Tensor Decompositions Meet Control Theory: Learning General Mixtures of Linear Dynamical Systems

Ainesh Bakshi [*1] **Allen Liu** [*1] **Ankur Moitra** [*1] **Morris Yau** [*1]

Abstract

Recently Chen and Poor initiated the study of learning mixtures of linear dynamical systems. While linear dynamical systems already have wide-ranging applications in modeling time-series data, using mixture models can lead to a better fit or even a richer understanding of underlying subpopulations represented in the data. In this work we give a new approach to learning mixtures of linear dynamical systems that is based on tensor decompositions. As a result, our algorithm succeeds without strong separation conditions on the components, and can be used to compete with the Bayes optimal clustering of the trajectories. Moreover our algorithm works in the challenging partially-observed setting. Our starting point is the simple but powerful observation that the classic Ho-Kalman algorithm is a relative of modern tensor decomposition methods for learning latent variable models. This gives us a playbook for how to extend it to work with more complicated generative models.

1. Introduction

In this work, we study the problem of learning mixtures of linear dynamical systems from unlabelled trajectories. Each system evolves according to the following rules:

$$\begin{aligned} x_{t+1} &= A_i x_t + B_i u_t + w_t, \\ y_t &= C_i x_t + D_i u_t + z_t, \end{aligned} \tag{1}$$

Here the u_t's are control inputs to the system, the w_t's are the process noise and the z_t's are the observation noise. We observe the input-output sequence $(u_1, y_1), (u_2, y_2), \cdots, (u_T, y_T)$ and the goal is to learn the underlying system parameters. When there is only one system, this is a classic problem in control theory called system identification (Åström & Eykhoff, 1971; Ljung, 1998). A long line of recent works have established finite sample guarantees, often times from a single long trajectory, in increasingly more general settings (Hardt et al., 2018; Faradonbeh et al., 2018; Hazan et al., 2018; Simchowitz et al., 2018b; Oymak & Ozay, 2019; Tsiamis & Pappas, 2019; Sarkar et al., 2019; Simchowitz et al., 2019; Bakshi et al., 2023).

But what about mixture models? Instead of one long trajectory, we observe many short trajectories. The main complication is that they are unlabelled — we don't know which system generated which trajectories. This problem has many potential applications. For example, the microbiome is a community of microorganisms that live in a host. They play a key role in human health and are affected by our environment in complex ways. In scientific studies, the composition of the microbiome is monitored over extended periods of time and researchers model its behavior using dynamical systems to discover new biological insights (Gonze et al., 2018). But when these dynamics are heterogenous across a population, it is natural to use a mixture model instead. More generally, there are wide-ranging applications of dynamical systems in biology and engineering and in many of these settings using a mixture model can lead to a better fit, or even a richer understanding of any underlying subpopulations represented in the data.

However there is not much in the way of theoretical guarantees. In an important recent work, Chen and Poor gave the first efficient algorithms for learning mixtures of linear dynamical systems (Chen & Poor, 2022). This work received an ICML 2022 Outstanding Paper award. They employed a two-stage approach where they use coarse estimates to cluster the trajectories and then, based on their clustering, further refine their estimates. Essentially, they use the stationary covariances to find subspaces according to which the trajectories from the systems are well-separated.

In this work, we give a new approach for learning mixtures of linear dynamical systems that is based on tensor decompositions. Our algorithm (Theorem 7.1) achieves essentially optimal guarantees in many respects:

[*]Equal contribution [1]MIT. Correspondence to: Ainesh Bakshi <ainesh@mit.edu>, Allen Liu <cliu568@mit.edu>, Ankur Moitra <moitra@mit.edu>, Morris Yau <morrisy@mit.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

(1) Chen & Poor (2022) require a number of strong and difficult to interpret technical conditions on the parameters. In contrast, we give an efficient algorithm for clustering that succeeds whenever clustering is possible (Theorem 7.5). In particular, whenever the systems have negligible statistical overlap as distributions, we will be able to find a clustering that misclassifies only a negligible fraction of the trajectories.

(2) A priori it could be possible to learn the parameters of the mixture even when clustering is information-theoretically impossible. There is still useful information about the parameters that can be gleaned from the moments of the distribution. Indeed our algorithm succeeds under a condition we call joint nondegeneracy (Definition 3.4) which is a natural generalization of (individual) observability and controllability, both of which are standard assumptions in control theory and known to be necessary (Bakshi et al., 2023). These conditions hold even when the systems in the mixture model are almost entirely overlapping as distributions, rather than almost entirely disjoint and clusterable. Thus our algorithm brings results on learning mixtures of linear dynamical systems, which have complex time-varying behavior, in line with the strongest known guarantees for learning Gaussian mixture models (Kalai et al., 2010; Belkin & Sinha, 2010; Moitra & Valiant, 2010).

(3) Chen & Poor (2022) work in the fully-observed setting — i.e.

$$x_{t+1} = A_i x_t + B_i u_t + w_t$$

where we directly observe the sequence of inputs and states of the system $(u_1, x_1), (u_2, x_2), \cdots, (u_T, x_T)$. In contrast, our algorithms work in the more challenging partially-observed setting where we only get indirect measurements y_t of the hidden state. Even with just one system, this renders the maximum likelihood estimator a nonconvex optimization problem rather than a simpler linear regression problem. We also show that our algorithm succeeds with optimally short trajectories.

Finally, our algorithms are based on a surprisingly undiscovered connection. The classic approach going back to the 1960's for solving system identification is to estimate the Markov parameters

$$[CB, CAB, \cdots, CA^{2s}B]$$

and use the Ho-Kalman algorithm (HO & Kálmán, 1966). It turns out, the Ho-Kalman algorithm sets up a generalized eigenvalue problem, which just so happens to be the workhorse behind algorithms for low rank tensor decompositions. In recent years, tensor methods have become a mainstay in theoretical machine learning, particularly for learning mixture models (Mossel & Roch, 2005; Hsu & Kakade, 2013; Anandkumar et al., 2014). We leverage this connection along with modern tensor methods to teach the classic Ho-Kalman algorithm new tricks, namely we design a generalization of Ho-Kalman that can handle mixture models.

2. Technical Overview

Recall, a linear dynamical system $\mathcal{L}$ follows the Markov process described in Equation (1), where A, B, C, D are matrices with dimensions $n \times n$, $n \times p$, $m \times n$ and $m \times p$ respectively. The random variables w_t and z_t are typically modeled as standard normal corresponding to process and measurement noise. In the most general setting, $\{y_t, u_t\}_{t\in[l]}$ is the dataset from which we wish to infer the system parameters A, B, C, and D. Note that it is only possible to recover the system parameters under an equivalence class of similarity transforms. A standard recipe for this task is the algorithm of Ho-Kalman which succeeds at recovering $\hat{A}$ such that there exists a similarity transform U satisfying $\left\|A - U\hat{A}U^{-1}\right\| = 0$ with analogous guarantees for $\hat{B}, \hat{C}$, and $\hat{D}$ in infinite samples.

The crux of the Ho-Kalman algorithm is to first estimate "Markov parameters" of the form CA^iB for varying values of $i \in \mathbb{Z}^+$. The Markov parameters are arranged in a corresponding Hankel matrix and an Eigendecomposition style procedure is applied to the Hankel matrix to recover the system parameters (see Algorithm 3). The key is to estimate Markov parameters which is difficult when the data $\{y_t, u_t\}_{t\in[l]}$ is drawn from a mixture of linear dynamical systems defined next.

Definition 2.1 (Mixture of LDS's)**.** A mixture of linear dynamical systems is represented as $\mathcal{M} = w_1\mathcal{L}_1 + \ldots w_k\mathcal{L}_k$, where $w_1, \ldots, w_k$ are positive real numbers summing to 1 and $\mathcal{L}_1((A_1, B_1, C_1, D_1), \ldots, \mathcal{L}_k(A_k, B_k, C_k, D_k)$ are each individual linear dynamical systems with the same dimensions (i.e. the same m, n, p). The trajectories we observe are sampled according to the following process. First an index $i \in [k]$ is drawn according to the mixing weights $w_1, \ldots, w_k$ and then a trajectory of length l, denoted by $\{(u_1, y_1), \ldots (u_\ell, y_\ell)\}$ is drawn from the corresponding dynamical system $\mathcal{L}_i$.

We obtain as input, N trajectories, each denoted by $\{(u_1^j, y_1^j), \ldots, (u_\ell^j, y_\ell^j)\}$, for $j \in [N]$. Our goal is to learn the parameters of the mixture, i.e. the individual linear dynamical systems and their mixing weights, given polynomially many samples from the mixture. In this setting, if the trajectory length l is large enough for the system parameters to be learned from a single trajectory then it would be possible to learn each dynamical system $\mathcal{L}$ separately (Bak-

shi et al., 2023). The question is whether we can learn the Markov parameters when l is small. Our general strategy is as follows. For a particular Markov parameter CA^iB we compute a carefully chosen 6-th order tensor that can be estimated from the control inputs (u_t's) and observation (y_t's). In particular, for a fixed t, given N trajectories, we construct:

$$\widehat{T}_i = \frac{1}{N}\sum_{j\in[N]} y^j_{t+3i+2} \otimes u^j_{t+2i+2} \otimes y^j_{t+2i+1} \otimes u^j_{t+i+1} \otimes y^j_{t+i} \otimes u^j_i.$$

We show that $\widehat{T}_i$ is an unbiased estimator of a tensor whose components are the Markov parameters (see Lemma 5.4):

$$\widehat{T}_i \sim \sum_{j\in[k]} w_j \left(C_jA^i_jB_j\right) \otimes \left(C_jA^i_jB_j\right) \otimes \left(C_jA^i_jB_j\right) \quad (2)$$

Brushing aside issues of sample complexity, we can assume we have access to the tensor in Eqn (2). Ideally, we would just like to read off the components of this tensor and obtain the Markov parameters.

However, provably recovering the components requires this tensor to be non-degenerate. To this end, we flatten the tensor along its first and second, third and fourth, and fifth and sixth modes to obtain a 3-rd order tensor, whose components are the Markov paramters of the j-th LDS, flattened to a vector. In particular, we have

$$\tilde{T}_i = \sum_{j\in[k]} w_j v\left(C_jA^i_jB_j\right) \otimes v\left(C_jA^i_jB_j\right) \otimes v\left(C_jA^i_jB_j\right),$$

where $v\left(C_jA^i_jB_j\right)$ simply flattens the matrix $C_jA^i_jB_j$. The crux of our analysis is to show that the Joint Non-degeneracy condition (see Definition 3.4) implies that components of the 3-rd order tensor are (robustly) linearly independent (Lemma A.4).

Once we have established linear independence, we can run Jennrich's tensor decomposition algorithm (Algorithm 4) on $\tilde{T}_i$ to obtain the components $w_jv\left(C_jA^i_jB_j\right) \otimes v\left(C_jA^i_jB_j\right) \otimes v\left(C_jA^i_jB_j\right)$. Assuming we know the mixing weights, we can just read off the first mode of this tensor, and construct the Markov parameter matrix. Once we have the Markov parameters, we can run (robust) Ho-Kalman (Algorithm 3) to recover the A_j, B_j, C_j's.

However, in the setting where the mixing weights are unknown, we cannot hope simply read off the Markov parameter matrix from the component above. Instead, we can obtain the vectors $\tilde{v}_j = w_j^{1/3} v\left(C_jA^i_jB_j\right)$, for all $j \in [k]$, by simply reading the first mode and dividing out by the Frobenius norm of the second and third mode. We set up a regression problem where we solve for the coefficients $c_1, \ldots c_k$ as follows:

$$\min_{c_1,c_2,\ldots,c_k} \left\| \sum_{\ell\in[k]} c_\ell \tilde{v}_\ell - \sum_{j\in[k]} w_jC_jA^i_jB_j \right\|_2^2, \quad (3)$$

where we can estimate $\sum_{j\in[k]} w_jC_jA^i_jB_j$ up to arbitrary polynomial accuracy using the input samples. We show that the solution to this regression problem results in c_ℓ's that are non-negative and $c_\ell \sim w_\ell^{2/3}$ for all $\ell \in [k]$, which suffices to learn the mixing weights (see Theorem 7.3 for details). We describe our complete algorithm in Section 6 and the analysis of each sub-routine in Section 7. Given space constraints, we defer all technical proofs to the Appendix.

3. Formal Setup and Assumptions

Our input is a set of N length ℓ trajectories generated according to a mixture $\mathcal{M}$, as defined in Model 2.1. Our goal is to learn the parameters of the mixture, i.e. the individual linear dynamical systems and their mixing weights, given polynomially many samples such trajectories.

For simplicity, throughout this paper, we will consider when all of the noise distributions are isotropic Gaussians i.e. $\mathcal{D}_0 = N(0, I_n), \mathcal{D}_u = N(0, I_p), \mathcal{D}_w = N(0, I_n), \mathcal{D}_z = N(0, I_m)$ although our results generalize to more general noise distributions as long as they have sufficiently many bounded moments. Throughout this paper, for a matrix A we will use $A^\top$ to denote its transpose and $A^\dagger$ to denote its pseudo-inverse. We use $\|A\|$ to denote its operator norm and $\|A\|_F$ to denote its Frobenius norm.

3.1. Assumptions for Learnability

We begin with standard definitions of the observability and controllability matrix of a single LDS.

Definition 3.1 (Observability Matrix)**.** For an LDS $\mathcal{L}(A, B, C, D)$ and an integer s, define the matrix $O_{\mathcal{L},s} \in \mathbb{R}^{sm\times n}$ as

$$O_{\mathcal{L},s} = \begin{bmatrix} C^\top & (CA)^\top & \ldots \left(CA^{s-1}\right)^\top \end{bmatrix}^\top.$$

A LDS is *observable* if for some s, the matrix O_s is full-rank. Similarly, we need to ensure that the control input is not degenerate, and only acts in a subspace that is not spanned by A. This is made precise by considering the *controllability matrix*:

Definition 3.2 (Controllability Matrix)**.** For an LDS $\mathcal{L}(A, B, C, D)$ and an integer s, define the matrix $Q_{\mathcal{L},s} \in \mathbb{R}^{n\times sp}$ as

$$Q_{\mathcal{L},s} = \begin{bmatrix} B & AB & \ldots & A^{s-1}B \end{bmatrix}$$

A LDS is *controllable* if the controllability matrix is full-rank. These two assumptions are necessary for the LDS

to be learnable and in fact it is necessary to make a quantitatively robust assumption of this form (see (Bakshi et al., 2023)). In other words, we need a bound on the condition number of the observability and controllability matrices.

In addition to the assumptions required to learn a single linear dynamical system, we will require addition assumptions on the interaction of the LDS's to obtain learning algorithms for the mixture (as otherwise there could be degeneracies such as two components being almost the same which would make it information-theoretically impossible to learn).

3.1.1. Joint Nondegeneracy

We introduce a joint nondegeneracy condition that prevents certain degeneracies arising from the interaction between the components of the mixture e.g. if the components are too close to each other.

Definition 3.3 (Markov Parameters)**.** Given a linear dynamical system, $\mathcal{L}(A, B, C, D)$, and an integer $T \geqslant 1$, the Markov Parameter matrix $G_{\mathcal{L},T} \in \mathbb{R}^{m\times(T+1)p}$ is defined as the following block matrix:

$$G_{\mathcal{L},T} = \begin{bmatrix} D & CB & CAB & \dots & CA^{T-1}B \end{bmatrix}.$$

Definition 3.4 (Joint Non-degeneracy)**.** For a mixture of LDS $\mathcal{M} = w_1\mathcal{L}_1 + \cdots + w_k\mathcal{L}_k$ where each individual LDS is given by $\mathcal{L}_i = \mathcal{L}(A_i, B_i, C_i, D_i)$ (with the same dimension parameters m, n, p), we say $\mathcal{M}$ is (γ, s)-jointly nondegenerate if for any real numbers $c_1, \dots, c_k$ with $c_1^2 + \cdots + c_k^2 = 1$, we have

$$\|c_1 G_{\mathcal{L}_1,s} + \cdots + c_k G_{\mathcal{L}_k,s}\|_F \geqslant \gamma\,.$$

We now state precisely the entire set of assumptions about the mixture $\mathcal{M}$ that we require.

Definition 3.5 (Well Behaved Mixture of LDS)**.** We say a mixture of LDS $\mathcal{M} = w_1\mathcal{L}_1 + \cdots + w_k\mathcal{L}_k$ where each $\mathcal{L}_i = \mathcal{L}(A_i, B_i, C_i, D_i)$ is well-behaved if the following assumptions hold

- **Non-trivial Mixing Weights:** for some $w_{\min} > 0$, we have $w_i \geqslant w_{\min}$ for all $i \in [k]$.
- **Non-trivial Individual Controllers and Measurements:** for all $i \in [k]$, $\|B_i\|, \|C_i\| \geqslant 1$
- **Individual Boundedness:** for some parameter κ,

$$\|A_i\|, \|B_i\|, \|C_i\|, \|D_i\| \leqslant \kappa \text{ for all } i \in [k].$$

- **Individual Observability and Controllability:** for some integer s and parameter κ, for all $i \in [k]$ the matrix $O_{\mathcal{L}_i,s}$ has full column rank, the matrix $Q_{\mathcal{L}_i,s}$ has full row rank and

$$\sigma_{\max}(O_{2s})/\sigma_{\min}(O_s) \leqslant \kappa,$$
$$\sigma_{\max}(Q_{2s})/\sigma_{\min}(Q_s) \leqslant \kappa.$$

- **Joint Nondegeneracy:** The mixture $\mathcal{M}$ is (γ, s) jointly nondegenerate for some parameter $\gamma > 0$.

The assumptions on the individual components mirror those in (Bakshi et al., 2023) where a more detailed discussion and justification can be found.

4. Related Work

There is a long history of work on identifying/learning linear dynamical systems from measurements (Ding, 2013; Zhang, 2011; Spinelli et al., 2005; Simchowitz et al., 2019; 2018a; Sarkar & Rakhlin, 2019; Faradonbeh et al., 2017; Shah et al., 2012; Hardt et al., 2018; Hazan et al., 2018; 2017). See (Galrinho, 2016) for a more extensive list of references. These works focus on learning a the parameters of a linear dynamical system from a single long trajectory. There has also been extensive empirical work on mixtures of time series and trajectories which have been successfully applied in a variety of domains such as neuroscience, biology, economics, automobile design and many others (Bulteel et al., 2016; Mezer et al., 2009; Li, 2000; Kalliovirta et al., 2016; Hallac et al., 2017).

Our setup can be viewed as a generalization of the more classical problem of learning mixtures of linear regressions which has been extensively studied theoretically (Chen et al., 2013; Yi et al., 2013; Li & Liang, 2018; Chen et al., 2019; Kwon et al., 2020; Diamandis et al., 2021). The fact that we receive many short trajectories parallels meta-learning framework in (Kong et al., 2020b;a). However, the system dynamics in our setting (which are not present in standard mixed linear regression) make our problem significantly more challenging. It also has connections to super-resolution (Candès & Fernandez-Granda, 2014; Moitra, 2015; Chen & Moitra, 2021) where tensor methods have also been employed (Huang & Kakade, 2015). Finally, our model is similar to the well-studied switched linear dynamical system model (see (Fox et al., 2008; Mudrik et al., 2022) and references therein).

5. Moment Statistics of Linear Dynamical Systems

We begin with some basic properties of a single linear dynamical system $\mathcal{L} = \mathcal{L}(A, B, C, D)$.

Fact 5.1 (Algebraic Identities for LDS's). Let $\mathcal{L}(A, B, C, D)$ be a Linear Dynamical System. Then, for any $t \in \mathbb{N}$,

$$y_t = \sum_{i=1}^{t} \left(CA^{i-1}Bu_{t-i} + CA^{i-1}w_{t-i}\right) + CA^t x_0 + Du_t + z_t\,.$$

Fact 5.2 (Cross-Covariance of Control and Observation). For any $t, k \in \mathbb{N}$, and any $0 \leqslant j \leqslant k$, given observations y_t and control inputs u_t from a linear dynamical system $\mathcal{L}(A, B, C, D)$ such that $\mathbb{E}\left[u_t u_t^\top\right] = I$ and the u_t's are independent, we have $\mathbb{E}\left[y_{t+j} u_t^\top\right] = D$, if $j = 0$, and $\mathbb{E}\left[y_{t+j} u_t^\top\right] = CA^{j-1}B$, otherwise.

In light of the above, we make the following definition.

Definition 5.3 (System Parameters)**.** For an LDS $\mathcal{L}(A, B, C, D)$ and an integer $j \geqslant 0$, we define the matrix $X_{\mathcal{L},j} = D$ if $j = 0$ and $X_{\mathcal{L},j} = CA^{j-1}B$ if $j > 0$.

Next, we show that the sixth moment tensor we consider, restricted to a single LDS, is a indeed a tensor of the system parameters.

Lemma 5.4 (Sixth-moment Statistics)**.** *Given a linear dynamical system $\mathcal{L}(A, B, C, D)$ and integers $t, k_1, k_2, k_3 \geqslant 0$, let $t_1 = t + k_1$, $t_2 = t_1 + k_2$ and $t_3 = t_2 + k_3$. Then, we have*

$$\mathbb{E}\left[y_{t_3+2} \otimes u_{t_2+2} \otimes y_{t_2+1} \otimes u_{t_1+1} \otimes y_{t_1} \otimes u_t\right] = X_{\mathcal{L},k_3} \otimes X_{\mathcal{L},k_2} \otimes X_{\mathcal{L},k_1}\,,$$

where $X_{\mathcal{L},j}$ is defined in Definition 5.3.

We defer the proof to the Appendix.

Now consider a mixture of LDS $\mathcal{M} = w_1\mathcal{L}_1 + \cdots + w_k\mathcal{L}_k$ where $\mathcal{L}_i = \mathcal{L}(A_i, B_i, C_i, D_i)$. Using Lemma 5.4, we have an expression for the sixth moments of the mixture.

Corollary 5.5. *For a mixture of LDS $\mathcal{M} = w_1\mathcal{L}_1 + \cdots + w_k\mathcal{L}_k$ and for $t, k_1, k_2, k_3 \geqslant 0$, let $t_1 = t + k_1$, $t_2 = t_1 + k_2$ and $t_3 = t_2 + k_3$. Then,*

$$\mathop{\mathbb{E}}_{\mathcal{M}}\left[y_{t_3+2} \otimes u_{t_2+2} \otimes y_{t_2+1} \otimes u_{t_1+1} \otimes y_{t_1} \otimes u_t\right] = \sum_{i=1}^{k} w_i X_{\mathcal{L}_i,k_3} \otimes X_{\mathcal{L}_i,k_2} \otimes X_{\mathcal{L}_i,k_1}.$$

Proof. This follows directly from Lemma 5.4. □

6. Algorithm

In this section, we describe our algorithm for learning a mixture of Linear Dynamical Systems. At a high level, our algorithm uses multiple trajectories to obtain an estimate of the tensor

$$\Pi_{\mathcal{M}} = \sum_{i\in[k]} w_i G_{\mathcal{L}_i,2s} \otimes G_{\mathcal{L}_i,2s} \otimes G_{\mathcal{L}_i,2s}\,.$$

where

$$G_{\mathcal{L}_i,s} = \begin{bmatrix} D_i & C_iB_i & C_iA_iB_i & \dots & C_iA_i^{s-1}B_i \end{bmatrix}.$$

Algorithm 1 Learning a Mixture of LDS's

Input: N sample trajectories of length l from a mixture of LDS $\mathcal{M} = \sum_{i\in[k]} w_i \mathcal{L}\left(A_i, B_i, C_i, D_i\right)$ denoted $\left\{(y_1^i, \dots, y_l^i)\right\}_{i\in[N]}$, the corresponding control inputs $\left\{(u_1^i, \dots, u_l^i)\right\}_{i\in[N]}$, parameter $s \in \mathbb{N}$ for individual observability and controllability and joint nondegeneracy, Accuracy parameter $0 < \varepsilon < 1$ and allowable failure probability $0 < \delta < 1$.

Operation:

1. Run Algorithm 2 on the input samples and let $\left\{\widetilde{G}_i\right\}_{i\in[k]}$ be the matrices returned
2. For $0 \leqslant k_1 \leqslant 2s$, compute estimate $\widehat{R}_{k_1}$ of $\mathbb{E}_{\mathcal{M}}\left[y_{k_1+1} \otimes u_1\right]$ as
$$\widehat{R}_{k_1} = \frac{1}{N}\sum_{i=1}^{N} y_{k_1+1}^i \otimes u_1^i\,.$$
3. Construct estimate $\widehat{R}_{\mathcal{M}}$ of $R_{\mathcal{M}}$ by stacking together estimates $\widehat{R}_0, \widehat{R}_1, \dots, \widehat{R}_{2s-1}$
4. Solve for weights $\widetilde{w_1}, \dots, \widetilde{w_k}$ that minimize
$$\|\widetilde{w_1}\widetilde{G}_1 + \cdots + \widetilde{w_k}\widetilde{G}_k - R_{\mathcal{M}}\|_F$$
5. Set $\widehat{G}_i = \widetilde{G}_i/\sqrt{\widetilde{w_i}}$
6. Set $\widehat{w}_i = \widetilde{w_i}^{3/2}$ for all $i \in [k]$
7. Run Algorithm 3 on $\hat{G}_i$ for each $i \in [k]$ to recover parameters $\left\{\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i\right\}_{i\in[k]}$

Output: The set of parameter estimates $\left\{\hat{w}_i, \hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i\right\}_{i\in[k]}$

Recall that $G_{\mathcal{L}_i,2s}$ has blocks that are of the form $X_{\mathcal{L}_i,s'}$ for $s' \leqslant 2s$ and thus it follows from Corollary 5.5 that we can construct a unbiased estimates of the individual blocks

$$T_{s_1,s_2,s_3} = \sum_{i\in[k]} w_i X_{\mathcal{L}_i,s} \otimes X_{\mathcal{L}_i,s_2} \otimes X_{\mathcal{L}_i,k_1}$$

of this tensor from the observations and control input. Piecing together the individual blocks lets us construct an estimate of $\Pi_{\mathcal{M}}$. Since we have access to multiple *independent* trajectories, we can show that the variance is bounded and we indeed have access to a tensor close to $\Pi_{\mathcal{M}}$.

We then run the classical Jennrich's tensor decomposition algorithm on the tensor $\Pi_{\mathcal{M}}$ to recover the factors $G_{\mathcal{L}_i,2s}$. The key is that the joint nondegeneracy assumption im-

plies that vectors obtained by flattening $G_{\mathcal{L}_1,2s}, \ldots, G_{\mathcal{L}_k,2s}$ are (robustly) linear independent. Therefore, Jennrich's algorithm indeed recovers the factors $G_{\mathcal{L}_1,2s}, \ldots, G_{\mathcal{L}_k,2s}$. These are exactly the Markov parameters of the individual components and we can then invoke a robust variant of Ho-Kalman (Oymak & Ozay, 2019) to recover the corresponding parameters.

Due to some minor technical complications from the unknown mixing weights, it will also be useful to define

$$R_{\mathcal{M}} = \sum_{i\in[k]} w_i G_{\mathcal{L}_i,2s}$$

which we can also estimate empirically by estimating each block

$$R_{s_1} = \mathbb{E}_{\mathcal{M}}[y_{k_1+1} \otimes u_1] = \sum_{i\in[k]} w_i X_{\mathcal{L}_i,s_1}$$

separately.

Required Trajectory Length: Our algorithm requires trajectories of length $\sim 6s$ where s is the observability/controllability parameter. Note that trajectories of length s are necessary as otherwise the parameters for even a single system are not uniquely recoverable so our required trajectory length is minimal up to this factor of 6.

7. Analysis

In this section, we provide the analysis of the algorithms we presented in Section 6. The main theorem we obtain is as follows:

Theorem 7.1 (Learning a Mixture of LDS's)**.** *Given $0 < \epsilon, \delta < 1$, an integer s, and*

$$N = \text{poly}\left(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\varepsilon, 1/\delta\right)$$

observations $\left\{(y_1^i, \ldots, y_\ell^i)\right\}_{i\in[N]}$, and the corresponding control inputs $\left\{(u_1^i, \ldots, u_\ell^i)\right\}_{i\in[N]}$ of trajectory length $\ell \geqslant 6(s+1)$, from a mixture of linear dynamical system $\mathcal{M} = \sum_{i\in[k]} w_i \mathcal{L}\left(A_i, B_i, C_i, D_i\right)$, satisfying the assumptions in Section 3, Algorithm 1 outputs estimates $\left\{\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i\right\}_{i\in[k]}$ such that with probability at least $1-\delta$, there is a permutation π on $[k]$ such that for each $i \in [k]$, there exists a similarity transform U_i satisfying

$$\max\Big(\left\|A_{\pi(i)} - U_i^{-1}\hat{A}_i U_i\right\|, \left\|C_{\pi(i)} - \hat{C}_i U_i\right\|, \left\|B_{\pi(i)} - U_i^{-1}\hat{B}_i\right\|, \left\|D_{\pi(i)} - \hat{D}_i\right\|, |w_{\pi(i)} - \hat{w}_i|\Big) \leqslant \varepsilon.$$

Further, Algorithm 1 runs in $\text{poly}\left(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\varepsilon, 1/\delta\right)$ *time.*

Algorithm 2 Learn Individual Markov Parameters

Input: N sample trajectories of length l from a mixture of LDS $\mathcal{M} = \sum_{i\in[k]} w_i \mathcal{L}\left(A_i, B_i, C_i, D_i\right)$, denoted by $\left\{(y_1^i, \ldots y_\ell^i)\right\}_{i\in[N]}$, the corresponding control inputs, denoted $\left\{(u_1^i, \ldots, u_\ell^i)\right\}_{i\in[N]}$, parameter $s \in \mathbb{N}$ for individual observability and controllability and joint non-degeneracy, accuracy parameter ε and allowable failure probability δ.

Operation:

1. For $0 \leqslant k_1 \leqslant 2s, 0 \leqslant k_2 \leqslant 2s, 0 \leqslant k_3 \leqslant 2s$,
   (a) Compute empirical estimate $\widehat{T}_{k_1,k_2,k_3}$ as follows:
   $$\widehat{T}_{k_1,k_2,k_3} = \frac{1}{N}\sum_{i\in[n]} y^i_{k_1+k_2+k_3+3} \otimes u^i_{k_1+k_2+3} \otimes y^i_{k_1+k_2+2} \otimes u^i_{k_1+2} \otimes y^i_{k_1+1} \otimes u^i_1$$
2. Construct estimate $\widehat{\Pi}_{\mathcal{M}}$ for $\Pi_{\mathcal{M}}$ by piecing together the blocks $\widehat{T}_{k_1,k_2,k_3}$ appropriately
3. Flatten pairs of dimensions of $\widehat{\Pi}_{\mathcal{M}}$ so that it is a order-3 tensor with dimensions $(2s+1)mp \times (2s+1)mp \times (2s+1)mp$
4. Run Jennrich's algorithm (Algorithm 4) to obtain the following decomposition
   $$\widehat{\Pi}_{\mathcal{M}} = \widehat{T}_1 + \cdots + \widehat{T}_k$$
5. For each $\widehat{T}_i$, compute the Frobenius norm of each slice in its second and third dimensions to obtain a vector $\widehat{v}_i \in \mathbb{R}^{(2s+1)mp}$
6. Construct $\widetilde{G}_i$ by rearranging the vector $\widehat{v}_i/\|\widehat{v}_i\|^{2/3}$ back into an $m \times (2s+1)p$ matrix (undoing the flattening operation)

Output: Matrices $\widetilde{G}_1, \ldots, \widetilde{G}_k$

We proceed by analyzing each sub-routine separately. In particular, Algorithm 1 proceeds by first taking the input samples and running Algorithm 2 to learn the individual sets of Markov parameters up to some scaling by the mixing weights. Formally,

Theorem 7.2 (Recovering the Markov Parameters)**.** *Given $\varepsilon, \delta > 0$ and*

$$N \geqslant \text{poly}(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\varepsilon, 1/\delta)$$

trajectories from a mixture of LDS's, $\mathcal{M} = \sum_{i\in[k]} w_i \mathcal{L}\left(A_i, B_i, C_i, D_i\right)$, Algorithm 2 outputs a

Algorithm 3 Parameter Recovery via Ho-Kalman (Oymak & Ozay, 2019)

Input: Parameter s, Markov parameter matrix estimate $\hat{G} = [\hat{X}_0, \ldots, \hat{X}_{2s}]$

Operation:

1. Set $\hat{D} = \hat{X}_0$
2. Form the Hankel matrix $\hat{H} \in \mathbb{R}^{ms \times p(s+1)}$ from $\hat{G}$ as
$$\hat{H} = \begin{bmatrix} \hat{X}_1 & \hat{X}_2 & \ldots & \hat{X}_{s+1} \\ \hat{X}_2 & \hat{X}_3 & \ldots & \hat{X}_{s+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{X}_s & \hat{X}_{s+1} & \ldots & \hat{X}_{2s} \end{bmatrix}$$
3. $\hat{H}^- \in \mathbb{R}^{ms \times ps} \leftarrow$ first ps columns of $\hat{H}$
4. $\hat{L} \in \mathbb{R}^{ms \times ps} \leftarrow$ rank n approximation of $\hat{H}^-$ obtained via SVD
5. $U, \Sigma, V = SVD(\hat{L})$
6. $\hat{O} \in \mathbb{R}^{ms \times n} \leftarrow U\Sigma^{1/2}$
7. $\hat{Q} \in \mathbb{R}^{n \times ps} \leftarrow \Sigma^{1/2} V^\top$
8. $\hat{C} \leftarrow$ first m rows of $\hat{O}$
9. $\hat{B} \leftarrow$ first p column of $\hat{Q}$
10. $\hat{H}^+ \in \mathbb{R}^{ms \times ps} \leftarrow$ last ps column of $\hat{H}$
11. $\hat{A} \leftarrow \hat{O}^\dagger \hat{H}^+ \hat{Q}^\dagger$

Output: $\hat{A} \in \mathbb{R}^{n \times n}, \hat{B} \in \mathbb{R}^{n \times p}, \hat{C} \in \mathbb{R}^{m \times n}, \hat{D} \in \mathbb{R}^{m \times p}$

set of matrices $\widetilde{G}_1, \widetilde{G}_2, \ldots \widetilde{G}_k$ such that with probability $1 - \delta$, there is a permutation π on $[k]$ such that

$$\|\widetilde{G}_{\pi(i)} - w_i^{1/3} G_{\mathcal{L}_i, 2s}\|_F \leqslant \varepsilon$$

for all $i \in [k]$. Further, Algorithm 2 runs in $\mathrm{poly}(N)$ time.

Next, we argue about the mixing weights $\widetilde{w}_1, \ldots, \widetilde{w}_k$ computed in the regression step in Algorithm 1.

Theorem 7.3 (Recovering the Mixing Weights)**.** *Assume that the matrices $\widetilde{G}_i$ computed in Algorithm 1 satisfy Theorem 7.2. Then, with $1 - \delta$ probability, the mixing weights $\widetilde{w}_1, \ldots, \widetilde{w}_k$ computed in Algorithm 1 satisfy*

$$|\widetilde{w}_{\pi(i)} - w_i^{2/3}| \leqslant \varepsilon \cdot \mathrm{poly}(\kappa, m, n, s, p, 1/\gamma, 1/w_{\min})$$

for all $i \in [k]$.

Proof. Recall by Fact 5.2 and Definition 5.3 that $\mathbb{E}[\widehat{R}_{\mathcal{M}}] = R_{\mathcal{M}}$. Also by the same argument as in the proof of Lemma A.3, the empirical estimate concentrates with high probability since the observations and control inputs are jointly Gaussian with bounded covariance. Thus, with $1 - \delta$ probability, we have $\|R_{\mathcal{M}} - \widehat{R}_{\mathcal{M}}\|_F \leqslant \varepsilon$. Recalling the definition of $R_{\mathcal{M}}$ and applying Theorem 7.2, we must have that

$$\|\widehat{R}_{\mathcal{M}} - w_i^{2/3} \widetilde{G}_{\pi(i)}\|_F \leqslant \varepsilon(k+1)\,.$$

Now consider any other set of choices for $\widetilde{w}_{\pi(i)}$. We must have that

$$\|\sum_{i=1}^{k} (w_i^{2/3} - \widetilde{w}_{\pi(i)}) \widetilde{G}_{\pi(i)}\|_F \leqslant 2(k+1)\varepsilon\,.$$

On the other hand we can write

$$\begin{aligned} &\left\|\sum_{i=1}^{k} (w_i^{2/3} - \widetilde{w}_{\pi(i)}) \widetilde{G}_{\pi(i)}\right\|_F \\ &\geqslant \left\|\sum_{i=1}^{k} (w_i^{2/3} - \widetilde{w}_{\pi(i)}) w_i^{1/3} G_{\mathcal{L}_i, 2s}\right\|_F \\ &\quad - \varepsilon \sum_{i=1}^{k} |w_i^{2/3} - \widetilde{w}_{\pi(i)}|\,. \end{aligned}$$

Now for any coefficients $c_1, \ldots, c_k$, we have

$$\|c_1 G_{\mathcal{L}_1, 2s} + \cdots + c_k G_{\mathcal{L}_k, 2s}\|_F \geqslant \frac{\gamma(|c_1| + \cdots + |c_k|)}{\sqrt{k}}$$

where we used the joint nondegeneracy assumption. Thus,

$$\begin{aligned} &\left\|\sum_{i=1}^{k} (w_i^{2/3} - \widetilde{w}_{\pi(i)}) \widetilde{G}_{\pi(i)}\right\|_F \\ &\geqslant \frac{\gamma w_{\min}^{1/3} \sum_{i=1}^{k} |w_i^{2/3} - \widetilde{w}_{\pi(i)}|}{\sqrt{k}} - \varepsilon \sum_{i=1}^{k} |w_i^{2/3} - \widetilde{w}_{\pi(i)}| \\ &\geqslant \left(\frac{\gamma w_{\min}^{1/3}}{\sqrt{k}} - \varepsilon\right) \max_i (|w_i^{2/3} - \widetilde{w_{\pi(i)}}|)\,. \end{aligned}$$

Combining this with the previous inequality gives the desired bound. □

As a corollary to the above two theorems, the estimates $\widehat{G}_i$ computed in Algorithm 1 are actually good estimates for the true individual Markov parameters $G_{\mathcal{L}_i, 2s}$. Now, running a stable variant of Ho-Kalman (Oymak & Ozay, 2019) on the individual block Henkel matrices suffices to obtain estimates $\widehat{A}_i, \widehat{B}_i, \widehat{C}_i, \widehat{D}_i$. Formally,

Theorem 7.4 (Stable Ho-Kalman, (Oymak & Ozay, 2019))**.** *For observability and controllability matrices that are rank n, the Ho-Kalman algorithm applied to $\hat{G}$ produces estimates $\hat{A}, \hat{B}$, and $\hat{C}$ such that there exists similarity transform $T \in \mathbb{R}^{n \times n}$ such that*

$$\max\{\|C - \hat{C}T\|_F, \|B - T^{-1}\hat{B}\|_F\} \leqslant 5\sqrt{n\|G - \hat{G}\|}$$

and

$$\|A - T^{-1}\hat{A}T\|_F \leqslant \frac{\sqrt{n\|G-\hat{G}\|}\|H\|}{\sigma_{min}^{3/2}(H^-)}$$

and

$$\left\|D - \hat{D}\right\|_F \leqslant \sqrt{n}\left\|G - \hat{G}\right\|$$

where in the above

$$G = [D, CB, CAB, ..., CA^{2s-1}B]$$

and H is the Hankel matrix constructed with the true parameters G.

Putting together the above theorems, we can prove our main result.

Proof of Theorem 7.1. The proof follows from simply combining the theorems above (rescaling ε appropriately by a polynomial in the other parameters). Note that for each $i \in [k]$, the Hankel matrix H_i with the true parameters, constructed in the Ho-Kalman algorithm satisfies $\|H_i\| \leqslant \sigma_{max}(\mathcal{O}_{\mathcal{L}_i,s})\sigma_{max}(\mathcal{Q}_{\mathcal{L}_i,s}) \leqslant \text{poly}(\kappa, s)$. We also have $\sigma_{min}(H_i^-) \geqslant \sigma_{min}(\mathcal{O}_{\mathcal{L}_i,s})\sigma_{min}(\mathcal{Q}_{\mathcal{L}_i,s}) \geqslant 1/\text{poly}(\kappa)$ (see Claim A.1 and Claim A.2). Thus, we can indeed apply Theorem 7.4. It is clear that the running time is a fixed polynomial in the number of samples N, once

$$N \geqslant \text{poly}\left(m, n, p, s, \kappa, 1/w_{\min}, 1/\varepsilon, 1/\gamma, 1/\delta\right).$$

□

We are also able to show that our parameter learning algorithm actually allows us to do nearly Bayes-optimal clustering in the fully observed case i.e. when $C_i = I$ for all $i \in [k]$ [1].

Theorem 7.5 (Bayes-Optimal Clustering)**.** *Let $\mathcal{M} = w_1\mathcal{L}_1 + \cdots + w_k\mathcal{L}_k$ be a mixture of LDS where each $\mathcal{L}_i = \mathcal{L}(A_i, B_i, C_i, D_i)$ with $C_i = I$ and assume that the mixture $\mathcal{M}$ satisfies the assumptions in Section 3. Then given*

$$N = \text{poly}\left(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\delta\right)$$

sample trajectories from this mixture, there is an algorithm that runs in $\text{poly}(N)$ time and has the following guarantees with probability $1-\delta$. There is a fixed permutation π on $[k]$ such that given any trajectory $(u_1, \ldots, u_l, y_1, \ldots, y_l)$ with $l \leqslant O(s)$ and $\|u_i\|, \|y_i\| \leqslant \text{poly}(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\delta)$ it computes a posterior distribution $(p_1, \ldots, p_k)$ on $[k]$ (with $p_1 + \cdots + p_k = 1$) such that $(p_{\pi(1)}, \ldots, p_{\pi(k)})$ is δ-close in TV distance to the posterior distribution on $\mathcal{L}_1, \ldots, \mathcal{L}_k$ from which the trajectory $(u_1, \ldots, u_l, y_1, \ldots, y_l)$ was drawn.

Remark 7.6. Note that the condition that $\|u_i\|, \|y_i\| \leqslant \text{poly}(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\delta)$ is satisfied with exponentially small failure probability for a random trajectory from any of the components since $l \leqslant O(s)$. The trajectories used in the learning algorithm have length $O(s)$ so in particular, we can nearly-optimally cluster those.

[1] We believe that our clustering result naturally generalizes to the partially observed setting as long as assume that all of the $\mathcal{L}_i = \mathcal{L}(A_i, B_i, C_i, D_i)$ are written in their balanced realization (see (Oymak & Ozay, 2019) for a formal definition) which is just a canonical choice of the similarity transformation U_i that is allowed to act on A_i, B_i, C_i

Acknowledgements

AB was supported by Ankur Moitra's ONR grant. AL was supported by an NSF Graduate Research Fellowship and a Fannie and John Hertz Foundation Fellowship. AM was supported by a grant from the ONR, NSF Award 1918656 and a David and Lucile Packard Fellowship.

References

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15:2773–2832, 2014.

Åström, K. J. and Eykhoff, P. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

Bakshi, A., Liu, A., Moitra, A., and Yau, M. A new approach to learning linear dynamical systems. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 335–348, 2023.

Belkin, M. and Sinha, K. Polynomial learning of distribution families. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 103–112. IEEE, 2010.

Bulteel, K., Tuerlinckx, F., Brose, A., and Ceulemans, E. Clustering vector autoregressive models: Capturing qualitative differences in within-person dynamics. *Frontiers in Psychology*, 7:1540, 2016.

Candès, E. J. and Fernandez-Granda, C. Towards a mathematical theory of super-resolution. *Communications on pure and applied Mathematics*, 67(6):906–956, 2014.

Chen, S. and Moitra, A. Algorithmic foundations for the diffraction limit. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 490–503, 2021.

Chen, S., Li, J., and Song, Z. Learning mixtures of linear regressions in subexponential time via fourier moments.

CoRR, abs/1912.07629, 2019. URL http://arxiv.org/abs/1912.07629.

Chen, Y. and Poor, H. V. Learning mixtures of linear dynamical systems. In *International Conference on Machine Learning*, pp. 3507–3557. PMLR, 2022.

Chen, Y., Yi, X., and Caramanis, C. A convex formulation for mixed regression with two components: Minimax optimal rates, 2013. URL https://arxiv.org/abs/1312.7006.

Diamandis, T., Eldar, Y. C., Fallah, A., Farnia, F., and Ozdaglar, A. A wasserstein minimax framework for mixed linear regression, 2021. URL https://arxiv.org/abs/2106.07537.

Ding, F. Two-stage least squares based iterative estimation algorithm for cararma system modeling. *Applied Mathematical Modelling*, 37(7):4798–4808, 2013. ISSN 0307-904X. doi: https://doi.org/10.1016/j.apm.2012.10.014. URL https://www.sciencedirect.com/science/article/pii/S0307904X12006191.

Faradonbeh, M. K. S., Tewari, A., and Michailidis, G. Finite time identification in unstable linear systems. *CoRR*, abs/1710.01852, 2017. URL http://arxiv.org/abs/1710.01852.

Faradonbeh, M. K. S., Tewari, A., and Michailidis, G. Finite time identification in unstable linear systems. *Automatica*, 96:342–353, 2018.

Fox, E., Sudderth, E., Jordan, M., and Willsky, A. Nonparametric bayesian learning of switching linear dynamical systems. *Advances in neural information processing systems*, 21, 2008.

Galrinho, M. Least squares methods for system identification of structured models. 2016.

Gonze, D., Coyte, K. Z., Lahti, L., and Faust, K. Microbial communities as dynamical systems. *Current opinion in microbiology*, 44:41–49, 2018.

Hallac, D., Vare, S., Boyd, S., and Leskovec, J. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 215–223, 2017.

Hardt, M., Ma, T., and Recht, B. Gradient descent learns linear dynamical systems. *Journal of Machine Learning Research*, 19:1–44, 2018.

Hazan, E., Singh, K., and Zhang, C. Learning linear dynamical systems via spectral filtering. *CoRR*, abs/1711.00946, 2017. URL http://arxiv.org/abs/1711.00946.

Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. Spectral filtering for general linear dynamical systems. *Advances in Neural Information Processing Systems*, 31, 2018.

HO, B. and Kálmán, R. E. Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548, 1966.

Hsu, D. and Kakade, S. M. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pp. 11–20, 2013.

Huang, Q. and Kakade, S. M. Super-resolution off the grid. *Advances in Neural Information Processing Systems*, 28, 2015.

Kalai, A. T., Moitra, A., and Valiant, G. Efficiently learning mixtures of two gaussians. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 553–562, 2010.

Kalliovirta, L., Meitz, M., and Saikkonen, P. Gaussian mixture vector autoregression. *Journal of econometrics*, 192(2):485–498, 2016.

Kong, W., Somani, R., Kakade, S., and Oh, S. Robust meta-learning for mixed linear regression with small batches, 2020a. URL https://arxiv.org/abs/2006.09702.

Kong, W., Somani, R., Song, Z., Kakade, S., and Oh, S. Meta-learning for mixed linear regression, 2020b. URL https://arxiv.org/abs/2002.08936.

Kwon, J., Ho, N., and Caramanis, C. On the minimax optimality of the em algorithm for learning two-component mixed linear regression, 2020. URL https://arxiv.org/abs/2006.02601.

Li, W. On a mixture autoregressive model. j royal stat soc ser b. *Journal of the Royal Statistical Society Series B*, 62:95–115, 02 2000. doi: 10.1111/1467-9868.00222.

Li, Y. and Liang, Y. Learning mixtures of linear regressions with nearly optimal complexity. *CoRR*, abs/1802.07895, 2018. URL http://arxiv.org/abs/1802.07895.

Ljung, L. System identification. In *Signal analysis and prediction*, pp. 163–173. Springer, 1998.

Mezer, A., Yovel, Y., Pasternak, O., Gorfine, T., and Assaf, Y. Cluster analysis of resting-state fmri time series. *Neuroimage*, 45(4):1117–1125, 2009.

Moitra, A. Super-resolution, extremal functions and the condition number of vandermonde matrices. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 821–830, 2015.

Moitra, A. *Algorithmic aspects of machine learning*. Cambridge University Press, 2018.

Moitra, A. and Valiant, G. Settling the polynomial learnability of mixtures of gaussians. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 93–102. IEEE, 2010.

Mossel, E. and Roch, S. Learning nonsingular phylogenies and hidden markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pp. 366–375, 2005.

Mudrik, N., Chen, Y., Yezerets, E., Rozell, C. J., and Charles, A. S. Decomposed linear dynamical systems (dlds) for learning the latent components of neural dynamics. *arXiv preprint arXiv:2206.02972*, 2022.

Oymak, S. and Ozay, N. Non-asymptotic identification of lti systems from a single trajectory. In *2019 American control conference (ACC)*, pp. 5655–5661. IEEE, 2019.

Sarkar, T. and Rakhlin, A. Near optimal finite time identification of arbitrary linear dynamical systems. In *International Conference on Machine Learning*, pp. 5610–5618. PMLR, 2019.

Sarkar, T., Rakhlin, A., and Dahleh, M. A. Nonparametric finite time lti system identification. *arXiv preprint arXiv:1902.01848*, 2019.

Shah, P., Bhaskar, B. N., Tang, G., and Recht, B. Linear system identification via atomic norm regularization. In *2012 IEEE 51st IEEE conference on decision and control (CDC)*, pp. 6265–6270. IEEE, 2012.

Simchowitz, M., Mania, H., Tu, S., Jordan, M. I., and Recht, B. Learning without mixing: Towards A sharp analysis of linear system identification. *CoRR*, abs/1802.08334, 2018a. URL http://arxiv.org/abs/1802.08334.

Simchowitz, M., Mania, H., Tu, S., Jordan, M. I., and Recht, B. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pp. 439–473. PMLR, 2018b.

Simchowitz, M., Boczar, R., and Recht, B. Learning linear dynamical systems with semi-parametric least squares. In *Conference on Learning Theory*, pp. 2714–2802. PMLR, 2019.

Spinelli, W., Piroddi, L., and Lovera, M. On the role of prefiltering in nonlinear system identification. *IEEE Transactions on Automatic Control*, 50(10):1597–1602, 2005. doi: 10.1109/TAC.2005.856655.

Tsiamis, A. and Pappas, G. J. Finite sample analysis of stochastic system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 3648–3654. IEEE, 2019.

Yi, X., Caramanis, C., and Sanghavi, S. Alternating minimization for mixed linear regression, 2013. URL https://arxiv.org/abs/1310.3745.

Zhang, Y. Unbiased identification of a class of multi-input single-output systems with correlated disturbances using bias compensation methods. *Mathematical and Computer Modelling*, 53(9):1810–1819, 2011. ISSN 0895-7177. doi: https://doi.org/10.1016/j.mcm.2010.12.059. URL https://www.sciencedirect.com/science/article/pii/S0895717711000045.

A. Appendix

Here we include proofs of the intermediate results that were omitted in the main body. We begin with the following two claims from (Bakshi et al., 2023) that give us bounds on the singular values of $O_{\mathcal{L}_i,s}, Q_{\mathcal{L}_i,s}, A^s$.

Claim A.1 (Claim 5.15 in (Bakshi et al., 2023))**.** Consider a well-behaved mixture of LDS (Definition 3.5) $\mathcal{M} = w_1\mathcal{L}_1 + \cdots + w_k\mathcal{L}_k$ where each $\mathcal{L}_i = \mathcal{L}(A_i, B_i, C_i, D_i)$. Then for all i, $\sigma_{\min}(O_{\mathcal{L}_i,s}) \leqslant \sqrt{s}\kappa$ and $\sigma_{\min}(Q_{\mathcal{L}_i,s}) \leqslant \sqrt{s}\kappa$.

Claim A.2 (Claim 5.16 in (Bakshi et al., 2023))**.** Consider a well-behaved mixture of LDS (Definition 3.5) $\mathcal{M} = w_1\mathcal{L}_1 + \cdots + w_k\mathcal{L}_k$ where each $\mathcal{L}_i = \mathcal{L}(A_i, B_i, C_i, D_i)$. Then for any integer $t > 0$,

$$\|A_i^t\|_F \leqslant (\sqrt{n}\kappa)^{t/s}.$$

A.1. Proof of Fact 5.2

Proof. Invoking the algebraic identity from Fact 5.1, consider the case where $j \neq 0$.

$$\begin{aligned}
\mathbb{E}\left[y_{t+j}u_t^\top\right] &= \mathbb{E}\left[\left(\sum_{i=1}^{t+j}\left(CA^{i-1}Bu_{t+j-i} + CA^{i-1}w_{t+j-i}\right) + CA^{t+j}x_0 + Du_{t+j} + z_{t+j}\right)u_t^\top\right] \\
&= \underbrace{\sum_{i=1}^{j-1} CA^{i-1}B\,\mathbb{E}\left[u_{t+j-i}u_t^\top\right]}_{(4).(1)} + CA^{j-1}B\,\mathbb{E}\left[u_tu_t^\top\right] + \underbrace{\sum_{i=j+1}^{t+j} CA^{i-1}B\,\mathbb{E}\left[u_{t+j-i}u_t^\top\right]}_{(4).(2)} \\
&\quad + \underbrace{\sum_{i=1}^{t+j} CA^{i-1}\,\mathbb{E}\left[w_{t+j-i}u_t^\top\right] + CA^{t+j}\,\mathbb{E}\left[x_0u_t^\top\right] + D\,\mathbb{E}\left[u_{t+j}u_t^\top\right] + \mathbb{E}\left[z_{t+j}u_t^\top\right]}_{(4).(3)} \\
&= CA^{j-1}B
\end{aligned} \tag{4}$$

where the last inequality follows from observing that by independence of u_t's, w_t's , z_t's and x_0, the terms (4).(1), (4).(2) and (4).(3) are 0. Similarly, when $j = 0$, the only non-zero term is $\mathbb{E}\left[Du_tu_t^\top\right] = D$ and the claim follows. □

A.2. Proof of Lemma 5.4

Proof. First, for simplicity consider the case where $k_1 = k_2 = k_3 = 0$. Then,

$$\begin{aligned}
\mathbb{E}\left[y_{t+2} \otimes u_{t+2} \otimes y_{t+1} \otimes u_{t+1} \otimes y_t \otimes u_t\right] &= \mathbb{E}\left[Du_{t+2} \otimes u_{t+2} \otimes Du_{t+1} \otimes u_{t+1} \otimes Du_t \otimes u_t\right] \\
&= \left(D\,\mathbb{E}\left[u_{t+2}u_{t+2}^\top\right]\right) \otimes \left(D\,\mathbb{E}\left[u_{t+1}u_{t+1}^\top\right]\right) \otimes \left(D\,\mathbb{E}\left[u_tu_t^\top\right]\right) \\
&= D \otimes D \otimes D,
\end{aligned} \tag{5}$$

where the second equality follows from u_{t+2}, u_{t+1} and u_t being independent random variables. Next, consider the case where $k_3 = 0$ and $k_2, k_1 > 0$. Observe, $y_{t+k_1+k_2+2} \otimes u_{t+k_1+k_2+2}$ has only one non-zero term in expectation. Therefore, we can split the sum as follows:

$$\begin{aligned}
&\mathbb{E}\left[y_{t+k_1+k_2+2} \otimes u_{t+k_1+k_2+2} \otimes y_{t+k_1+k_2+1} \otimes u_{t+k_1+1} \otimes y_{t+k_1} \otimes u_t\right] \\
&= \mathbb{E}\left[D\left(u_{t+k_1+k_2+2} \otimes u_{t+k_1+k_2+2}\right) \otimes \left(y_{t+k_1+k_2+1} \otimes u_{t+k_1+1}\right) \otimes \left(y_{t+k_1} \otimes u_t\right)\right] \\
&= D\left(\mathbb{E}\left[u_{t+k_1+k_2+2} \otimes u_{t+k_1+k_2+2}\right]\right) \otimes \underbrace{\mathbb{E}\left[\left(y_{t+k_1+k_2+1} \otimes u_{t+k_1+1}\right) \otimes \left(y_{t+k_1} \otimes u_t\right)\right]}_{(6).(1)}
\end{aligned} \tag{6}$$

where the second equality follows from observing that $u_{t+k_1+k_2+2}$ is independent of all the terms appearing in the expansion of $y_{t+k_1+k_2+1}$ and y_{t+k_1}, and the random variables $u_{t+k_1+k_2+2}$ and u_{t+k_1}.

Now, we focus on simplifying term (6).(1). Let $\zeta_t = CA^t x_0 + Du_t + z_t$. Observe that $\mathbb{E}\left[w_t \otimes u_{t'}\right] = \mathbb{E}\left[CA^t x_0 \otimes u_t\right] = \mathbb{E}\left[z_t \otimes u_t\right] = 0$, for all t, t', and $\mathbb{E}\left[u_{t_1} \otimes u_{t_2} \otimes u_{t_3} \otimes u_{t_4}\right] = 0$ for all $t_1 > t_2 > t_3 > t_4$. Further, any permutation of t_1, t_2, t_3 and t_4 is also 0. Plugging in the definition from Fact 5.1, we have

$$
\begin{aligned}
&\mathbb{E}\Big[y_{t+k_1+k_2+1} \otimes u_{t+k_1+1} \otimes y_{t+k_1} \otimes u_t\Big] \\
&= \mathbb{E}\Bigg[\left(\sum_{i=1}^{t+k_1+k_2+1} \left(CA^{i-1}Bu_{t+k_1+k_2+1-i} + CA^{i-1}w_{t+k_1+k_2+1-i}\right) + \zeta_{t+k_1+k_2+1}\right) \otimes u_{t+k_1+1} \\
&\qquad \otimes \left(\sum_{i=1}^{t+k_1} \left(CA^{i-1}Bu_{t+k_1-i} + CA^{i-1}w_{t+k_1-i}\right) + \zeta_{t+k_1}\right) \otimes u_t\Bigg] \\
&= \mathbb{E}\Bigg[\left(\sum_{i=k_2}^{t+k_1+k_2+1} \left(CA^{i-1}Bu_{t+k_1+k_2+1-i}\right)\right) \otimes u_{t+k_1+1} \\
&\qquad \otimes \left(\sum_{i=1}^{t+k_1} \left(CA^{i-1}Bu_{t+k_1-i} + CA^{i-1}w_{t+k_1-i}\right)\right) \otimes u_t\Bigg] \\
&= \underbrace{\mathbb{E}\Big[\left(CA^{k_2-1}Bu_{t+k_1+1}\right) \otimes u_{t+k_1+1}\Big]}_{(7).(1)} \otimes \underbrace{\mathbb{E}\Bigg[\left(\sum_{i=1}^{t+k_1} \left(CA^{i-1}Bu_{t+k_1-i} + CA^{i-1}w_{t+k_1-i}\right)\right) \otimes u_t\Bigg]}_{(7).(2)} \\
&\quad + \underbrace{\mathbb{E}\Bigg[\left(\sum_{i=k_2+1}^{t+k_1+k_2+1} \left(CA^{i-1}Bu_{t+k_1+k_2+1-i}\right)\right) \otimes u_{t+k_1+1} \otimes y_{t+k_1} \otimes u_t\Bigg]}_{(7).(3)}
\end{aligned} \tag{7}
$$

where the second equality follows from observing that $\mathbb{E}\left[w_{t+k_1k_2+1-i} \otimes u_{t+k_1+1} \otimes w_{t+k_1-j} \otimes u_t\right] = 0$ for all $i \in [1, t+k_1+k_2+1]$ and $j \in [1, t+k_1]$. Similarly, $\mathbb{E}\left[\zeta_{t+k_1+k_2+1} \otimes u_{t+k_1+1} \otimes y_{t+k_1} \otimes u_t\right] = 0$. Further, for all $i \in [1, k_2 - 1]$, $\mathbb{E}\left[u_{t+k_1+k_2+1-i} \otimes u_{t+k_1+1} \otimes y_t \otimes u_t\right] = 0$. The third equality follows from observing that u_{t+k_1+1} is independent of $y_t \otimes u_t$. Next, observe

$$
(7).(1) = CA^{k_2-1}B, \tag{8}
$$

since $\mathbb{E}\left[u_{t+k_1+1} \otimes u_{t+k_1+1}\right] = I$. Using a similar argument, we observe that all the terms in (7).(2) are zero in expectation apart from the one corresponding to $CA^{k_1-1}B$. Therefore,

$$
(7).(2) = CA^{k_1-1}B. \tag{9}
$$

Next, recall that $\mathbb{E}\left[u_t\right] = 0$ for all t, and since u_{t+k_1+1} is independent of all $u_{t'}$ where $t' < t + k_1 + 1$,

$$
(7).(3) = 0. \tag{10}
$$

Similarly, when $k_1 = 0$, $(7).(1) = D$ and when $k_2 = 0$, $(7).(2) = D$.

Therefore, combining equations (8),(9) and (10), and plugging them back into equation (6), we have

$$
\mathbb{E}\left[y_{t+k_1+k_2+2} \otimes u_{t+k_1+k_2+2} \otimes y_{t+k_1+k_2+1} \otimes u_{t+k_1+1} \otimes y_{t+k_1} \otimes u_t\right] = D \otimes X_{\mathcal{L},k_2} \otimes X_{\mathcal{L},k_1} \tag{11}
$$

It remains to consider the case where $k_3 > 0$. We can now simply repeat the above argument and observe that instead of picking up the term $Du_{t+k_1+k_2+k_3+2}$ from the expansion of $y_{t+k_1+k_2+k_3+2}$, we now pick up the term $CA^{k_3-1}Bu_{t+k_1+k_2+k_3+2}$. This concludes the proof. □

A.3. Proof of Theorem 7.2

We first show that the empirical 6-th moment tensor is close to the true tensor in Frobenius norm.

Lemma A.3 (Empirical Concentration of the 6-th Moment)**.** *Given $\varepsilon, \delta > 0$ and $N \geqslant N_0$ length $6s$ trajectories from a mixture of linear dynamical systems $\mathcal{M} = \sum_{i\in[k]} w_i \mathcal{L}(A_i, B_i, C_i, D_i)$, if*

$$N_0 \geqslant \mathrm{poly}(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\varepsilon, 1/\delta),$$

with probability at least $1 - \delta$ Algorithm 2 outputs a tensor $\widehat{\Pi}_{\mathcal{M}}$ such that

$$\left\|\widehat{\Pi}_{\mathcal{M}} - \Pi_{\mathcal{M}}\right\|_F \leqslant \varepsilon,$$

where $\Pi_{\mathcal{M}} = \sum_{i\in[k]} w_i G_{\mathcal{L}_i,2s} \otimes G_{\mathcal{L}_i,2s} \otimes G_{\mathcal{L}_i,2s}$.

Proof. Note that the joint distribution of $(u_1^i, \ldots, u_l^i, y_1^i, \ldots, y_l^i)$ is Gaussian. Furthermore, the covariance of this Gaussian has entries bounded by $\mathrm{poly}(m, n, p, s, \kappa)$. Thus, by standard concentration inequalities, the empirical sixth moment tensor concentrates around its mean with high probability. Since $\Pi_{\mathcal{M}}, \widehat{\Pi}_{\mathcal{M}}$ are obtained by taking a linear transformation of the sixth moment tensor and the coefficients of this transformation are also bounded by $\mathrm{poly}(m, n, p, s, \kappa)$, we are done.

□

Next, we show that running Jennrich's algorithm on an appropriate flattening of the tensor $\widehat{P}_{\mathcal{M}}$ recovers an estimate of the Markov parameters of each individual component of the mixture.

Lemma A.4 (Markov Parameters via Tensor Decomposition)**.** *Given $\varepsilon, \delta > 0$ and $N \geqslant N_0$ length $6s$ trajectories from a mixture of linear dynamical systems $\mathcal{M} = \sum_{i\in[k]} w_i \mathcal{L}(A_i, B_i, C_i, D_i)$, if*

$$N_0 \geqslant \mathrm{poly}(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\varepsilon, 1/\delta),$$

with probability at least $1 - \delta$, Jennrich's algorithm (Algorithm 4) outputs tensors $\widehat{T}_1, \widehat{T}_2, \ldots, \widehat{T}_k$ such that there is some permutation π on $[k]$ such that for all $i \in [k]$,

$$\left\|\widehat{T}_{\pi(i)} - w_i \cdot v(G_{\mathcal{L}_i,2s}) \otimes v(G_{\mathcal{L}_i,2s}) \otimes v(G_{\mathcal{L}_i,2s})\right\|_F \leqslant \varepsilon,$$

where $v(G_{\mathcal{L}_i,2s})$ denotes flattening the matrix $G_{\mathcal{L}_i,2s}$ into a $mp(2s+1)$-dimensional vector.

Proof. Let K be the matrix whose columns are $v(G_{\mathcal{L}_i,2s})$. By the joint nondegeneracy assumption, $\sigma_k(K) \geqslant \gamma$. On the other hand, we know that

$$\|K\|_F \leqslant \mathrm{poly}(k, m, p, n, s, \kappa)$$

so we can apply Theorem A.5 and Lemma A.3 (with ε rescaled appropriately by a polynomial in the other parameters) to get the desired bound. □

Now we can complete the proof of Theorem 7.2.

Proof of Theorem 7.2. Lemma A.4 implies that

$$\|\widehat{v}_i - w_i \cdot v(G_{\mathcal{L}_i,2s})\|v(G_{\mathcal{L}_i,2s}\|^2\| \leqslant \varepsilon\,.$$

This also implies that

$$\left|\|\widehat{v}_i\| - w_i\|v(G_{\mathcal{L}_i,2s}\|^3\right| \leqslant \varepsilon\,.$$

Also note that we must have

$$1 \leqslant \|v(G_{\mathcal{L}_i,2s})\| \leqslant \mathrm{poly}(k, m, n, p, s, \kappa)\,.$$

Thus

$$\|\widehat{v}_i/\|\widehat{v}_i\|^{2/3} - w_i^{1/3} v(G_{\mathcal{L}_i,2s})\| \leqslant \varepsilon \cdot \mathrm{poly}(k, m, n, p, s, \kappa)\,.$$

We now get the desired bound by simply rescaling the setting of ε in Lemma A.4 by a polynomial in the other parameters. □

Algorithm 4 Jennrich's Algorithm

Input: Tensor $T' \in \mathbb{R}^{n \times n \times n}$ where

$$T' = T + E$$

for some rank-r tensor T and error E

Operation:

1. Choose unit vectors $a, b \in \mathbb{R}^n$ uniformly at random
2. Let $T^{(a)}, T^{(b)}$ be $n \times n$ matrices defined as

$$T^{(a)}_{ij} = T'_{i,j,\cdot} \cdot a$$
$$T^{(b)}_{ij} = T'_{i,j,\cdot} \cdot b$$

3. Let $T^{(a)}_r, T^{(b)}_r$ be obtained by taking the top r principal components of $T^{(a)}, T^{(b)}$ respectively.
4. Compute the eigendecompositions of $U = T^{(a)}_r (T^{(b)}_r)^\dagger$ and $V = \left((T^{(a)}_r)^\dagger T^{(b)}_r\right)^\top$
5. Let $u_1, \ldots, u_r, v_1, \ldots, v_r$ be the eigenvectors computed in the previous step.
6. Permute the v_i so that for each pair (u_i, v_i), the corresponding eigenvalues are (approximately) reciprocals.
7. Solve the following for the vectors w_i

$$\arg\min \|T' - \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i\|_2^2$$

Output: the rank-1 components $\{u_i \otimes v_i \otimes w_i\}_{i=1}^r$

A.4. Jennrich's Algorithm

Jennrich's Algorithm is an algorithm for decomposing a tensor, say $T = \sum_{i=1}^r (x_i \otimes y_i \otimes z_i)$, into its rank-1 components that works when the fibers of the rank 1 components i.e. $x_1, \ldots, x_r$ are linearly independent (and similar for $y_1, \ldots, y_r$ and $z_1, \ldots, z_r$).

Moitra (Moitra, 2018) gives a complete analysis of JENNRICH'S ALGORITHM. The result that we need is that as the error E goes to 0 at an inverse-polynomial rate, JENNRICH'S ALGORITHM recovers the individual rank-1 components to within any desired inverse-polynomial accuracy.

Theorem A.5 ((Moitra, 2018)). *Let*

$$T = \sum_{i=1}^{r} \sigma_i (x_i \otimes y_i \otimes z_i)$$

where the x_i, y_i, z_i are unit vectors and $\sigma_1 \geqslant \ldots \geqslant \sigma_r > 0$. Assume that the smallest singular value of the matrix with columns given by $x_1, \ldots, x_r$ is at least c and similar for the y_i and z_i. Then for any constant d, there exists a polynomial P such that if

$$\|E\|_2 \leqslant \frac{\sigma_1}{P(n, \frac{1}{c}, \frac{\sigma_1}{\sigma_r})}$$

then with $1 - \frac{1}{(10n)^d}$ probability, there is a permutation π such that the outputs of JENNRICH'S ALGORITHM *satisfy*

$$\|\sigma_{\pi(i)}(x_{\pi(i)} \otimes y_{\pi(i)} \otimes z_{\pi(i)}) - u_i \otimes v_i \otimes w_i\|_2 \leqslant \sigma_1 \left(\frac{\sigma_r c}{10 \sigma_1 n}\right)^d$$

for all $1 \leqslant i \leqslant r$.

Remark A.6. Note that the extra factors of σ_1 in the theorem above are simply to deal with the scaling of the tensor T.

A.5. Bayes Optimal Clustering

In this section, we prove that our parameter learning algorithm actually allows us to do nearly Bayes-optimal clustering in the fully observed case i.e. when $C_i = I$ for all $i \in [k]$

Proof of Theorem 7.5. We apply Theorem 7.1 with ε set as a sufficiently small inverse polynomial in the other parameters. Because $C_i = I$, we can eliminate the similarity transformations U_i and also without loss of generality the permutation π on $[k]$ is the identity so we have

$$\max_{i\in[k]} \left(\|A_i - \widehat{A}_i\|, \|B_i - \widehat{B}_i\|, \|C_i - \widehat{C}_i\|, \|D_i - \widehat{D}_i\|, |w_i - \widehat{w}_i| \right) \leqslant \varepsilon \,.$$

Now fix a choice of $i \in [k]$. Define $\mathcal{P}_i$ to be the probability that $(u_1, \ldots u_l, y_1, \ldots, y_l)$ is sampled from the LDS $\mathcal{L}(A_i, B_i, C_i, D_i)$ and let $\widehat{\mathcal{P}}_i$ be the probability that it is sampled from the LDS $\widehat{\mathcal{L}}_i = \mathcal{L}(\widehat{A}_i, \widehat{B}_i, \widehat{C}_i, \widehat{D}_i)$. We can explicitly compute $\widehat{\mathcal{P}}_i$ from $\widehat{A}_i, \widehat{B}_i, \widehat{C}_i, \widehat{D}_i$ using regression. Now we will bound the ratio $\mathcal{P}_i/\widehat{\mathcal{P}}_i$ and prove that it is close to 1. We can write $\mathcal{P}_i$ as an integral over all possibilities for $x_1, \ldots, x_l$. Now the likelihood of $(u_1, \ldots u_l, y_1, \ldots, y_l, x_1, \ldots, x_l)$ is simply

$$C \exp\left(-\frac{1}{2} \left(\sum_{t=1}^{l-1} \|x_{t+1} - A_i x_t - B_i u_t\|^2 + \sum_{t=1}^{l} \|y_t - C_i x_t - D_i u_t\|^2 + \sum_{t=1}^{l} \|u_t\|^2 \right) \right)$$

where C is an appropriate normalizing constant obtained from the standard normal. The formula is the same for $\widehat{\mathcal{P}}_i$ except with A_i, B_i, C_i, D_i replaced with $\widehat{A}_i, \widehat{B}_i, \widehat{C}_i, \widehat{D}_i$. As long as

$$\|x_1\|, \ldots, \|x_l\| \leqslant \mathrm{poly}(m, n, p, s, \kappa, 1/w_{\min}, 1/\gamma, 1/\delta)$$

then the ratio between the two likelihoods is in the interval $[1 - \sqrt{\varepsilon}, 1 + \sqrt{\varepsilon}]$ as long as ε was chosen sufficiently small initially. However, the above happens with exponentially small failure probability for both $\mathcal{L}_i$ and $\widehat{\mathcal{L}}_i$ so we actually have

$$1 - 2\sqrt{\varepsilon} \leqslant \frac{\widehat{\mathcal{P}}_i}{\mathcal{P}_i} \leqslant 1 + 2\sqrt{\varepsilon} \,.$$

Combining the above over all $i \in [k]$ immediately implies the desired statement about the posterior distribution. □