markov chain via rank-constrained likelihood. In *International Conference on Machine Learning*, pp. 3033–3042. PMLR, 2018.

Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Bhaskar Saha and Kai Goebel. Battery Data Set. NASA Ames Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA, 2007. URL https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/.

Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69(1):401–412, 2020. Dataset: XJTU-SY Bearing Datasets, https://ieee-dataport.org/open-access/xjtu-sy-bearing-datasets.

Timm Weitzel and Christoph H. Glock. Energy management for stationary electric energy storage systems: A systematic literature review. *European Journal of Operational Research*, 264(2):582–606, 2018. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2017.06.052. URL https://www.sciencedirect.com/science/article/pii/S0377221717305933.

Mingye Zhu, Quan Ouyang, Yong Wan, and Zhisheng Wang. Remaining useful life prediction of lithium-ion batteries: A hybrid approach of grey–markov chain model and improved gaussian process. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 11(1):143–153, 2023. doi: 10.1109/JESTPE.2021.3098378.

STRUCTURE OF THE APPENDIX

The appendix is organized as follows.

- **A. Proof of Theorem 4.1** — Parameterized diagonalization of the monotone bidiagonal family; algebraic derivation of $U(\vec{p})$, $D(\vec{p})$, $V(\vec{p})$.
- **B. Proof of Theorem 4.3** — $O(T^2+TH)$ likelihood and gradient cost; analytical gradient derivation.
- **C. More General Dynamics** — Absorbing failure states (Proposition C.1), wider jump widths $J>1$ (Proposition C.2), and downstream policy optimization.
- **D. Novelty vs. Matrix-Function Methods** — Why existing diagonalization techniques do not solve the parameterized MLE problem.
- **E. Usage-Replacement Policy Optimization** — Closed-form policy derivation from fitted $\hat{\vec{p}}$.
- **F. Evaluation Tables: Synthetic Data** — Full sensitivity tables for N, K, T, G, A.
- **G. Model Misspecification Analysis** — Recovery-transition contamination and wider jump-width stress tests.
- **H. Real-World Example: NASA Battery B0006** — Dataset, fitted transition matrices, and sample-efficiency sweep (BDS vs. GSS vs. EMP).
- **I. Real-World Example: XJTU-SY Bearing Degradation** — Dataset, fitted stay-probabilities, and sample-efficiency sweep across 15 bearings and three load conditions.
- **J. Evaluation Tables: Variable Reduction** — Basis-function reparametrization results (B=3, T up to 200).
- **K. Evaluation Tables: Absorbing States** — Joint estimation of stay- and failure-transition probabilities.
- **L. Evaluation Tables: Belief-Distribution Observations** — Results for the KL-objective formulation.
- **M. Numerical Stability: Applicability and Robustness** — Eigenvalue-gap analysis, $h_{\max}(\Delta p)$ characterization, log-space implementation, L'Hôpital-based stable evaluation (20) for coincident eigenvalues, and the repulsion-attraction geometry that governs optimizer behavior near the stability boundary.

A Proof of Theorem 4.1 (Section 4)

Throughout this proof we assume $0 < p_i < 1$ for $i = 1, \dots, T-1$ with $p_1, \dots, p_{T-1}$ pairwise distinct, and we adopt the convention $p_0 := 1$. Recall the auxiliary products (empty product $= 1$):

$$\lambda(i,j) := \prod_{k=i}^{j} (p_k - 1),$$
$$\kappa(i,j) := \prod_{k=i}^{j-1} (p_j - p_k),$$
$$\theta(i,j) := \prod_{k=i+1}^{j} (p_i - p_k).$$

The diagonal entries of D are $d_1 := 1$ and $d_r := p_{r-1}$ for $r = 2, \dots, T$.

A.1 Right eigenvectors and the columns of U

Eigenvalue $d_1 = 1$. Every row of $\bar{P}$ sums to one, so $\bar{P}\,\mathbf{1} = \mathbf{1}$. Hence the all-ones vector is a right eigenvector for eigenvalue 1, giving $U_{i,1} = 1$ for all i, which matches (5).

Eigenvalue $d_r = p_{r-1}$, $r \geq 2$. For $i < T$ the i-th row of $\bar{P}\,u^{(r)} = p_{r-1} u^{(r)}$ gives the recurrence

$$u_{i+1}^{(r)} = \frac{p_{r-1} - p_i}{1 - p_i}\, u_i^{(r)}. \tag{9}$$

We normalize by $u_{r-1}^{(r)} = 1$. Setting $i = r-1$ in (9) yields $u_r^{(r)} = 0$ (since $p_{r-1} - p_{r-1} = 0$); by induction every subsequent entry vanishes, so $u_i^{(r)} = 0$ for $i \geq r$. This gives $U_{i,j} = 0$ for $j > 1$ and $i \geq j$, and $U_{j-1,j} = 1$, consistent with (5).

Lemma A.1. *For $1 \leq i \leq r-2$,*

$$u_i^{(r)} = \frac{\lambda(i, r-2)}{\kappa(i, r-1)} (-1)^{i+r-1}.$$

Proof. We proceed by backward induction on i.

Base case $i = r-2$. From (9) with $u_{r-1}^{(r)} = 1$, $u_{r-2}^{(r)} = \frac{1-p_{r-2}}{p_{r-1}-p_{r-2}}$. The formula gives $\frac{\lambda(r-2,r-2)}{\kappa(r-2,r-1)}(-1)^{2r-3} = \frac{p_{r-2}-1}{p_{r-1}-p_{r-2}}(-1)^{2r-3} = \frac{1-p_{r-2}}{p_{r-1}-p_{r-2}}$. These agree.

Inductive step. Assume the formula holds for $i+1$. From (9), $u_i^{(r)} = \frac{1-p_i}{p_{r-1}-p_i} u_{i+1}^{(r)}$. Using $\lambda(i, r-2) = (p_i - 1)\lambda(i+1, r-2)$ and $\kappa(i, r-1) = (p_{r-1} - p_i)\kappa(i+1, r-1)$, a direct substitution gives the formula for i. □

Combining $U_{i,1} = 1$, $U_{r-1,r} = 1$, $U_{i,r} = 0$ for $i \geq r$, and Lemma A.1 recovers exactly (5). Hence

$$\bar{P}\,U = U\,D. \tag{10}$$

A.2 Left eigenvectors and the rows of V

Eigenvalue $d_1 = 1$. The last state T is absorbing ($\bar{P}_{T,T} = 1$), so $e_T^\top$ is a left eigenvector for eigenvalue 1. This gives $V_{1,j} = \mathbf{1}\{j = T\}$, matching the first case of (6).

Eigenvalue $d_r = p_{r-1}$, $r \geq 2$. For $j < T$ the j-th component of $v^{(r)\top} \bar{P} = p_{r-1} v^{(r)\top}$ gives the backward recurrence

$$v_{j-1}^{(r)} = \frac{p_{r-1} - p_j}{1 - p_{j-1}}\, v_j^{(r)}. \tag{11}$$

We normalize by $v_{r-1}^{(r)} = 1$. Setting $j = r-1$ in (11) gives $v_{r-2}^{(r)} = 0$; repeated application yields $v_j^{(r)} = 0$ for $j \leq r-2$, giving $V_{i,j} = 0$ for $i > j+1$ and $V_{i,i-1} = 1$, consistent with (6).

Lemma A.2. *For $r \le j < T$,*

$$v_j^{(r)} = \frac{\lambda(r-1, j-1)}{\theta(r-1, j)}(-1)^{r+j-1}.$$

Proof. We proceed by forward induction on j.

Base case $j = r$. Rearranging (11), $v_r^{(r)} = \frac{1-p_{r-1}}{p_{r-1}-p_r} v_{r-1}^{(r)} = \frac{1-p_{r-1}}{p_{r-1}-p_r}$. The formula gives $\frac{\lambda(r-1,r-1)}{\theta(r-1,r)}(-1)^{2r-1} = \frac{p_{r-1}-1}{p_{r-1}-p_r}(-1)^{2r-1} = \frac{1-p_{r-1}}{p_{r-1}-p_r}$. These agree.

Inductive step. Assume the formula holds for $j-1$ where $r < j < T$. Rearranging (11), $v_j^{(r)} = \frac{1-p_{j-1}}{p_{r-1}-p_j} v_{j-1}^{(r)}$. Using $\lambda(r-1, j-1) = \lambda(r-1, j-2)(p_{j-1}-1)$ and $\theta(r-1, j) = \theta(r-1, j-1)(p_{r-1}-p_j)$, a direct substitution gives the formula for j. □

Terminal entry $v_T^{(r)}$. The T-th component of $v^{(r)\top}\bar{P} = p_{r-1}v^{(r)\top}$ gives $(1-p_{T-1})v_{T-1}^{(r)} + v_T^{(r)} = p_{r-1}v_T^{(r)}$, hence $v_T^{(r)} = \frac{1-p_{T-1}}{p_{r-1}-1} v_{T-1}^{(r)}$. Substituting Lemma A.2 for $j = T-1$ and using $\lambda(r-1, T-2)\cdot(p_{T-1}-1) = (p_{r-1}-1)\cdot\lambda(r, T-1)$ one obtains $v_T^{(r)} = \frac{\lambda(r,T-1)}{\theta(r-1,T-1)}(-1)^{r+T-1}$, matching the $i < j = T$ case of (6). For $r = T$ the normalization gives $v_T^{(T)} = -1$, the $i = j = T$ case.

Combining all cases recovers exactly (6). Hence

$$V\bar{P} = DV. \tag{12}$$

A.3 U AND V ARE INVERSES

From (10) and (12), if $r \neq s$ then

$$d_r(VU)_{r,s} = (V\bar{P}U)_{r,s} = d_s(VU)_{r,s},$$

so $(d_r - d_s)(VU)_{r,s} = 0$. Since $d_1, \dots, d_T$ are pairwise distinct (eigenvalue 1 is distinct from all $p_{r-1} \in (0,1)$, and the p_i are pairwise distinct by assumption), we conclude $(VU)_{r,s} = 0$ for $r \neq s$.

For $r = s = 1$: the first row of V is $e_T^\top$ and the first column of U is $\mathbf{1}$, so $(VU)_{1,1} = 1$.

For $r = s \ge 2$: $U_{i,r} = 0$ for $i \ge r$ and $V_{r,j} = 0$ for $j \le r-2$. The only surviving term in the inner product $v^{(r)\top}u^{(r)}$ is $V_{r,r-1}U_{r-1,r} = 1 \cdot 1 = 1$.

Therefore $VU = I_T$, and since both matrices are square,

$$V = U^{-1}. \tag{13}$$

A.4 COMPLETION

From $V\bar{P} = DV$ and $U = V^{-1}$,

$$\bar{P} = U\,D\,V.$$

Since $D^n = \mathrm{diag}(1, p_1^n, \dots, p_{T-1}^n)$, for every positive integer n,

$$\bar{P}^n = U\,D^n\,V, \qquad \bar{P}_{i,j}^n = \sum_{t=1}^{T} U_{i,t}\, d_t^n\, V_{t,j}.$$

This proves Theorem 4.1. □

B Proof of Theorem 4.3

We analyze the cost of evaluating the likelihood for a single action a; the multi-action result follows by summing over actions. Let $H_a = \#\{(x, y, n) : (x, y, n, a) \text{ occurs in the data}\}$.

B.1 Precomputing $U(\vec{p})$

We construct U column by column using the right-eigenvector recurrence (9) derived in Appendix A.

Column 1 satisfies $U_{i,1} = 1$ for all i, requiring $O(T)$. For column $r = 2, \ldots, T$, set $U_{i,r} = 0$ for $i \geq r$ and $U_{r-1,r} = 1$, then apply (9) backwards:

$$U_{i,r} = \frac{1 - p_i}{p_{r-1} - p_i} U_{i+1,r}, \qquad i = r-2,\ r-3,\ \ldots,\ 1.$$

Within column r the eigenvalue p_{r-1} is fixed, so each step is a single multiply-divide. Column r requires $O(r)$ operations; summing over $r = 1, \ldots, T$ constructs U in $O(T^2)$ total.

B.2 Precomputing $V(\vec{p})$

Rather than using the product representation (6) column-by-column (which requires recomputing the θ product from scratch as the base p_{i-1} changes), we apply the left-eigenvector recurrence (11) established in Appendix A.

The first row is $V_{1,j} = \mathbf{1}\{j = T\}$. For $r = 2, \ldots, T$, initialize $V_{r,r-1} = 1$ and apply (11) forward:

$$V_{r,j} = \frac{1 - p_{j-1}}{p_{r-1} - p_j} V_{r,j-1}, \qquad j = r, \ldots, T-1.$$

The terminal entry $V_{r,T}$ follows from the $j = T$ component of the left-eigenvector equation (see Appendix A, §A.2):

$$V_{r,T} = \frac{1 - p_{T-1}}{p_{r-1} - 1} V_{r,T-1} \quad (r < T), \qquad V_{T,T} = -1.$$

Each row r requires $O(T - r + 1)$ operations, so all T rows are constructed in $O(T^2)$ total.

B.3 Evaluating the required n-step probabilities

Once U and V are available, Theorem 4.1 gives

$$\bar{P}^n_{x,y} = \sum_{t=1}^{T} U_{x,t}\, p^n_{t-1}\, V_{t,y}, \qquad p_0 := 1.$$

The dependence on n enters only through the scalar powers p^n_{t-1}; no recursion over $1, \ldots, n$ is required. Each of the H_a distinct triples costs $O(T)$, giving $O(TH_a)$ after preprocessing.

Total per action. The preprocessing costs $O(T^2)$ and the likelihood evaluation costs $O(TH_a)$, for a combined cost of $O(T^2 + TH_a)$. Summing over A actions and setting $H := \sum_{a=1}^{A} H_a$ yields $O(AT^2 + TH)$ total. For a fixed number of actions this is $O(T^2 + TH)$; in the regime $H = \Omega(T)$ the preprocessing is absorbed, giving $O(TH)$.

B.4 Gradient complexity

We show that the full gradient $\nabla_{\vec{p}} \log \ell$ costs $O(T^2 + TH)$. Naively, computing the derivative matrices U'_k, D'_k, V'_k for each of the $T-1$ parameters separately would cost $O(T^2)$ per parameter and $O(T^3)$ in total. Instead, we differentiate the scalar log-likelihood through the forward computational graph, tracing the reverse pass stage by stage.

Stage 1: prefix/suffix products for U and V. Each row of U (resp. V) is built by a running product as described in Sections B.1–B.2. Each running product involves $O(T)$ multiply-accumulate steps with $O(1)$ inputs per step. The reverse pass of a length-L running product accumulates gradients in $O(L)$ operations (standard chain-rule unrolling). Summed over all T rows, the reverse pass through U and V costs $O(T^2)$.

Stage 2: diagonal powers p_k^n. For each distinct observed step count n, the powers $p_1^n, \ldots, p_{T-1}^n$ are computed in $O(T)$ and their reverse pass (returning $\partial\ell/\partial p_k$ contributions of the form $n\, p_k^{n-1}\, \partial\ell/\partial(p_k^n)$) also costs $O(T)$ per observation, for $O(TH)$ total across all H observations.

Stage 3: likelihood contractions. Each of the H required entries $\bar{P}^n_{x,y} = \sum_t U_{x,t}\, p^n_{t-1}\, V_{t,y}$ is an $O(T)$ dot product; its reverse pass accumulates $O(T)$ additive contributions to $\partial\ell/\partial U_{x,\cdot}$, $\partial\ell/\partial(p^n_{t-1})$, and $\partial\ell/\partial V_{\cdot,y}$, all in $O(T)$. Total across all H entries: $O(TH)$.

Total. Combining the three stages, the reverse pass costs $O(T^2) + O(TH) + O(TH) = O(T^2 + TH)$, matching the forward cost. All $T-1$ gradient components are obtained in a single reverse pass; no per-parameter derivative matrix is formed.

Empirical gradient validation. We validate gradient correctness by comparing the automatic-differentiation gradient (ForwardDiff, the same source used by L-BFGS) against central finite differences with step $h = 10^{-5}$, over 100 random observation sets per cell. Using the ground-truth parameters of Example 5.2 (eigenvalue gap $\Delta p \approx 1/[2(1+T)\sqrt{N}]$, matching all paper experiments):

Table 2: Relative error $e_{\text{rel}} = \|\nabla_{\text{AD}}\ell - \nabla_{\text{FD}}\ell\| / \max(1, \|\nabla_{\text{FD}}\ell\|)$ between the ForwardDiff gradient and central finite differences ($h = 10^{-5}$), over 100 random observation sets per cell. Ground-truth $\vec{p}$ of Example 5.2.

T	K	MEAN e_{rel}	P99 e_{rel}	MAX e_{rel}
5	50	4.5×10^{-8}	1.3×10^{-7}	1.3×10^{-7}
10	50	6.1×10^{-8}	1.7×10^{-7}	4.6×10^{-7}
10	500	1.7×10^{-7}	3.2×10^{-7}	3.8×10^{-7}
20	50	1.4×10^{-7}	7.1×10^{-7}	2.6×10^{-6}
20	500	2.1×10^{-6}	4.2×10^{-5}	7.5×10^{-5}

Relative errors remain below 10^{-4} for all settings at $T \leq 20$, confirming that the AD gradient is accurate to near-machine precision. At larger T the error grows consistently with the hop-distance cancellation documented in Appendix M: both AD and FD degrade together for the same high-hop observations, so the growth reflects BDS evaluation error rather than any defect in gradient propagation.

B.5 Comparison with BSS

The BSS formulation represents $\bar{P}^1, \ldots, \bar{P}^N$ through the recursive constraint $\bar{P}^{k+1} = \bar{P}^k \cdot \bar{P}$. Because $\bar{P}$ is bidiagonal, each multiplication costs $O(T^2)$, and building the full sequence costs $O(T^2 N)$ per action. BSS also requires $O(AT^2N)$ auxiliary storage for the sequence of n-step matrices, whereas BDS requires only the parameterized factors U, D, V, whose storage is $O(T^2)$ independent of N.

The principal computational advantage of BDS is therefore the complete elimination of the horizon N from both runtime and storage. □

C More General Dynamics and Future Work

This section describes two structural extensions of the bidiagonal aging model and outlines further research directions. The first extension — direct transitions to an absorbing failure state — requires a modified eigenvector construction in which advance probabilities q_k replace the factors $(1 - p_k)$ in the λ-products of Theorem 4.1; the log-space formula and complexity carry over unchanged. The second — wider banded transition matrices — retains the diagonal eigenvalue structure and is identified as a natural next step requiring a higher-order recurrence.

C.1 Direct transitions to an absorbing failure state

Proposition C.1 (Absorbing failure-state extension)**.** *Suppose state T is absorbing. For $i \in \{1, \dots, T-2\}$:*

$$P_{i,i} = p_i, \quad P_{i,i+1} = q_i, \quad P_{i,T} = r_i = 1 - p_i - q_i,$$

with $p_i, q_i > 0$ and $r_i \geq 0$; for $i = T-1$: $P_{T-1,T-1} = p_{T-1}$, $P_{T-1,T} = 1 - p_{T-1}$. Let $Q = P_{1:T-1,\,1:T-1}$ be the transient submatrix and assume $p_1, \dots, p_{T-1}$ are pairwise distinct. Define $(T-1) \times (T-1)$ matrices $U(\vec{p}, \vec{q})$, $D(\vec{p})$, $V(\vec{p}, \vec{q})$ by

$$U_{i,t} = \begin{cases} 1 & i = t, \\ \displaystyle\prod_{k=i}^{t-1} \frac{q_k}{p_t - p_k} & i < t, \\ 0 & i > t, \end{cases} \qquad V_{t,j} = \begin{cases} 1 & j = t, \\ \displaystyle\prod_{k=t}^{j-1} \frac{q_k}{p_t - p_{k+1}} & j > t, \\ 0 & j < t, \end{cases}$$

and $D(\vec{p}) = \operatorname{diag}(p_1, \dots, p_{T-1})$. Then $Q^n = U(\vec{p}, \vec{q})\, D(\vec{p})^n\, V(\vec{p}, \vec{q})$, and for all $i \in \{1, \dots, T-1\}$ and $n \geq 1$,

$$P^n_{i,T} = 1 - \sum_{t=1}^{T-1} U_{i,t}\, w_t\, p_t^n, \qquad w_t := \sum_{j=1}^{T-1} V_{t,j}.$$

The $\lambda/\kappa/\theta$ log-space formula of Theorem 4.1 extends to $U(\vec{p}, \vec{q})$ and $V(\vec{p}, \vec{q})$ by replacing $(1 - p_k)$ with q_k in the λ-products; see Appendix M. The weights w_t can be precomputed in $O(T^2)$; thereafter each $P^n_{i,T}$ costs $O(T)$, independent of n.

Proof. Since Q is upper triangular with distinct diagonal entries, its eigenvalues are $p_1, \dots, p_{T-1}$. For eigenvalue p_t, the right-eigenvector equation $(Q - p_t I)u = 0$ in component i gives

$$p_i u_i + q_i u_{i+1} = p_t u_i \implies u_i = \frac{q_i}{p_t - p_i}\, u_{i+1} \quad (i < t),$$

and $u_{T-1} = 0$ for $t < T-1$ (from the last row). Setting $u_t = 1$ and iterating backwards yields $U_{i,t} = \prod_{k=i}^{t-1} q_k/(p_t - p_k)$ for $i \leq t$, with $U_{i,t} = 0$ for $i > t$. A symmetric argument on the left-eigenvector equation $v(Q - p_t I) = 0$ gives $V_{t,j} = \prod_{k=t}^{j-1} q_k/(p_t - p_{k+1})$ for $j \geq t$. Since $p_1, \dots, p_{T-1}$ are distinct, U is invertible with $V = U^{-1}$, so $Q = UDV$ and $Q^n = UD^nV$. The standard absorbing-chain identity gives

$$P^n = \begin{pmatrix} Q^n & \mathbf{1} - Q^n \mathbf{1} \\ 0 & 1 \end{pmatrix},$$

so $P^n_{i,T} = 1 - \sum_{j=1}^{T-1} Q^n_{i,j} = 1 - \sum_t U_{i,t} p_t^n w_t$. Because $q_k > 0$, the log-space sign-tracking of Appendix M applies verbatim with $\log q_k$ in place of $\log(1 - p_k)$. □

Practical implications. The extension adds $T-2$ new free parameters per action: the stay-or-advance probabilities $q_i = 1 - p_i - r_i$ for $i = 1, \dots, T-2$ (the terminal state needs no q_{T-1} since $r_{T-1} = 1 - p_{T-1}$), bringing the per-action parameter count from $T-1$ to $2T-3$. The likelihood evaluation complexity is unchanged at $O(T^2 + TH)$: the w_t weights are row sums of V, computed once in $O(T^2)$ alongside the precomputation of V, and each failure probability $P^n_{i,T}$ then evaluates in $O(T)$. Within a single action, p_i and r_i are coupled through the eigendecomposition: p_i drives both the transient-state probabilities $Q^n_{i,j}$ (via $U(\vec{p}, \vec{q})$, $D(\vec{p})^n$, $V(\vec{p}, \vec{q})$) and the weights w_t used in the failure formula, while r_i affects the likelihood only through the complementary probability $r_i = 1 - p_i - q_i$. Consequently, transient observations and failure observations jointly inform all p_i, and the two parameter sets are not independently identifiable from failure observations alone. Across actions, the structure is unchanged: the log-likelihood decomposes as a sum over a, so the per-action estimation problems remain independent.

Empirical results for this extension are presented in Appendix K.

C.2 Wider banded matrices

The bidiagonal model studied in this paper enforces that aging advances by at most one state per period. A natural extension is to allow larger monotone jumps: an upper-banded transition matrix with bandwidth J captures accelerated degradation events where the system can advance up to J states in one step, while retaining the no-recovery monotonicity. Such matrices are upper triangular with the same diagonal eigenvalues, so they remain diagonalizable under the pairwise-distinct assumption.

Proposition C.2 ($J = 2$ extension)**.** *Let P be an upper-triangular stochastic matrix with $P_{i,i} = p_i$, $P_{i,i+1} = q_i$, $P_{i,i+2} = r_i = 1 - p_i - q_i$ for $i = 1, \dots, T-2$, and $P_{T,T} = 1$, with $p_1, \dots, p_{T-1}$ pairwise distinct. For $i = T-1$ the jump-by-two destination $T+1$ lies outside the state space, so $r_{T-1} = 0$ and $P_{T-1,T-1} = p_{T-1}$, $P_{T-1,T} = 1 - p_{T-1}$. The right eigenvector $u^{(k)}$ of P for eigenvalue p_k satisfies, for $j < k$,*

$$u_j^{(k)} \;=\; \frac{q_j\, u_{j+1}^{(k)} + r_j\, u_{j+2}^{(k)}}{p_k - p_j}, \tag{14}$$

with boundary conditions $u_k^{(k)} = 1$ and $u_{k+1}^{(k)} = 0$. This recurrence runs in $O(T)$ per eigenvector, building U in $O(T^2)$ total. Row k of $V = U^{-1}$ is the left eigenvector for p_k, which satisfies the depth-2 forward recurrence

$$v_j^{(k)} = \frac{q_{j-1}\, v_{j-1}^{(k)} + r_{j-2}\, v_{j-2}^{(k)}}{p_k - p_j}, \qquad j = k+1, \dots, T, \tag{15}$$

initialized with $v_{k-1}^{(k)} = 0$ and $v_k^{(k)} = 1$, then normalized by dividing by $\langle v^{(k)}, u^{(k)} \rangle$ (an $O(k)$ inner product). Each row requires $O(T)$ operations; all T rows are built in $O(T^2)$ total without inverting U. Consequently, $\bar{P}^n_{x,y} = \sum_k U_{x,k}\, p_{k-1}^n\, V_{k,y}$ evaluates in $O(T)$ per entry, and BDS likelihood evaluation and gradient computation cost $O(T^2 + TH)$ — identical to the bidiagonal case.

The bidiagonal case $J = 1$ is the unique member of this family where the recurrence (14) telescopes into a product: with $r_j = 0$ it reduces to $u_j^{(k)} = q_j\, u_{j+1}^{(k)} / (p_k - p_j)$, which yields the closed-form $\lambda/\kappa/\theta$ expressions of Theorem 4.1. For general J, the recurrence has depth J but remains $O(T)$ per eigenvector and $O(T^2)$ for U and V; the $O(T^2 + TH)$ complexity guarantee thus extends to any fixed bandwidth. Deriving closed-form product expressions analogous to Theorem 4.1 for $J > 1$ remains open.

C.3 Further directions

State-dependent transition probabilities $p_i(a)$ with separate basis expansions per state (the per-row estimation problems decouple), mixture-of-chains models for products with multiple degradation modes, and Bayesian extensions of the maximum-likelihood objective are further natural avenues.

D Why BDS Is Novel: Comparison with Matrix-Function Methods

The relevant bottleneck is easy to understate. The MLE objective requires evaluating selected entries $\bar{P}(\vec{p})^n_{x,y}$ and their gradient $\nabla_{\vec{p}} \log \ell$ repeatedly as $\vec{p}$ changes at every L-BFGS iteration. The challenge is therefore *not* computing $\bar{P}^n$ once for a fixed matrix, but sustaining $O(N)$-independent per-gradient cost across hundreds of optimizer calls, each querying up to H distinct (x, y, n) triples. We examine each candidate approach against this criterion.

Matrix powers (sequential or repeated squaring). Sequential multiplication $\bar{P}, \bar{P}^2, \dots, \bar{P}^n$ costs $O(T^2 n)$ and builds the full matrix at every horizon up to n. Repeated squaring reduces this to $O(T^3 \log n)$ per unique n-value, but the full $T \times T$ matrix is formed and must be recomputed from scratch whenever $\vec{p}$ changes. Across H distinct triples, the per-gradient cost is $O(T^3 H \log N)$, growing with both T and N. Automatic differentiation through repeated squaring carries the same forward cost plus $O(T^2 \log n)$ memory per unique n for the intermediate matrices.

Schur decomposition and Parlett's recurrence. For a general upper-triangular matrix, Parlett's algorithm evaluates the matrix function $f(A)$ in $O(T^3)$ given the Schur factor. Since $\bar{P}$ is already upper triangular, the Schur factor is $\bar{P}$ itself, and for $f(z) = z^n$ Parlett's recurrence recovers a divided-difference formula mathematically equivalent to BDS. The cost per unique n is $O(T^3)$, it produces the full matrix (not sparse entries), and it must be rerun at every $\vec{p}$ update. BDS achieves the same result by exploiting the bidiagonal recurrence structure to construct U and V in $O(T^2)$ once per optimizer step and evaluate any entry in $O(T)$.

Divided differences (Newton form). Newton's divided-difference algorithm computes $f[p_1, \ldots, p_k]$ for $f(z) = z^n$ in $O(k^2)$ and is mathematically equivalent to the BDS sum for a single entry $\bar{P}^n_{x,y}$. The contribution of Theorem 4.1 is the closed-form $\lambda/\kappa/\theta$ product expressions for U and V: they (i) reduce amortized cost from $O(T^2)$ per entry to $O(T^2)$ total shared across all H observations, and (ii) expose a smooth parameterization in $\vec{p}$ that admits an exact reverse-mode gradient pass at $O(T^2+TH)$, without rebuilding divided-difference tables or forming per-parameter derivative matrices. A generic divided-difference implementation redoes $O(T^2)$ work per entry, incurring $O(T^2H)$ per gradient call with no amortization.

Generic eigendecomposition. For a general matrix, computing the eigendecomposition costs $O(T^3)$ via the QR algorithm and yields numerical eigenvectors without closed-form expressions. BDS exploits the bidiagonal structure: eigenvalues are the diagonal entries p_i (read off in $O(1)$ each), and the eigenvectors satisfy a two-term recurrence (Appendix A) that yields explicit $\lambda/\kappa/\theta$ product formulas and $O(T^2)$ construction of U and V. Rediagonalizing at each optimizer iterate thus costs $O(T^2)$ rather than $O(T^3)$, a saving of one factor of T per gradient call.

Krylov and matrix-exponential methods. Krylov-based methods (Arnoldi iteration) approximate $f(A)v$ for matrix-vector products, not individual (i, j) entries. They provide no closed-form gradient with respect to $\vec{p}$, and their approximation quality is not uniform across optimizer iterates. Matrix-exponential methods are designed for continuous-time generators; adapting them to parameterized discrete-time likelihoods reintroduces $O(N)$-dependent truncation.

Uniformization. Uniformization expresses the matrix exponential as a Poisson mixture of matrix powers: $e^{Qt} = e^{-\lambda t}\sum_{k=0}^{\infty}(\lambda t)^k P_u^k/k!$ where $P_u = I + Q/\lambda$. The series is truncated at order $O(\lambda N)$, so cost grows linearly with N. The method applies to continuous-time generators and produces full distribution vectors, not selected (x, y) entries.

Table 3 summarizes. The right-most column records whether the method admits an $O(T^2 + TH)$ gradient without rebuilding the full matrix or rerunning an $O(N)$-cost computation at each optimizer step.

Table 3: Comparison of approaches for evaluating n-step transition probabilities $\bar{P}^n_{x,y}(\vec{p})$ during likelihood optimization. "N-dependent" indicates whether the per-gradient cost grows with $\max_k n(k)$; "Sparse entry" indicates whether individual entries can be evaluated without forming the full n-step matrix; "Gradient cost" is per optimizer iteration across H observed triples, assuming the n-step sequence is precomputed once per gradient call where applicable.

Method	N-dep.?	Sparse entry?	Gradient cost	Gradient w/o full matrix?
Seq. matrix powers	Yes	No	$O(T^2N+H)$	No
Rep. squaring + AD	Yes	No	$O(T^3H \log N)$	No
Schur / Parlett	Yes	No	$O(T^3H)$	No
Divided differences (naive)	No	Yes	$O(T^2H)$	Yes, no amortization
Generic eigendecomposition	No	Yes	$O(T^3+TH)$	Yes, $O(T^3)$ diag.
Krylov / matrix exp.	Dep. on n	No	approx. only	No
Uniformization	Yes	No	$O(T^2N+H)$	No
BDS (ours)	**No**	**Yes**	$\mathbf{O(T^2+TH)}$	**Yes**

Empirical wall-clock comparison

To move beyond asymptotic claims, we measure the wall-clock time per gradient evaluation for three implementations, sweeping $T \in \{10, 20\}$ and $N \in \{20, 100, 500, 1{,}000\}$ at $K = 500$ observations:

BDS-AD closed-form $U(\vec{p})/V(\vec{p})$ differentiated via ForwardDiff (this paper); per-gradient cost $O(T^3 + T^2 H)$ (forward-mode AD over $T-1$ parameters, each at cost $O(T^2 + TH)$), N-independent. The theoretical $O(T^2 + TH)$ bound (Theorem 4.3, Appendix B) is achievable via a reverse-mode or explicit adjoint implementation.

RS-AD repeated squaring $\bar{P}(\vec{p})^n$ for each unique n, differentiated via ForwardDiff through the matrix-multiplication chain; cost $O(T^3 H \log N)$, grows with N.

EIG-FD Julia's built-in `eigen()` for n-step entry evaluation, gradient via central finite differences ($2(T-1)$ evaluations); represents a practitioner without access to the closed-form structure; cost $O(T^4)$ per gradient call.

Table 4: Median wall-clock time per gradient evaluation (ms), $K = 500$, ground-truth $\vec{p}$ of Example 5.2. RS/BDS and EIG/BDS give the speedup factor; both grow with N, confirming the horizon-dependence predicted by Table 3.

T	N	BDS-AD (ms)	RS-AD (ms)	EIG-FD (ms)	RS/BDS	EIG/BDS
10	20	0.09	2.0	2.3	22×	25×
10	100	0.10	4.6	5.1	46×	51×
10	500	0.14	20.3	22.8	142×	159×
10	1 000	0.17	27.5	28.0	161×	164×
20	20	0.38	8.1	10.8	21×	29×
20	100	0.48	54.1	42.4	112×	88×
20	500	0.63	227.7	154.2	361×	245×
20	1 000	0.74	310.1	180.9	420×	245×

BDS-AD is 22–420× faster than RS-AD and 25–245× faster than EIG-FD across the tested settings. Crucially, the BDS-AD time is nearly flat in N (0.09 ms at $N = 20$ to 0.17 ms at $N = 1{,}000$ for $T = 10$), while RS-AD and EIG-FD grow proportionally, confirming the $O(N)$-independence of the closed-form approach. The comparison is conservative for BDS: in a full MLE run the $O(T^2)$ precomputation of U and V is amortized over all H observations, whereas RS-AD and EIG-FD redo their respective decompositions from scratch at every optimizer step.

Note on EIG-FD and Table 3. Table 3 classifies generic eigendecomposition as N-independent ($O(T^3+TH)$): once $P = UDU^{-1}$, any entry $\bar{P}^n_{x,y} = \sum_k U_{x,k}\, d_k^n\, V_{k,y}$ costs $O(1)$ per eigenvalue via scalar exponentiation regardless of n. The EIG-FD benchmark diverges from this ideal: it evaluates gradients via $2(T-1)$ central finite differences, and each perturbed likelihood evaluation forms full $\bar{P}^n$ matrices for every distinct horizon via Julia's matrix-power routines. It is this matrix-power step — not the eigendecomposition itself — that reintroduces N-dependence in the benchmark timings.

E USAGE-REPLACEMENT POLICY OPTIMIZATION

This appendix gives (i) a compact MDP formulation showing how estimated $\hat{\vec{p}}$ feeds into operational decision-making, (ii) a proof of Proposition E.1, and (iii) numerical evidence that data-driven policies achieve 92–98% of optimal performance.

E.1 MDP FORMULATION

We formulate a discounted infinite-horizon MDP over states $s \in \{1, \dots, T\}$ and actions $a \in \{0, 1, \dots, A\}$, where $a = 0$ replaces the product (transition to $s' = 1$; net outlay $PN = PR - SV(s)$, yielding reward $SV(s) - PR = -PN$) and $a > 0$ continues use at intensity level a. The per-period reward is

$$R(a, s) := \begin{cases} SV(s) - PR, & a = 0, \\ u(a, s) - c(a, s), & a > 0, \end{cases} \tag{16}$$

with utility $u(a, s)$ and wear cost $c(a, s)$. One-step transition probabilities are $P(k{=}0, a, s) = p_s(a)$ and $P(k{=}1, a, s) = 1 - p_s(a)$ for $a > 0$, and $P(k{=}0, 0, s) = 1$. The state map is $\Gamma(k, a, s) = 1$ for $a = 0$, and $\Gamma(k, a, s) = s + k$ (capped at T) for $a > 0$. Value iteration, initialized at $V_0 \equiv 0$, converges to V^*:

$$V_t(s) = \max_a \Big[R(a, s) + \gamma \sum_{k=0,1} P(k, a, s)\, V_{t+1}(\Gamma(k, a, s)) \Big]. \tag{17}$$

The optimal policy $a^*(s)$ defines the critical threshold $s_c = \min\{s : a^*(s) = 0\}$ and the induced transition matrix $P^*_{s,s'} = p^*_s \cdot 1_{\{s'=s\}} + (1{-}p^*_s) \cdot 1_{\{s'=s+1\}}$ for $s < s_c$ and $P^*_{s,1} = 1$ for $s \geq s_c$, where $p^*_s := p_s(a^*(s))$.

Proposition E.1 (Stationary distribution under replacement policy)**.** *Let $a^*(s)$ be a control-limit (threshold) policy with $a^*(s) = 0$ for all $s \geq s_c$, where $s_c = \min\{s : a^*(s) = 0\}$, and write $p^*_s := p_s(a^*(s))$. The unique stationary distribution of the induced Markov chain is*

$$\pi^*_s = \begin{cases} \pi^*_{s_c} / (1 - p^*_s) & s < s_c, \\ \big(1 + \sum_{i=1}^{s_c - 1} (1 - p^*_i)^{-1}\big)^{-1} & s = s_c, \\ 0 & s > s_c, \end{cases} \tag{18}$$

with $\pi^*_{s_c}$ determined by the normalisation condition.

E.2 PROOF OF PROPOSITION E.1

Given $a^*(s)$ and s_c, the stationary equation $\pi^* = \pi^* P^*$ gives, recursively, $\pi^*_s = \pi^*_{s_c} / (1 - p^*_s)$ for $s < s_c$. We verify this directly. Let $\pi^*_{s_c} = \big(1 + \sum_{i=1}^{s_c - 1} 1/(1{-}p^*_i)\big)^{-1}$ and define $\pi^*_s = \pi^*_{s_c} / (1 - p^*_s)$ for $s < s_c$, $\pi^*_s = 0$ for $s > s_c$. Computing $(\pi_1, \dots, \pi_T) \cdot P^*$, the first entry is

$$\pi^*_1 p^*_1 + \pi^*_{s_c} = \tfrac{\pi^*_{s_c}}{1 - p^*_1} \cdot p^*_1 + \pi^*_{s_c} = \pi^*_{s_c} \Big(\tfrac{p^*_1}{1 - p^*_1} + 1 \Big) = \tfrac{\pi^*_{s_c}}{1 - p^*_1} = \pi^*_1.$$

For $i = 1, \dots, s_c{-}2$, the $(i{+}1)$-th entry satisfies

$$\pi^*_i (1 - p^*_i) + \pi^*_{i+1} p^*_{i+1} = \pi^*_{s_c} + \tfrac{\pi^*_{s_c} p^*_{i+1}}{1 - p^*_{i+1}} = \tfrac{\pi^*_{s_c}}{1 - p^*_{i+1}} = \pi^*_{i+1},$$

and the s_c-th entry gives $\pi^*_{s_c - 1}(1 - p^*_{s_c - 1}) = \pi^*_{s_c}$. States $s > s_c$ have $\pi^*_s = 0$ by reachability. Hence (18) satisfies $\pi^* = \pi^* P^*$, and normalization $\sum_s \pi^*_s = 1$ uniquely determines $\pi^*_{s_c}$. □

E.3 NUMERICAL ILLUSTRATION

For a representative instance ($T = 20$, $\gamma = 0.99$, $A = 10$, $PN = 150$, utility $u(a, s) = 5 + 0.2(a/A)^{0.3}$, cost $c(a, s) = 1_{\{s<T\}} \cdot 3(s/T)^{0.3} + 1_{\{s=T\}} \cdot 1\,000$), the optimal value function satisfies $V^*(1) = 366.26$ and $V^*(T) = 212.59$. Plugging estimated $\hat{\vec{p}}$ (from $K = A \cdot 1\,000$ observations) into (17) yields policies achieving approximately 92% of $V^*(1)$. As expected, with more data results improve; at $K = A \cdot 10\,000$ this rises to over 98%.

F Evaluation Tables - Observable Data (Section 5.3)

Tables 5–8 report parameter sensitivity of BDS in the observable-data setting (Example 5.2), sweeping T, K, N, G, and A in turn while holding the remaining parameters at their defaults.

Table 5: Impact of the number of states T; Example 5.2, $K = 1000$. Rows $T \leq 70$ use the direct product formula (5)–(6); rows $T \geq 100$ use the log-space implementation described in Appendix M. Julia reproduction, median time over 10 seeds.

T	MAPE	MAE	MED. TIME IN S
5	0.004± 0.003	0.004± 0.003	0.001
10	0.005± 0.001	0.005± 0.001	0.003
20	0.009± 0.001	0.008± 0.001	0.015
30	0.010± 0.002	0.010± 0.002	0.044
50	0.015± 0.002	0.014± 0.002	0.29
70	0.018± 0.002	0.017± 0.001	0.59
100	0.021± 0.002	0.020± 0.002	4.2
150	0.030± 0.004	0.028± 0.003	16.6
200	0.041± 0.011	0.038± 0.011	57.3

MAPE increases mildly from 0.4% at $T = 5$ to 2.1% at $T = 100$ and to 4.1% at $T = 200$, remaining well below the 5% practical threshold throughout. Solve time grows roughly as $O(T^2)$ for $T \leq 70$, consistent with Theorem 4.3; the log-space implementation for $T \geq 100$ (Appendix M) is slower, yet reaches $T = 200$ in under 60 seconds.

Table 6: Impact of the number of observations K; Example 5.2. Julia reproduction, median time over 10 seeds.

K	MAPE	MAE	MED. TIME IN S
50	0.077± 0.033	0.071± 0.031	0.007
100	0.031± 0.007	0.029± 0.007	0.007
200	0.019± 0.003	0.017± 0.002	0.012
500	0.012± 0.002	0.011± 0.002	0.011
1 000	0.009± 0.001	0.008± 0.001	0.020
2 000	0.006± 0.001	0.005± 0.001	0.036
5 000	0.004± 0.001	0.003± 0.001	0.100
10 000	0.003± 0.000	0.002± 0.000	0.179

MAPE falls below 1% already at $K = 1\,000$ and reaches 0.3% at $K = 10\,000$, with solve time remaining under one second throughout.

Table 7: Impact of the maximum number of steps N; Example 5.2. Julia reproduction, median time over 10 seeds.

N	MAPE	MAE	MED. TIME IN S
1	0.072± 0.011	0.050± 0.008	0.014
5	0.024± 0.006	0.021± 0.005	0.016
10	0.016± 0.003	0.014± 0.003	0.018
20	0.009± 0.001	0.008± 0.001	0.019
50	0.005± 0.001	0.005± 0.001	0.021
100	0.003± 0.001	0.003± 0.001	0.028
200	0.002± 0.000	0.002± 0.000	0.029
500	0.001± 0.000	0.001± 0.000	0.046
1 000	0.001± 0.000	0.001± 0.000	0.052

MAPE falls from 7.2% at $N = 1$ to 0.1% at $N = 1\,000$; solve time remains flat (0.014–0.052 s), confirming horizon-independent complexity (Corollary 4.4). Two notes on interpretation: (i) the

true parameters are calibrated as $p_i = 1 - O(N^{-0.5})$ so that mean hop counts remain bounded as N grows (keeping the experiment within the numerical stability region of Appendix M), which means the underlying process slows slightly across rows; (ii) MAPE measures relative error on the stay probabilities p_i, which approach 1 as N grows — relative error on the advance rates $q_i = 1 - p_i$ is approximately 18–37% and roughly constant across N, but the absolute error on q_i remains small (of order 10^{-3}), so the MAPE improvement reflects both increased observation informativeness and metric compression as $p_i \to 1$.

Table 8: Impact of the maximum observation horizon G (as fraction of N); Example 5.2. Each observation uses a step count n drawn uniformly from 1 to $\lfloor G \cdot N \rfloor$. Julia reproduction, median time over 10 seeds.

G	MAPE	MAE	MED. TIME IN S
0.05	0.029± 0.003	0.027± 0.003	0.024
0.1	0.023± 0.003	0.021± 0.003	0.026
0.2	0.017± 0.003	0.016± 0.003	0.022
0.3	0.014± 0.002	0.013± 0.002	0.018
0.4	0.012± 0.002	0.011± 0.002	0.017
0.5	0.012± 0.001	0.011± 0.001	0.016
0.6	0.011± 0.003	0.010± 0.003	0.018
0.7	0.010± 0.002	0.009± 0.001	0.018
0.8	0.009± 0.001	0.009± 0.001	0.017
0.9	0.009± 0.002	0.008± 0.002	0.020

MAPE decreases monotonically as G grows, from 2.9% at $G = 0.05$ (only very short-horizon observations) to 0.9% at $G = 0.9$ (near-full-horizon observations). Longer observation windows carry more information per trajectory about the multi-step transition structure, enabling tighter parameter identification. Solve time is essentially flat across all G values since BDS evaluates each (x, y, n) tuple independently with cost independent of n.

Table 9: Impact of the number of usage levels A; Example 5.2. Total observations $K = T \cdot A \cdot 50$ scale with A, so each action receives a fixed budget of $T \cdot 50 = 1\,000$ observations. Julia reproduction, median time over 10 seeds.

A	MAPE	MAE	MED. TIME IN S
1	0.009± 0.002	0.008± 0.002	0.02
2	0.009± 0.001	0.008± 0.001	0.04
3	0.009± 0.001	0.008± 0.001	0.06
4	0.009± 0.001	0.009± 0.001	0.08
5	0.009± 0.001	0.009± 0.001	0.08
7	0.009± 0.001	0.009± 0.001	0.11
10	0.009± 0.001	0.008± 0.001	0.20

MAPE remains near 0.9% across all values of A, reflecting that each action is estimated independently from its own fixed observation budget of $K/A = T \cdot 50 = 1\,000$. Solve time grows approximately linearly with A, consistent with the $O(AT^2 + TH)$ complexity of Theorem 4.3.

G Model Misspecification Analysis

The synthetic experiments in Section 5.3 use data generated under the exact bidiagonal assumption. This appendix tests two structural violations: recovery transitions (forbidden by the monotone model) and wider jump bandwidths ($J > 1$, the extension discussed in Appendix C).

G.1 Recovery Transitions

The bidiagonal model assumes monotone degradation: states cannot improve. Real systems occasionally exhibit apparent recovery (e.g. battery capacity regeneration, Section H). We test synthetic recovery by injecting recovery steps at rate r: with probability r each period, the state decreases by

one (from s to $\max(1, s-1)$); with probability $1-r$ the standard bidiagonal dynamics apply. BDS is fit on the contaminated data without modification; structurally inadmissible observations (recovery transitions $y < x$) have $\bar{P}^n_{x,y} = 0$ exactly under the bidiagonal model, so the log-likelihood applies a probability floor of $\varepsilon_{\text{machine}}$ to prevent $\log(0)$.

Table 10 shows MAPE of the estimated stay probabilities $\hat{\vec{p}}$ against the true bidiagonal $\vec{p}^*$ as the recovery rate varies.

Table 10: Effect of recovery-transition contamination on BDS estimation accuracy. $T = 20$, $K = 1{,}000$, $N = 20$, 10 seeds. MAPE of estimated $\hat{\vec{p}}$ vs. true $\vec{p}^*$.

Recovery rate r	MAPE	STD
0% (well-specified)	0.009	0.001
1%	0.010	0.001
3% (cf. NASA B0006)	0.011	0.002
5%	0.013	0.002
10%	0.017	0.002

Degradation is graceful: MAPE increases from 0.9% at $r = 0$ to 1.7% at $r = 10\%$, less than a factor of two over a tenfold increase in recovery rate. At the NASA-observed recovery rate of 3.6%, the model mismatch adds only ≈0.3 percentage points to the estimation error. The monotone bidiagonal approximation therefore remains practically adequate for moderate levels of recovery contamination.

G.2 Wider Jump Widths

The bidiagonal model restricts the state to advance by at most one step per period ($J = 1$). We test data generated with bandwidth $J \in \{1, 2, 3\}$: with probability proportional to q^j the state advances by $j \in \{0, 1, \ldots, J\}$ steps ($q = 0.15$). BDS ($J = 1$ only) is fit on the contaminated data; observations with $y - x > 1$ have $\bar{P}^n_{x,y} = 0$ under $J = 1$, so the $\varepsilon_{\text{machine}}$ floor (see recovery subsection above) is applied. Because the "true $\vec{p}$" concept does not carry over to $J > 1$, error is measured as the MAPE of the BDS predicted 1-step matrix $\hat{P}^1$ against the empirical 1-step distribution on a large held-out set ($K_{\text{test}} = 50{,}000$); entries with empirical probability below 10^{-4} are excluded from the MAPE to avoid division by near-zero values.

Table 11: Effect of jump-width misspecification on BDS predictive accuracy. $T = 20$, $K = 1{,}000$, $N = 20$, 10 seeds. MAPE of BDS predicted $\hat{P}^1$ vs. empirical 1-step distribution.

Bandwidth J	MAPE	STD
$J = 1$ (well-specified)	0.077	0.019
$J = 2$	0.409	0.029
$J = 3$	0.559	0.031

The $J = 1$ baseline MAPE of 7.7% reflects the difficulty of recovering 1-step structure from multi-step training data ($n \in \{1, \ldots, 20\}$) when the true 1-step matrix has nearly uniform entries. Under $J = 2$ and $J = 3$ the MAPE rises to 41% and 56%: BDS predicts zero probability for the $(i, i+2)$ and beyond entries that the true model places substantial mass on. This confirms that the $J = 1$ assumption is load-bearing: BDS should not be applied when the data-generating process allows multi-step jumps within a single period. The $J > 1$ extension outlined in Appendix C (Proposition C.2) directly addresses this case.

H Real-World Example: NASA Battery B0006

The NASA battery dataset (Saha & Goebel, 2007) provides real-world degradation trajectories under controlled charge-discharge cycling.

As an example, for battery B0006, we linearly mapped the remaining capacity with minimum (1.154 Ahr) and maximum values (2.035 Ahr) to our state representation $i = 1, ..., T$, with choosable T, where 1 is the best and T is the worst or terminal state.

To get an overview of the entire dataset, see file meta.csv in our codebase (column "capacity", the data we use is extracted specifically from B0006.mat in the NASA dataset).

From the provided time series for remaining capacity we collected $K = 3{,}150$ observations with n-step transitions between 1 and $N = 20$ (of which $K = 167$ are 1-step transitions used in Table 12 below).

Table 12: Number of transitions from state $i = x_1$ (rows) to $j = y$ (columns) for real-world data; $K = 167$ observations, $T = 10$, for 1-step intervals $n(k) = 1$, $k = 1, ..., K$ (battery B0006).

$i \backslash j$	1	2	3	4	5	6	7	8	9	10
1	6	1	0	0	0	0	0	0	0	0
2	0	13	3	0	0	0	0	0	0	0
3	0	2	12	2	0	0	0	0	0	0
4	0	0	1	13	1	0	0	0	0	0
5	0	0	0	0	8	1	0	0	0	0
6	0	0	0	0	0	15	3	0	0	0
7	0	0	0	0	0	2	24	2	0	0
8	0	0	0	0	0	0	1	26	1	0
9	0	0	0	0	0	0	0	0	20	1
10	0	0	0	0	0	0	0	0	0	9

The data reflects a state-dependent aging dynamic and also includes occasional state improvements (battery capacity regeneration phenomenon). Of the 167 1-step transitions: 146 stay at i to i, 15 degrade from i to $i+1$, 6 recover from i to $i-1$ (violating the monotone assumption), and 0 exhibit large jumps i to $i+2$ or beyond. The 6 recovery transitions (3.6% of 1-step data) constitute a systematic model mismatch; the remaining 96.4% are consistent with the monotone upper-bidiagonal structure.

H.1 Fitted Transition Matrices for Example 5.4

We compare the GSS estimate (full $T \times T$ transition matrix P) against the BDS estimate (bidiagonal $\bar{P}$) to assess how much information is lost by enforcing the monotone bidiagonal structure.

Table 13 shows the GSS results. The fitted GSS matrix is strongly concentrated on the bidiagonal structure, with only small probability mass on recovery transitions, suggesting that the monotone upper-bidiagonal approximation is reasonable for this dataset.

Table 13: Results when using a general aging matrix P: Fitted values for 1-step transition probabilities P with full $T^2 - T$ variables $p_{i,j}$, $i = 1, ..., T-1$, $j = 1, ..., T$ ($K = 3\,150$ observations, $T = 10$, $N = 20$ with $n(k)$ from 1 to 20, general model solved via GSS; Example 5.4).

$i \backslash j$	1	2	3	4	5	6	7	8	9	10
1	0.780	0.220	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.909	0.091	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.020	0.903	0.076	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.014	0.919	0.067	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.893	0.107	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.912	0.088	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.019	0.942	0.039	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.961	0.036	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.959	0.041
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Table 14 shows the BDS-fitted matrix. The maximum absolute difference between any entry of the two fitted matrices is 0.027, occurring in row 3 (BDS diagonal 0.930 vs. GSS 0.903) where GSS assigns small recovery probability that BDS absorbs into the diagonal. Despite the 3.6% recovery mismatch in the raw data, the two models agree closely, suggesting the monotone bidiagonal approximation is practically adequate for this dataset.

Table 14: Results when using a band matrix $\bar{P}$ (cf. BDS): Fitted values for bi-diagonal 1-step transition probabilities $\bar{P}$ with only $T-1$ variables $p_{i,i}$, $i = 1, ..., T-1$ ($K = 3\,150$ observations, $T = 10$, $N = 20$ with $n(k)$ from 1 to 20, band matrix model solved via BDS; Example 5.4).

$i\backslash j$	1	2	3	4	5	6	7	8	9	10
1	0.780	0.220	0	0	0	0	0	0	0	0
2	0	0.921	0.079	0	0	0	0	0	0	0
3	0	0	0.930	0.070	0	0	0	0	0	0
4	0	0	0	0.931	0.069	0	0	0	0	0
5	0	0	0	0	0.892	0.108	0	0	0	0
6	0	0	0	0	0	0.924	0.076	0	0	0
7	0	0	0	0	0	0	0.961	0.039	0	0
8	0	0	0	0	0	0	0	0.964	0.036	0
9	0	0	0	0	0	0	0	0	0.959	0.041
10	0	0	0	0	0	0	0	0	0	1.000

H.2 Sample Efficiency: NASA Battery B0006

Setup. We split the $K = 3{,}150$ multi-step observations chronologically: the first 80% form the pool ($K_{\text{pool}} = 2{,}520$) and the last 20% the held-out test set ($K_{\text{test}} = 630$), ensuring no interval overlap between train and test. We vary $K_{\text{train}} \in \{50, 100, 200, 500, 1\,000, 2\,000, 2\,520\}$ by drawing random subsets from the pool and evaluate mean negative log-likelihood (NLL) on the fixed test set for three models: BDS ($T-1 = 9$ parameters), GSS ($T(T-1) = 90$ parameters, softmax + L-BFGS), and EMP (empirical 1-step matrix raised to the n-th power). Each setting is repeated over 5 random subsets; mean $\pm$ std is reported. Zero-probability test transitions (e.g. recovery steps $y < x$) are floored at 10^{-19} in the NLL evaluation.

Results. At $K_{\text{train}} = 50$, BDS achieves NLL 6.13, clearly outperforming GSS (16.89) and EMP (7.59); the bidiagonal prior provides strong regularization when predicting late-degradation cycles from sparse early data. BDS remains the best model throughout: NLL increases slowly from 6.13 to 7.61 as training grows (the test set covers harder late-life cycles), while GSS and EMP remain far worse. EMP performs poorly throughout because only $\sim 5\%$ of observations are 1-step.

Table 15: Mean NLL (nats, lower is better) on the chronological held-out test set ($K_{\text{test}} = 630$, last 20% of cycles), NASA battery B0006. $\pm$ std over 5 random subsets of the training pool.

K_{TRAIN}	BDS	GSS	EMP
50	6.13 ± 0.18	16.89 ± 10.18	7.59 ± 5.17
100	6.53 ± 0.36	21.10 ± 10.69	9.76 ± 5.84
200	6.59 ± 0.14	11.53 ± 1.75	14.74 ± 1.04
500	7.00 ± 0.09	12.19 ± 2.58	16.88 ± 1.08
1000	7.31 ± 0.15	11.96 ± 1.51	17.64 ± 0.02
2000	7.56 ± 0.12	12.31 ± 0.74	17.65 ± 0.00
2520	7.61	13.57	17.64

I Real-World Example: XJTU-SY Bearing Degradation

Dataset. The XJTU-SY dataset Wang et al. (2020) contains 15 rolling-element bearings operated under three accelerated degradation conditions (35 Hz/12 kN, 37.5 Hz/11 kN, 40 Hz/10 kN) until failure, with vibration measured at 25.6 kHz once per minute. This yields between 42 and 2,538 measurement epochs per bearing depending on the run-to-failure lifetime under each load condition. The scalar health indicator is the root-mean-square (RMS) of the horizontal vibration channel:

$$z_t = \sqrt{\frac{1}{M}\sum_{m=1}^{M} x_{t,m}^2}, \qquad M = 1{,}536{,}000 \text{ samples/epoch.}$$

The RMS increases monotonically as bearing wear accumulates and terminates sharply at a failure threshold approximately $5\times$ the initial value. We discretize z_t into $T = 10$ ordered states (state 1 = healthy, state 10 = failed) via equal-width binning on $[z_1,\, 5z_1]$.

Observation construction. For each bearing trajectory we sample $O = 300$ observation pairs (x, y, n) by drawing $t_1 \sim U(1, L{-}1)$ and $n \sim U(1, \min(N_{\max}, L{-}t_1))$ with $N_{\max} = 30$, yielding $t_2 = t_1 + n$. This mimics a sparse condition-monitoring schedule with irregular, long intervals — exactly the long-horizon sparse regime of the paper. Across all 15 bearings this produces $K = 4{,}500$ observations.

Condition-stratified analysis. To verify that the GSS advantage is not a pooling artifact, we fit BDS and GSS *separately* on each operating condition (5 bearings, $K{=}1{,}500$ observations each, 80/20 split). Table 16 shows that GSS leads in every condition, ruling out heterogeneous-population confounding as the explanation. The bidiagonal prior is genuinely too restrictive for this dataset: wear dynamics within a single condition already require transition patterns that exceed the one-step advancement constraint.

Table 16: Condition-stratified NLL (nats, lower is better) on held-out test set. Each condition: 5 bearings, $K{=}1{,}500$ observations, 80/20 split.

Condition	BDS	GSS
35 Hz / 12 kN	2.830	2.113
37.5 Hz / 11 kN	1.890	0.829
40 Hz / 10 kN	1.729	0.733

Fitted BDS parameters. Table 17 shows the BDS stay-probabilities fitted on the full dataset. The irregular pattern ($p_4{=}0.325$, $p_7{=}0.214$ alongside high values elsewhere) reflects heterogeneous wear rates across the three load conditions and 15 individual bearings pooled into a single model.

Table 17: BDS fitted stay-probabilities p_i on all $K = 4{,}500$ bearing observations ($T = 10$, $N_{\max} = 30$).

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
0.982	0.921	0.822	0.325	0.761	0.755	0.214	0.756	0.376

I.1 Sample Efficiency: XJTU-SY Bearing Dataset

Setup. Random 80/20 split into pool and fixed test set; same sweep and repetition protocol as Appendix H.2; zero-probability test transitions floored at 10^{-19}.

Results. At $K{=}50$, BDS leads (3.25 vs. 4.60 nats), confirming that its structural prior compensates for scarce data. From $K{=}200$ onward, GSS overtakes BDS and reaches 1.11 nats at full data versus 2.41 for BDS — the reverse of the battery pattern. The results illustrate the bias–variance trade-off

induced by the structural prior: BDS provides strong sample efficiency when observations are scarce, whereas the unrestricted GSS model benefits from additional data when the underlying population violates the monotone-homogeneous assumption — as is the case here, with heterogeneous, non-monotone wear rates pooled across 15 bearings and three load conditions. EMP performs poorly throughout (5.6–6.9 nats).

Table 18: Mean NLL (nats, lower is better) on the held-out test set, XJTU-SY bearing dataset (15 bearings, $T = 10$, $N_{\max} = 30$, $K = 4{,}500$ total). $\pm$ std over 5 random subsets of the training pool.

K_{TRAIN}	BDS	GSS	EMP
50	3.25 ± 0.59	4.60 ± 1.57	4.42 ± 1.96
100	2.83 ± 0.36	2.82 ± 0.92	6.57 ± 0.74
200	2.71 ± 0.26	1.80 ± 0.30	6.33 ± 0.68
500	2.46 ± 0.07	1.52 ± 0.28	6.93 ± 1.10
1000	2.50 ± 0.11	1.20 ± 0.05	5.65 ± 2.59
2000	2.48 ± 0.04	1.12 ± 0.02	6.87 ± 0.48
3600	2.41	1.11	6.31

J Evaluation Tables - Variable Reduction (Section 4.3)

Table 19: Impact of the number of states T using $B = 3$ basis functions; Example 5.6. Julia reproduction, median time over 10 seeds.

T	MAPE	MAE	MED. TIME IN S
5	0.003 ± 0.002	0.003 ± 0.002	0.002
10	0.003 ± 0.002	0.002 ± 0.002	0.010
20	0.003 ± 0.001	0.003 ± 0.001	0.019
30	0.003 ± 0.001	0.003 ± 0.001	0.030
50	0.002 ± 0.001	0.002 ± 0.001	0.112
70	0.004 ± 0.001	0.003 ± 0.001	0.256
100	0.003 ± 0.001	0.003 ± 0.001	1.061

Two effects are visible. First, the $B = 3$ basis parametrization reduces the variable count from $T-1$ to a constant; median solve time grows from 0.002 s to 1.1 s across $T = 5, \ldots, 100$ (median reported due to outlier seeds at large T), compared to 0.001 s to 4.2 s for unstructured BDS (Table 5). At large T ($T \geq 50$), the $B = 3$ solver is up to $4\times$ faster than unstructured (1.061 s vs. 4.2 s at $T = 100$); the gradient computation is $O(T^2 + TH + BT)$ per iteration, and a small number of seeds require substantially more iterations to converge the 3-parameter surface. Second, MAPE remains at or below 0.4% across the entire range $T = 5, \ldots, 100$ — notably, the $T = 100$ result (0.3%) improves over the unstructured counterpart because the basis acts as implicit regularization that prevents overfitting when the per-state sample count is low. Compared to unstructured BDS at the same settings, accuracy is comparable or slightly better at all T, confirming that three basis coefficients are sufficient to capture the smooth degradation profile.

K Evaluation Tables - Absorbing States (Section 4.3)

Example K.1 (Absorbing failure-state estimation). We extend Example 5.2 ($A = 1$, $T = 20$, $N = 20$, $K = 1\,000$) with an absorbing failure state T. The true staying probabilities p_i follow the same formula as in Example 5.2. Failure probabilities increase proportionally to state index and staying probability:

$$r_i = \gamma \cdot \frac{i}{T} \cdot (1 - p_i), \qquad q_i = (1 - p_i) - r_i, \qquad \gamma = 0.5,$$

so that $\rho_i := q_i/(1-p_i) = 1-\gamma i/T \in (0, 1)$ is strictly interior for all transient states $i = 1, \ldots, T-2$, and state $T-1$ transitions only to the sink (no q_{T-1} entry in Q). About 20% of simulated observations end at the failure sink. We jointly estimate $(p_1, \ldots, p_{T-1})$ and $(q_1, \ldots, q_{T-2})$ — a

total of $2T - 3 = 37$ parameters — by maximizing the BDS log-likelihood of Proposition C.1 via L-BFGS with automatic differentiation. We sweep $K \in \{500, 1\,000, 2\,000\}$ with 3 repeated runs each.

Results. Table 20 shows accuracy, solve time, and convergence rate. MAPE_p at $K = 1\,000$ (0.008 ± 0.001) matches Example 5.2 at the same K (Table 6: MAPE $= 0.009$). MAPE_q is larger, reflecting weaker statistical signal for advance parameters: the q_i values are small (0.010–0.056) and failure observations provide the primary information at each state. All runs converge across all three K settings, and accuracy improves monotonically: MAPE_q falls from 0.31 at $K = 500$ to 0.15 at $K = 2\,000$.

Table 20: Absorbing failure-state estimation; Example K.1, $T = 20$, $N = 20$, $\gamma = 0.5$, 3 runs per K; median time over 3 seeds.

K	MAPE_p	MAPE_q	MED. TIME IN S
500	0.014 ± 0.001	0.309 ± 0.043	0.3
1 000	0.008 ± 0.001	0.198 ± 0.021	0.3
2 000	0.006 ± 0.001	0.145 ± 0.027	0.4

L Evaluation Tables - Belief Distribution Data (Section 5.5)

Tables 21–24 report parameter sensitivity of BDS in the belief distribution setting (Example 5.8), sweeping T, N, K, and A in turn while holding the remaining parameters at their defaults ($T = 10$, $N = 10$, $K = 1{,}000$, $A = 1$).

Table 21: Impact of the number of states T; Example 5.8. Julia reproduction, median time over 5 seeds. Default LBFGS gradient tolerance `g_tol`$= 10^{-6}$ (relaxed $= 10^{-3}$).

T	MAPE	MAE	MED. TIME IN S
5	0.022± 0.005	0.020± 0.005	0.00
10	0.013± 0.002	0.012± 0.002	0.02
20	0.008± 0.001	0.008± 0.001	0.21
30	0.007± 0.001	0.007± 0.001	1.11
50	0.006± 0.000	0.006± 0.000	1.33
70	0.006± 0.000	0.006± 0.000	2.95

Table 21 shows MAPE remaining below 2.2% for $T \leq 30$ (decreasing from 2.2% at $T = 5$ to 0.7% at $T = 30$) and stays at 0.6% through $T = 70$, reflecting the added difficulty of parameter identification when states are not directly observed. Solve time grows more steeply than in the observable case since the KL-objective evaluates a $T \times T$ matrix product per observation, reaching roughly $O(KT^2)$ per gradient call; using a relaxed gradient tolerance of 10^{-3} keeps solve times under 3 s for $T \geq 50$.

Table 22 shows accuracy improving monotonically with N: MAPE falls from 4.9% at $N = 1$ to below 0.1% at $N = 10{,}000$. Solve time is near-constant across N (range 0.23–1.32 s for $N = 1, \ldots, 100{,}000$), confirming that BDS evaluation cost is independent of N, as guaranteed by Theorem 4.3. Larger N provides richer multi-step information that tightens parameter identification in the hidden-state setting.

Table 23 shows accuracy improving with K up to around $K = 5{,}000$ and then plateaus near 0.5% MAPE regardless of further data. This floor reflects the irreducible uncertainty of the hidden-state setting: even with unlimited observations, the belief distribution cannot be pinpointed with arbitrary precision from n-step transitions alone. Solve time grows approximately linearly with K, consistent with the $O(KT^2)$ gradient evaluation cost; at $K = 100{,}000$ the median is 33 s with low variance.

Table 24 shows MAPE remaining stable near 0.8% across all values of A: the summation over a in (8) is outside the logarithm, so each $\vec{p}(a)$ appears only in its own terms and the A estimation problems decouple completely; each is solved from a fixed per-action budget of $K/A = T \cdot 50 = 1\,000$

Table 22: Impact of the maximum number of steps N; Example 5.8. Julia reproduction, median time over 10 seeds.

N	MAPE	MAE	MED. TIME IN S
1	0.049± 0.009	0.033± 0.006	0.25
5	0.021± 0.005	0.018± 0.005	0.23
10	0.013± 0.002	0.012± 0.002	0.32
20	0.008± 0.001	0.008± 0.001	0.32
50	0.005± 0.001	0.005± 0.001	0.36
100	0.004± 0.000	0.004± 0.000	0.66
200	0.003± 0.000	0.003± 0.000	0.48
500	0.002± 0.000	0.002± 0.000	0.38
1 000	0.001± 0.000	0.001± 0.000	0.46
2 000	0.001± 0.000	0.001± 0.000	1.28
5 000	0.001± 0.002	0.001± 0.002	1.00
10 000	0.000± 0.000	0.000± 0.000	1.03
20 000	0.000± 0.000	0.000± 0.000	1.32
50 000	0.000± 0.000	0.000± 0.000	1.06
100 000	0.011± 0.023	0.010± 0.022	1.23

Table 23: Impact of the number of observations K; Example 5.8. Julia reproduction, median time over 10 seeds.

K	MAPE	MAE	MED. TIME IN S
50	0.025± 0.002	0.023± 0.002	0.05
100	0.017± 0.003	0.016± 0.003	0.10
200	0.013± 0.002	0.012± 0.001	0.19
500	0.010± 0.001	0.010± 0.001	0.23
1 000	0.008± 0.001	0.008± 0.001	0.39
2 000	0.007± 0.001	0.006± 0.000	0.57
5 000	0.006± 0.001	0.006± 0.001	1.45
10 000	0.006± 0.001	0.005± 0.001	3.09
20 000	0.005± 0.000	0.005± 0.000	6.54
50 000	0.005± 0.000	0.005± 0.000	14.58
100 000	0.005± 0.000	0.005± 0.000	33.21

Table 24: Impact of the number of usage levels A; Example 5.8. Total observations $K = T{\cdot}A{\cdot}50$ scale with A; each action is optimized independently from its own $K/A = T{\cdot}50 = 1\,000$ observations. Julia reproduction, median time over 10 seeds.

A	MAPE	MAE	MED. TIME IN S
1	0.008± 0.001	0.007± 0.001	0.30
2	0.008± 0.001	0.008± 0.001	0.60
3	0.008± 0.001	0.007± 0.001	0.90
4	0.008± 0.000	0.007± 0.000	1.20
5	0.008± 0.000	0.007± 0.000	1.50
7	0.008± 0.000	0.007± 0.000	2.10
10	0.008± 0.001	0.007± 0.001	3.00

observations. Solve time grows roughly linearly with A, consistent with A independent optimization problems of equal cost.

M Numerical Stability: Applicability and Robustness

This section builds the full picture in four steps, distinguishing between *local accuracy* (correctness of individual $\bar{P}^n_{x,y}$ evaluations) and *estimation-level accuracy* (quality of the final parameter estimates $\hat{\hat{p}}$): (1) [local] map the safe evaluation regime as a function of two observable quantities; (2) [local] characterise the boundary via a heuristic digit-loss argument and confirm it empirically; (3) [local] describe the log-space implementation and the L'Hôpital-based stable evaluation (20) that together cover the full parameter space; (4) [estimation] show that the repulsion-attraction geometry keeps the optimizer productive throughout, and that even exactly coincident true parameters yield accurate estimates. Steps (1)–(2) correspond to paragraph (i) below; step (3) to paragraph (ii); step (4) to paragraphs (iii)–(vi).

(i) Local evaluation accuracy: heuristic and empirical reference bounds for safe hop distances. The BDS spectral sum $\bar{P}^n_{x,y} = \sum_k U_{x,k}\, p^n_{k-1}\, V_{k,y}$ is mathematically exact, but in floating-point arithmetic individual terms can far exceed the result $\bar{P}^n_{x,y} \in [0,1]$, so catastrophic cancellation in the signed sum is the failure mode. Local evaluation accuracy is governed by two observable quantities at the current optimization iterate: the *eigenvalue gap* $\Delta p = \min_{i\neq j} |p_i - p_j|$ (not the unknown true gap) and the *hop distance* $h = y - x$ (the state distance, not the horizon n). BDS computes $\bar{P}^n_{x,y}$ with relative error below a chosen target ϵ whenever $h \le h_{\max}(\Delta p, \epsilon)$ — a conservative sufficient condition, not a hard failure boundary; accuracy degrades smoothly beyond it. This is a statement about individual likelihood evaluations, not the final parameter estimates — the connection to estimation accuracy is made in paragraph (iii). Note that computational horizon-independence ($O(T^2{+}TH)$ cost independent of N) is a separate, unconditional property. Table 25 maps $h_{\max}(\Delta p)$ empirically. We stress-test using the ground truth of Example 5.2, $p^*_i = 1 - \frac{1}{2}\frac{1+i}{(A+T)\sqrt{N}}$, with uniformly spaced eigenvalues and gap $\Delta p = \frac{1}{2(A+T)\sqrt{N}}$. For each hop distance $h = y - x$ we evaluate $\bar{P}^n_{x,y}$ via BDS and compare against direct matrix exponentiation for all pre-states x and intervals $n \in [N/2, N]$; we report the maximum relative error over all entries with $|(\bar{P}^n)_{x,y}| \ge 10^{-12}$.

Table 25: BDS relative error by hop distance $h = y{-}x$ and eigenvalue gap Δp. Each cell is the maximum relative error over all admissible (x, n) pairs. Mean hops per observation: ≈0.6 (T=20, N=20), ≈2 (T=20, N=200), ≈4 (T=50 and T=100 with N=1,000). Shaded cells exceed 10^{-4}; bold boundary marks $h_{\max}(\Delta p)$.

	T=20, N=20 $\Delta p \approx 0.0053$	T=20, N=200 $\Delta p \approx 0.0017$	T=50, N=1000 $\Delta p \approx 0.00031$	T=100, N=1000 $\Delta p \approx 0.00016$
$h = 0$	8×10^{-16}	7×10^{-15}	4×10^{-14}	4×10^{-14}
$h = 2$	3×10^{-13}	3×10^{-14}	3×10^{-14}	1×10^{-13}
$h = 4$	3×10^{-10}	3×10^{-12}	6×10^{-12}	1×10^{-10}
$h = 6$	2×10^{-6}	5×10^{-10}	1×10^{-9}	6×10^{-8}
$\mathbf{h = 8}$	$\mathbf{3 \times 10^{-2}}$	7×10^{-8}	1×10^{-7}	5×10^{-5}
$h = 10$	$> 10^2$	$\mathbf{1 \times 10^{-5}}$	2×10^{-5}	3×10^{-2}
$h = 12$	$> 10^2$	2×10^{-3}	$\mathbf{4 \times 10^{-3}}$	∼20

The table shows that accuracy degrades smoothly with h — there is no hard cliff. The reference level $h_{\max}$ depends on the chosen accuracy target ϵ. To see why: the κ and θ products in U and V introduce $h-1$ denominator factors of order Δp, so individual BDS terms scale as $(\Delta p)^{-(h-1)}$ while their signed sum lies in $[0,1]$ — cancellation therefore costs roughly $-\log_{10}(\Delta p)$ digits per unit increase in h. Starting from $D \approx 15$ double-precision digits and requiring $-\log_{10}\epsilon$ digits of accuracy to remain gives the heuristic bound

$$h_{\max}(\Delta p,\, \epsilon) \;\approx\; \left\lfloor \frac{D + \log_{10}\epsilon}{-\log_{10}\Delta p} \right\rfloor, \tag{19}$$

where $D \approx 15$ is the number of significant digits in double precision. For $\epsilon = 10^{-4}$ the numerator is $15-4=11$; for $\epsilon = 10^{-2}$ it is $15-2=13$. The table below shows the heuristic bound alongside empirical values read from Table 25 at the four tested gaps; the bound is conservative by design — the empirical safe region is consistently larger because the formula assumes a worst-case data regime whereas each column in Table 25 also scales T and N:

	$\epsilon = 10^{-4}$		$\epsilon = 10^{-2}$	
Δp	Bound (19)	Empirical†	Bound (19)	Empirical†
0.0053	4	≈7	5	≈7
0.0017	3	≈11	4	>12
0.00031	3	≈11	3	>12
0.00016	2	≈9	3	≈9

† Empirical values from Table 25 (varying T, N per column).

For well-separated parameters ($\Delta p \geq 0.05$) the bound covers $h \geq 8$, comfortably exceeding the mean hop count of 0.6–4 in all main experiments. The bound contracts only when parameters approach coincidence, which is the regime addressed by the stable evaluation and the repulsion-attraction dynamics of Remark 4.2.

From local to global accuracy. The $h_{\max}$ boundary characterises the accuracy of individual $\bar{P}^n_{x,y}$ evaluations — it says nothing directly about whether the final parameter estimates $\hat{\vec{p}}$ are accurate. The connection runs through the log-likelihood: each observation (x, y, n) contributes $\log \bar{P}^n_{x,y}$ to the gradient, so observations with $h > h_{\max}$ introduce noise into individual gradient terms. Whether this matters depends on how common such observations are. For all settings evaluated in this paper, the *mean* hop count per observation is 0.6–4 ("long horizon" means large N, not large h: with $\bar{p} \approx 0.992$ and a 0.8% per-period transition probability, even $N = 1{,}000$ periods produce ≈4 expected hops per trajectory).

Two pieces of evidence bound the impact of large-hop observations on global MAPE. First, in the main experimental settings the fraction of observations exceeding $h_{\max}$ is small (sub-1% at $T = 100$, $N = 1{,}000$), so their noisy gradient contributions are diluted by the well-conditioned majority.

Second, and more telling: Table 8 shows MAPE decreasing monotonically as G grows from 0.05 to 0.9 — even though larger G means observations are drawn from longer intervals, producing larger expected hop distances and therefore more observations with $h > h_{\max}$. This means large-hop observations are informative rather than harmful: their likelihood signal, even if computed with reduced local accuracy, still points the optimizer toward the MLE. The local accuracy boundary $h_{\max}$ is therefore a characterization of floating-point behaviour, not a constraint on the method's applicability. Paragraph (iii) below confirms this at the level of individual optimization runs.

(ii) Extending local accuracy: log-space implementation and L'Hôpital-based stable evaluation. Three routes address the tail, in increasing order of completeness.

Log-space evaluation (resolves underflow, not cancellation). At the MLE the sign of each BDS term is deterministic, $\operatorname{sign}(U_{x,k}\, p^n_{k-1}\, V_{k,y}) = (-1)^{x+k-1}$, because the λ-product sign cancels exactly with the $(-1)^{i+j-1}$ correction in (5)–(6). The structural zeros and boundary values of (5)–(6) ($U_{i,1} = 1$, $U_{i,i+1} = 1$, $U_{i,j} = 0$ for $j > 1 \wedge i \geq j$, and the analogous cases for V) carry over unchanged; the expressions below replace only the interior "else" branch. During optimization, signs of κ and θ products are tracked entry-by-entry while log-magnitudes are summed:

$$U_{x,t} = \operatorname{sign}\big(\kappa(x,t-1)\big) \cdot \exp\Big(\log|\lambda(x,t-2)| - \log|\kappa(x,t-1)|\Big),$$

$$V_{t,j} = \begin{cases} +\operatorname{sign}\big(\theta(t-1,j)\big) \cdot \exp\Big(\log|\lambda(t-1,j-1)| - \log|\theta(t-1,j)|\Big) & j < T, \\ -\operatorname{sign}\big(\theta(t-1,T-1)\big) \cdot \exp\Big(\log|\lambda(t,T-1)| - \log|\theta(t-1,T-1)|\Big) & j = T. \end{cases}$$

This is a drop-in replacement for (5)–(6) and is beneficial at large T: without it, naïve products can underflow to zero beyond $T \approx 100$. It does not reduce cancellation in the final signed sum.

L'Hôpital-based stable evaluation (complete, implemented). The κ and θ denominators in U and V develop $0/0$ forms as $p_i \to p_j$. Because the BDS spectral representation is an explicit analytical formula, L'Hôpital's rule applies directly to each $0/0$ form: whenever any subgroup of eigenvalues coincides at p, the corresponding limit evaluates exactly to $\binom{n}{k}p^{n-k}$ where k is the subgroup size minus one. This covers all coincidence patterns via an $O(h^2)$ recurrence that substitutes this limit for any near-zero denominator. In the fully-coincident case $p_x = \cdots = p_y = p$ the result simplifies to the closed form

$$\bar{P}^n_{x,y}\big|_{p_x=\cdots=p_y=p} \;=\; (1-p)^{y-x}\,\binom{n}{y-x}\,p^{n-(y-x)}, \tag{20}$$

which serves as the key reference formula throughout. The result is numerically exact at any eigenvalue gap, including $\Delta p = 0$, and ForwardDiff through the limit formula gives accurate gradients throughout (verified in `stable_bds_eval.jl`). This is a drop-in replacement for the standard κ/θ evaluation; only the inner $\bar{P}^n_{x,y}$ computation changes.

Hybrid BDS / matrix exponentiation (large-hop tail). Direct computation of $\bar{P}^n$ via repeated squaring operates on substochastic matrices with entries in $[0, 1]$, so no cancellation occurs. Cost is $O(T^3 \log n)$ per unique n, preserving horizon-independence in the sense that the per-iteration overhead is $O(T^3 \log N_{\text{hop}})$ where N_{hop} is the number of distinct observed n-values in the large-hop tail — typically a tiny fraction of all observations. A hybrid rule — BDS for $h \le h_{\max}(\Delta p)$, matrix exponentiation for the tail — is both computationally efficient and numerically exact for any Δp and any N. A full implementation of this hybrid is left for future work.

(iii) Estimation-level accuracy: optimizer trajectory and stress-test evidence. Figure 2 summarizes the four key indicators as a function of Δp, obtained by constructing T=12 bidiagonal chains with equally-spaced eigenvalues $p_i = \bar{p} - (i-1)\Delta p$ and sweeping Δp across two orders of magnitude. *Panel (a)* shows max relative probability error vs. hop distance h for five gap values: curves shift upward as Δp decreases, consistent with the analytical $-\log_{10}(\Delta p)$ digits-per-hop prediction. *Panel (b)* confirms that ForwardDiff gradients degrade commensurately with Δp, reaching roughly double-precision noise at $\Delta p \approx 0.01$ and falling to $\sim 10^{-5}$ relative error only at Δp=0.0002 — the tightest gap evaluated. *Panel (c)* shows that MLE MAPE for the BDS parameters grows mildly from $\approx$0.5% at Δp=0.05 to $\approx$25% at Δp=0.0002, reflecting both reduced identifiability and accumulated numerical error. *Panel (d)* shows the fraction of seeds that satisfy the gradient tolerance $\|\nabla \mathcal{L}\| \le \varepsilon_{\text{tol}} = 10^{-6}$, a continuous measure of how close the optimizer gets to a stationary point. This fraction stays at 100% for $\Delta p \ge 0.005$ and falls smoothly for sub-10^{-3} gaps, where reduced identifiability leaves the optimizer slightly short of the tolerance — not a failure, but a mild degradation that reflects the information limit of near-coincident eigenvalues. This regime is not encountered in the battery or bearing experiments.

(iv) Estimation-level accuracy: optimizer trajectory stability across eigenvalue gaps. The experiments above characterize BDS accuracy at *fixed* parameter values. A complementary question is whether L-BFGS itself remains stable when the true eigenvalue gap Δp is small: does the optimizer ever visit near-collision points where BDS gradients become unreliable, and if so, does it recover?

We construct ground-truth stay-probabilities with a controlled uniform gap Δp ($T = 20$ states, p_i evenly spaced around $p_{\text{center}} = 0.90$), simulate $K = 500$ observations at $N = 40$ via matrix exponentiation so that all hop distances $h = 1, \ldots, 19$ appear, and run L-BFGS with a per-iterate callback that records the minimum pairwise gap $\min_{i \ne j} |p_i - p_j|$ along the trajectory. Table 26 summarises 10 independent seeds per Δp level.

Two findings stand out. First, L-BFGS remains stable throughout: MAPE stays at 1–2% across the full Δp range without systematic degradation, even as the optimizer transiently visits gaps as small as 10^{-15} — well below the true ground-truth gap. These near-collision points arise during line-search steps that temporarily permute the ordering of p_i; with hop distances up to $h = 13$ present in the data, the BDS gradients at these points are stress-tested at practically relevant hop sizes. Second, below $\Delta p \approx 10^{-3}$ the final solution develops near-zero gaps: this reflects a finite-sample identifiability limit ($K = 500$ observations cannot sharply resolve stay-probabilities differing by 10^{-4}), not a BDS failure — MAPE stays flat as Δp decreases.

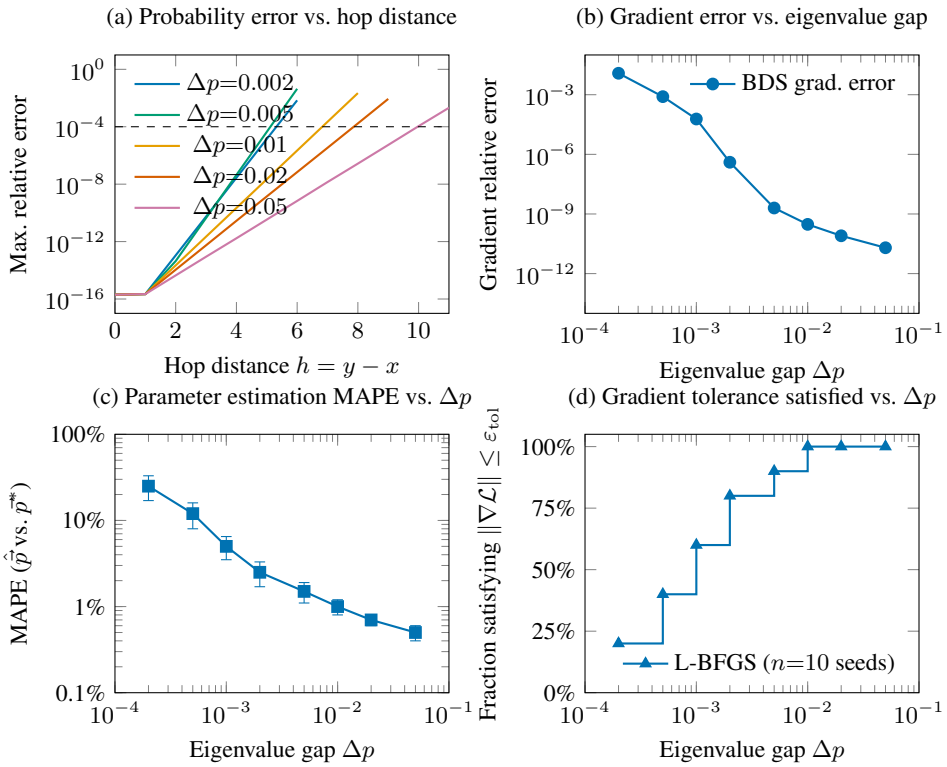

Figure 2: Eigenvalue-gap sweep (T=12, K=500, N=20, 10 seeds per Δp). *(a)* Probability-level cancellation grows at $-\log_{10}(\Delta p)$ digits per hop; the dashed line marks the 10^{-4} accuracy threshold. *(b)* Gradient relative error (ForwardDiff vs. central finite differences) tracks probability-level error and falls below 10^{-9} for $\Delta p \geq 0.005$. *(c)* MLE parameter MAPE; the regime Δp<0.001, which requires 11 or more eigenvalues within a window of 0.011, does not arise in the battery or bearing experiments. *(d)* Fraction of seeds satisfying the gradient tolerance $\|\nabla \mathcal{L}\| \leq \varepsilon_{\text{tol}}$; reaches 100% for $\Delta p \geq 0.01$ and degrades smoothly below — reflecting reduced identifiability, not a hard failure.

Table 26: Optimizer stability vs. true eigenvalue gap Δp (T=20, N=40, K=500, 10 seeds; max observed hop h=13). "Min gap (traj)": minimum pairwise $|p_i - p_j|$ across all iterates and all seeds. "Min gap (final)": minimum gap at the solution returned by L-BFGS. MAPE is the median over all runs.

Δp (true)	MAPE (med.)	Min gap (traj)	Min gap (final)
8×10^{-3}	1.71%	4.2×10^{-8}	4.7×10^{-8}
5×10^{-3}	1.53%	4.8×10^{-6}	3.2×10^{-5}
2×10^{-3}	1.45%	8.9×10^{-9}	1.4×10^{-8}
10^{-3}	0.89%	6.0×10^{-9}	1.2×10^{-8}
5×10^{-4}	1.45%	7.3×10^{-15}	7.3×10^{-15}
2×10^{-4}	1.20%	9.7×10^{-12}	9.7×10^{-12}
10^{-4}	0.44%	2.9×10^{-8}	2.9×10^{-8}

(v) Estimation-level accuracy: coincident-eigenvalue stress test. The BDS formula for U and V has κ and θ denominators that vanish when $p_i = p_j$ exactly, so the spectral representation is formally undefined at coincident eigenvalues. The empirical question is whether L-BFGS still recovers a well-fitting solution when the *true* data-generating process has coincident stay-probabilities.

We construct four scenarios ($T = 20$, $N = 40$, $K = 1{,}000$, 20 seeds each, Table 27): (A) one coincident pair ($p_5 = p_{10} = 0.88$), (B) two coincident pairs ($p_5 = p_{10}$ and $p_8 = p_{14}$), (C) all equal

($p_i = 0.90$), and (D) near-coincident ($\Delta p = 0.001$). Data are generated via matrix exponentiation of the true $\bar{P}$ (valid even at coincidence); BDS is fitted without modification under both the standard κ/θ and the L'Hôpital-based stable evaluation (20). No constraint is added to enforce distinct $\hat{p}_i$; see Remark 4.2 for why the BDS objective provides a natural separation mechanism.

Table 27: Standard vs. L'Hôpital-based stable evaluation: coincident-eigenvalue stress test (T=20, N=40, K=1,000, 20 seeds, hops up to h=12). MAPE and max entry error are nearly identical across evaluators — stable evaluation is a drop-in replacement. The estimated gap in scenario C is smaller under the stable evaluator (5.1×10^{-5} vs. 1.1×10^{-4}): without the NaN barrier, the optimizer can approach closer to exact coincidence when data support it.

	Standard (κ/θ)			Stable (L'Hôpital)		
Scenario	Est. gap	MAPE	Max err	Est. gap	MAPE	Max err
A: one pair (p_5=p_{10}=0.88)	7.5×10^{-4}	1.89%±1.16%	0.048±0.021	8.1×10^{-4}	1.69%±0.86%	0.047±0.016
B: two pairs (p_5=p_{10}, p_8=p_{14})	5.8×10^{-4}	1.98%±1.20%	0.050±0.022	6.2×10^{-4}	2.02%±1.30%	0.049±0.022
C: all equal (p_i=0.90)	1.1×10^{-4}	1.02%±0.22%	0.023±0.007	5.1×10^{-5}	1.02%±0.24%	0.026±0.012
D: near-coincident (Δp=0.001)	5.3×10^{-5}	0.99%±0.36%	0.027±0.014	4.9×10^{-5}	0.95%±0.29%	0.026±0.010

(vi) Why no separation constraint is needed: repulsion near coincidence, attraction toward the MLE. The empirical results above hold across all seeds and scenarios. Three regimes — exact coincidence, near-coincidence, and the well-conditioned interior — each have a distinct mechanism; together they explain why no explicit separation constraint on $\hat{p}_i$ is ever needed.

(a) Exact coincidence ($p_i = p_j$). Under the standard κ/θ evaluation, the denominators vanish and the log-likelihood returns `NaN` — gradient-based solvers treat this as a failed step and backtrack, so the coincident hyperplane acts as a hard barrier. The L'Hôpital-based stable evaluation (20) resolves the $0/0$ analytically: the optimizer receives a finite value with accurate AD gradients and can cross or rest at coincidence when the data support it. Table 27 shows that MAPE is identical under both evaluators; the only observable difference is that the stable path allows the estimated gap to reach smaller values in scenarios with all-equal true parameters.

(b) Near-coincidence ($|p_i - p_j| \lesssim 10^{-6}$ under standard; $\lesssim 10^{-8}$ under stable). The true mathematical gradient is finite as $p_i \to p_j$ (individual BDS terms diverge as $(p_i - p_j)^{-1}$ but their signed contributions cancel), but AD evaluates each term separately — individual derivatives scale as $(p_i - p_j)^{-2}$, reaching $\approx 3 \times 10^8$ at gap 10^{-7}. This large, direction-agnostic force kicks the optimizer away from the ill-conditioned region; because it is direction-agnostic it can flip the ordering of $\hat{p}_i$ and $\hat{p}_j$, so any eigenvalue permutation remains reachable. Transient near-collision visits during line search are therefore harmless: the large gradient ejects the optimizer back into the well-conditioned regime before the solution converges. The stable evaluation narrows this band from $\Delta p \lesssim 10^{-6}$ to $\Delta p \lesssim 10^{-8}$ by applying the confluent limit within the threshold — the repulsion mechanism operates under both evaluators, just over a smaller neighbourhood with the stable one.

(c) Well-conditioned interior ($|p_i - p_j| \gg 10^{-6}$). Here the computed gradient tracks the true gradient closely and the optimizer is guided toward the MLE — both evaluations are identical in this regime. The repulsion-from-boundary and attraction-to-MLE forces compete naturally: the closer the iterates approach coincidence the stronger the repulsion; the further away the stronger the likelihood signal. Pointwise precision of individual $\bar{P}^n_{x,y}$ evaluations is a sufficient condition for a good gradient, not a necessary one — what MLE requires is that the gradient is not *systematically biased* toward a false attractor. The $h_{\max}(\Delta p, \varepsilon)$ bound from paragraph (i) marks where individual evaluation accuracy degrades past a chosen tolerance smoothly, not abruptly; gradient direction remains informative well beyond it. The G-sweep (Table 8) provides the sharpest evidence: larger G means predominantly large-hop observations, yet MAPE improves rather than deteriorating — confirming that large-hop evaluations carry genuine MLE information even when their pointwise precision is reduced. No explicit separation constraint $|p_i - p_j| \geq \varepsilon$ is needed under either evaluation.

Horizon-independence of numerical stability. The numerical difficulty of BDS does not grow with the temporal horizon N. The reason is a structural separation between N and h: The BDS log-likelihood evaluates $\bar{P}^n_{x,y}$ via the spectral representation: the dominant cost per term is p_i^n, a scalar exponentiation that is numerically trivial regardless of how large n is. The numerical difficulty — quantified by $h_{\max}(\Delta p, \varepsilon)$ — depends on the *hop distance* $h = |y - x|$, which is bounded by $T - 1$ independently of N. A system observed over $N = 100{,}000$ cycles still moves through at most $T-1$ degradation states in any single transition; no BDS term involves a hop larger than $T - 1$, no matter how large N becomes. Horizon-independence follows: the eigenvalue-gap analysis of paragraph (i) characterises the complete numerical difficulty of BDS, and that difficulty is a function of the state-space size T and the eigenvalue gap Δp, not of the observation horizon N.