PAC-Bayesian Meta-Learning for Few-Shot Identification of Linear Dynamical Systems

Anonymous authors
Paper under double-blind review

Abstract

Identifying linear time-invariant (LTI) dynamical systems from data is especially challenging when trajectories are short, noisy, or high-dimensional. Traditional system identification methods typically treat each system in isolation and therefore discard shared information that may exist across related systems. We propose a PAC-Bayesian Meta-Learning framework for LTI system identification (PBML-LTI) that explicitly leverages cross-task structure while preserving task-level heterogeneity. Each task corresponds to an unknown LTI system, and a meta-learner uses a collection of training trajectories to learn a data-dependent prior over system parameters. Given a new system with limited trajectory data, the method performs Bayesian inference to produce a posterior distribution over the new system's parameters, enabling calibrated uncertainty quantification and principled adaptation in the few-shot regime.

A key technical challenge is temporal dependence: trajectories generated by LTI systems violate i.i.d. assumptions underlying standard learning theory. To address this, we develop generalization guarantees for meta-learned priors under sequential dependence using martingale-based PAC-Bayes analysis with sub-normalized concentration tools. The resulting bounds characterize how the quality of the learned prior controls expected identification error on unseen systems, with explicit dependence on trajectory length, noise, and the divergence between task posteriors and the meta-prior. This connects uncertainty-aware meta-identification with finite-sample theory for dependent dynamical data.

1 Introduction

Linear time-invariant (LTI) dynamical systems are the cornerstone of control theory and time-series analysis. The fundamental challenge in this domain is recovering the system's transition operator from finite observations. While the identification task can be cast as linear regression, it is complicated by the data's intrinsic autoregressive structure. The covariates are generated by the underlying dynamics, creating temporal dependencies that violate standard i.i.d. assumptions. Recent non-asymptotic results have made significant strides in characterizing these challenges, establishing error bounds based on trajectory length and system stability (Faradonbeh et al., 2018; Simchowitz et al., 2018; Sarkar & Rakhlin, 2019).

In many modern applications, however, identification is not performed once for a single system but repeatedly across a family of related systems, such as different individuals, devices, environments, operating modes, or geographic regions. As emphasized by the joint learning perspective, such multi-system settings are widespread and often exhibit shared latent structures that can be exploited to improve data efficiency when individual trajectories are short or high-dimensional. Existing joint and multi-task identification methods leverage this structure by pooling trajectories across systems, typically through shared-basis or representation assumptions, in order to improve estimation accuracy on the observed collection of systems (Lin et al., 2024; Modi et al., 2024; Chen et al., 2023).

Yet, many downstream objectives require a distinct capability: rapid adaptation to previously unseen systems using only a small amount of new data. This challenge arises when deploying dynamical systems/models across populations of related, yet *heterogeneous* entities. In such settings, each system is governed by unique

local dynamics, necessitating adaptation within a brief observational window. For example, one may need to tune a controller for a new aircraft operating condition or configuration (Bosworth, 1992), or calibrate a forecasting model for a newly observed region or economic unit in financial monitoring and forecasting settings (Pesaran, 2015; Stock & Watson, 2016).

In these settings, the primary challenge shifts from accurately estimating the dynamics of observed systems to adapting quickly and reliably to novel systems under strict data constraints. Acquiring long trajectories is often precluded by high costs, safety concerns, or environmental nonstationarity (García & Fernández, 2015; Ji et al., 2023). Similar limitations plague domains like physiological monitoring, where only short data segments are practically available (Ludvig & Perreault, 2012). This necessitates a shift in strategy: rather than merely pooling data to minimize estimation error on the training set, we must extract transferable structural knowledge that supports principled few-shot adaptation and uncertainty quantification on unseen systems.

To this end, we propose a Bayesian meta-learning framework for identifying linear dynamical systems. We treat each system as a task sampled from a shared environment, using trajectories from multiple training tasks to learn a data-dependent prior over system dynamics.

When a new system is encountered, this learned prior enables rapid adaptation from sparse observations via Bayesian inference, yielding both a point estimate and a calibrated measure of uncertainty. Unlike joint learning approaches that primarily leverage shared structure to refine estimates within the training cohort, our meta-learning objective explicitly targets the learning of an *inductive bias* optimized for the few-shot identification of future systems (Finn et al., 2017; Patacchiola et al., 2020; Muthirayan et al., 2025).

This formulation invites a critical question: how can we characterize generalization to new systems when adaptation is based on limited data? PAC-Bayes theory offers a rigorous framework for addressing this issue, as it explicitly bounds generalization error using a trade-off between empirical fit and a complexity penalty measured by the Kullback–Leibler divergence from a prior distribution(McAllester, 1999; Seeger, 2002; Catoni, 2007). This structure renders PAC-Bayes uniquely suited for meta-learning, where the 'quality' of the learned prior directly dictates the speed and reliability of adaptation to new tasks (Liu et al., 2021; Rezazadeh, 2022; Guan & Lu, 2022; Farid & Majumdar, 2021).

However, most existing PAC-Bayes meqt-learning results rely on the assumption of independent observations within each task (Liu et al., 2021; Rezazadeh, 2022; Guan & Lu, 2022; Farid & Majumdar, 2021), whereas system identification typically provides temporally dependent trajectories per task. To bridge this gap, we develop PAC-Bayes guarantees tailored to dynamical data that remain valid under temporal dependence by leveraging martingale-based tools. These guarantees capture the key factors that govern few-shot identification performance in regimes where classical i.i.d. analyses do not apply.

Our contributions can be summarized as follows:

- We propose PBML-LTI, a PAC-Bayesian meta-learning framework for few-shot identification of linear time-invariant dynamical systems that learns a transferable prior distribution and enables closed-form task adaptation via conjugate Bayesian inference.
- We derive martingale PAC-Bayes generalization bounds that explicitly handle temporally dependent trajectory data, extending meta-learning theory beyond the standard i.i.d. setting.
- We illustrate through synthetic and real fMRI-based experiments that PBML-LTI achieves improved data efficiency and robust multi-step prediction in high-dimensional, few-shot regimes.

2 Related Work

Finite-sample identification of LTI systems from dependent trajectories. Classical system identification typically treats dynamical systems in isolation, focusing on estimation from a single trajectory where covariates are endogenously generated and thus temporally dependent. A rich body of non-asymptotic analysis has established sharp finite-time guarantees across stable, marginally stable, and unstable regimes,

characterizing how sample complexity scales with the time horizon, system dimension, and spectral properties (Faradonbeh et al., 2018; Simchowitz et al., 2018; Sarkar & Rakhlin, 2019). These foundational results underscore two critical constraints that our framework must address: the necessity of rigorously handling intra-task sequential dependence, and the dominant role of stability in determining estimation error when data are scarce.

Joint and multi-task identification across related systems. Recent advances address the identification of multiple LTI systems by exploiting shared structural assumptions, such as common bases, low-dimensional subspaces, or coupled optimization objectives (Modi et al., 2024; Chen et al., 2023). These joint-learning formulations excel at improving estimation accuracy across a fixed collection of observed systems by "borrowing strength" between tasks, aligning with classical multi-task learning paradigms (Argyriou et al., 2008). In contrast, our framework targets few-shot generalization to previously *unseen* systems. Here, the goal is not merely to refine estimates for the training set, but to leverage the learned structure to rapidly and reliably calibrate a new system from a short trajectory at deployment time.

Meta-learning and hierarchical Bayes. Meta-learning formalizes knowledge transfer by explicitly optimizing for rapid adaptation to new tasks drawn from a *shared* environment. While gradient-based methods like MAML target performance after a task-specific descent step (Finn et al., 2017), a complementary perspective interprets meta-learning as hierarchical Bayesian inference: training tasks are used to learn a data-driven prior distribution, and test-time adaptation corresponds to computing the posterior distribution (Grant et al., 2018). This Bayesian formulation has been successful for few-shot uncertainty quantification in supervised settings (Patacchiola et al., 2020) and recently applied to accelerate online control in dynamical systems (Muthirayan et al., 2025). Our approach builds on this hierarchical view, but addresses the specific challenges of LTI identification: we handle temporally dependent trajectory data and leverage conjugate priors to enable exact, closed-form posterior updates—avoiding the computational overhead of iterative gradient steps (Dawid, 1981; Gupta & Nagar, 2018; West & Harrison, 1997).

PAC-Bayes generalization and meta-learning. PAC-Bayes theory bounds generalization error by balancing empirical fit against a complexity term measured by the KL divergence to a reference prior (McAllester, 1999; Seeger, 2002; Catoni, 2007). This framework is naturally suited to meta-learning, as the quality of the learned prior distribution directly dictates the bounds on adaptation performance (Liu et al., 2021; Rezazadeh, 2022; Guan & Lu, 2022; Farid & Majumdar, 2021). However, as previously mentioned, existing PAC-Bayes meta-learning bounds predominantly assume independent observations within each task. In contrast, system identification involves trajectories with inherent temporal correlations. We bridge this theoretical gap by integrating PAC-Bayes with martingale concentration inequalities and self-normalized analysis techniques standard in sequential learning (Abbasi-Yadkori et al., 2011; 2012; Faradonbeh et al., 2018; Howard et al., 2020), thereby extending certification to the non-i.i.d. dynamical setting.

Joint learning versus meta-learning in the experimental evaluation. While both joint learning and meta-learning evaluate performance on unseen tasks, they differ fundamentally in their training objectives and adaptation mechanisms. Joint learning typically extracts a static shared structure (e.g., a common subspace) from training tasks, which is then applied "frozen" or with minimal adjustment to new data (Modi et al., 2024; Chen et al., 2023). In contrast, our experimental protocol *explicitly targets few-shot adaptation.* We adopt a rigorous support-query split: for every test task, the model receives only a short "support" prefix to calibrate a task-specific posterior distribution, while evaluation is conducted on a disjoint "query" window. This distinction is crucial; unlike joint learning, which optimizes aggregate performance over a cohort, our meta-learning objective is specifically designed to maximize post-adaptation accuracy, leveraging the learned prior to recover stable dynamics from minimal data.

3 Problem Formulation

We consider the problem of identifying a family of related LTI dynamical systems. The available data comprise trajectories collected from multiple tasks, indexed by $m \in [M] := \{1, 2, \ldots, M\}$. For each task m

and time $t = 0, 1, \ldots, T_m - 1$, the system evolves according to

$$x_{m,t+1} = A_m x_{m,t} + w_{m,t+1}, \tag{1}$$

where $x_{m,t} \in \mathbb{R}^d$ denotes the system state, $A_m \in \mathbb{R}^{d \times d}$ is an unknown transition matrix, and $\{w_{m,t}\}_{t \geq 1}$ is a stochastic noise process. We assume the full state trajectory $\{x_{m,t}\}_{t=0}^{T_m}$ is observed for each task.

It is convenient to rewrite equation 1 in regression form. To this end, define the per-task data matrices

$$X_m := \left[x_{m,0}, x_{m,1}, \ldots, x_{m,T_m-1}\right] \in \mathbb{R}^{d \times T_m}, \qquad Y_m := \left[x_{m,1}, x_{m,2}, \ldots, x_{m,T_m}\right] \in \mathbb{R}^{d \times T_m}, \tag{2}$$

where X_m collects the state vectors serving as predictors at each time step, and Y_m collects the responses, corresponding to the next-step-ahead states. Let $W_m := [w_{m,1}, \ldots, w_{m,T_m}] \in \mathbb{R}^{d \times T_m}$ denote the noise matrix. Then, the trajectory can be written compactly as

$$Y_m = A_m X_m + W_m. \tag{3}$$

A critical distinction from standard linear regression is that the columns of X_m exhibit inherent temporal dependence and are adapted to the history of the noise process.

To complete the model formulation, we impose the following assumptions to ensure that the model is well posed and to clearly specify the relationships among the members of the family of LTIs in equation 1.

Assumption 1 (**Controlled Growth**). *We assume the dynamics of each system exhibit controlled growth over the observed horizon. Specifically, there exists a constant $c_\rho > 0$ such that for all tasks m,*

$$\rho(A_m) \leq 1 + c_\rho / T_m, \tag{4}$$

where $\rho(A_m)$ denotes the spectral radius of A_m. This type of controlled-growth condition allows near-marginal stability while preventing rapid explosion over the finite horizon, which is standard in non-asymptotic LTI identification and is also adopted in single and multi-system learning settings (Faradonbeh et al., 2018; Modi et al., 2024; Simchowitz et al., 2018; Sarkar & Rakhlin, 2019).

Assumption 2 (**Martingale Difference Noise Process**). *Let $\{\mathcal{F}_{m,t}\}_{t \geq 0}$ denote the natural filtration for task m, capturing the history of information available up to time t. Formally, we define $\mathcal{F}_{m,t} := \sigma(x_{m,0}, w_{m,1}, \ldots, w_{m,t})$, which includes the initial state and the sequence of noise realizations driving the trajectory. We assume the noise process forms a martingale difference sequence with respect to this filtration and is conditionally sub-Gaussian. That is,*

$$\mathbb{E}[w_{m,t+1} \mid \mathcal{F}_{m,t}] = 0, \tag{5}$$

and for all $u \in \mathbb{R}^d$,

$$\mathbb{E}\left[\exp(\langle u, w_{m,t+1} \rangle) \mid \mathcal{F}_{m,t}\right] \leq \exp\left(\frac{\sigma_w^2}{2} \|u\|_2^2\right). \tag{6}$$

Additionally, we assume the conditional covariance is uniformly bounded by a fixed positive semidefinite matrix, satisfying $\mathbb{E}[w_{m,t+1} w_{m,t+1}^\top \mid \mathcal{F}_{m,t}] \preceq \Sigma_w$. These conditions place our analysis within the self-normalized martingale framework, a standard setting for analyzing sequential least squares and LTI identification from single trajectories (Abbasi-Yadkori et al., 2011; Simchowitz et al., 2018; Sarkar & Rakhlin, 2019).

Assumption 3 (**Shared Environment**). *We posit a hierarchical generative process where tasks are related via a shared latent structure. Specifically, there exists an unknown hyper-parameter $\phi_\star$ governing the distribution of system dynamics, such that each task parameter A_m is drawn independently:*

$$A_m \mid \phi_\star \overset{\text{i.i.d.}}{\sim} p(\cdot \mid \phi_\star), \qquad m = 1, \ldots, M. \tag{7}$$

Conditional on A_m, the observed trajectories are generated according to the LTI dynamics in equation 1. This assumption of i.i.d. task sampling from a fixed meta-distribution is standard in learning-to-learn theory and serves as the foundation for establishing generalization guarantees to novel tasks (Baxter, 2000; Maurer, 2005; Finn et al., 2017; Liu et al., 2021).

4 PAC-Bayesian Meta-Learning for LTI Identification (PBML-LTI)

In this section, we introduce the proposed PBML-LTI framework, progressing from a high-level intuition to its rigorous mathematical formulation. We build directly upon the setup established in Section 3. Assumption 3 justifies modeling the task parameters $\{A_m\}$ via a shared environment distribution, while Assumptions 1 and 2 serve as the necessary regularity conditions to derive generalization guarantees.

4.1 Hierarchical Bayesian Model And Task Adaptation

We treat each LTI system as a distinct task characterized by a transition matrix A_m. Our objective is to learn shared meta-parameters ϕ that define a transferable prior distribution over the family of matrices $\{A_m\}$. When a new task with limited data is introduced, adaptation is performed via Bayesian inference under this learned prior, yielding both a refined point estimate and a principled measure of uncertainty (Grant et al., 2018).

Hierarchical Bayes interpretation of meta-learning. Our formulation is framed as a hierarchical Bayesian model spanning three levels:

$$\phi \sim p(\phi), \qquad A_m \mid \phi \sim p(A \mid \phi), \qquad D_m \mid A_m \sim p(D \mid A_m), \tag{8}$$

where ϕ encodes the environment-level meta structure shared across tasks, A_m defines the task-specific dynamics, and D_m denotes the observed trajectory data for task m. In classical hierarchical Bayesian modeling, one typically places a hyper-prior distribution $p(\phi)$ and performs joint Bayesian inference over both $\{A_m\}$ and ϕ. In contrast, we adopt an empirical-Bayes perspective: ϕ is treated as a learnable parameter optimized across tasks, while adaptation for a specific task consists of exact Bayesian updates for A_m conditional on the learned ϕ. This perspective is computationally advantageous, as it avoids the overhead of full posterior sampling over ϕ while still leveraging the statistical power of the shared prior distribution (Grant et al., 2018).

Prior Distribution And Working Likelihood. Consistent with Assumption 3, we posit a hierarchical model in which each task parameter is drawn from a shared prior family P_ϕ. We instantiate a matrix-normal prior (Dawid, 1981; Gupta & Nagar, 2018; West & Harrison, 1997)

$$A_m \mid \phi \;\sim\; \mathcal{MN}(W,\, I_d,\, V)\,, \qquad \phi := (W, V, \sigma^2), \tag{9}$$

where $W \in \mathbb{R}^{d\times d}$ is the shared mean, and $V \in \mathbb{R}^{d\times d}$ is a symmetric positive definite matrix. Equivalently, $\mathrm{vec}(A_m) \sim \mathcal{N}(\mathrm{vec}(W),\, V \otimes I_d)$, so that V controls how the entries of A_m co-vary across columns. The use of a matrix-normal prior is particularly well suited to modeling related LTI systems, as it provides a mathematically elegant and computationally efficient mechanism for coupling tasks through a shared mean and covariance structure, while significantly reducing the number of parameters to be learned compared with unstructured priors.

For tractable Bayesian updates, we use a conditional Gaussian working likelihood (West & Harrison, 1997)

$$x_{m,t+1} \mid x_{m,t}, A_m \;\sim\; \mathcal{N}\big(A_m x_{m,t},\, \sigma^2 I_d\big)\,, \qquad t = 0, \ldots, T_m - 1. \tag{10}$$

Closed-Form Task Adaptation. Given task data X_m, Y_m from the matrix regression model equation 3, pairing the Gaussian working likelihood equation 10 with the matrix-normal prior equation 9 yields a conjugate matrix-normal posterior (Dawid, 1981; Gupta & Nagar, 2018; West & Harrison, 1997)

$$Q_{m,\phi} = A_m \mid D_m, \phi \;\sim\; \mathcal{MN}(M_m,\, I_d,\, V_m)\,. \tag{11}$$

The posterior column covariance and mean admit closed-form expressions.

$$V_m = \left(V^{-1} + \frac{1}{\sigma^2} X_m X_m^\top\right)^{-1}, \tag{12}$$

$$M_m = \left(\frac{1}{\sigma^2} Y_m X_m^\top + W V^{-1}\right) V_m. \tag{13}$$

We use the posterior mean $\widehat{A}_m := M_m$ as a point estimate for identification, and we use V_m to quantify parameter uncertainty.

While closed-form adaptation defines the mechanism for updating a prior distribution P_ϕ into a task-specific posterior $Q_{m,\phi}$ from trajectories, the central challenge remains: how to optimally learn the hyperparameter ϕ from many training tasks to ensure generalization to unseen systems. To achieve this in the presence of inherent temporal correlations, we leverage a martingale PAC-Bayes bound explicitly designed to handle trajectory dependence.

4.2 Martingale PAC-Bayes Motivation For Meta-Training

Within each task, observations form a single trajectory and are therefore not i.i.d. Classical PAC-Bayes bounds for independent samples do not apply directly since the instantaneous losses depend on the past through the state. A standard way to handle this dependence is to work with conditional risks and exploit the martingale structure induced by the filtration $\mathcal{F}_t$ in Assumption 2.

For the remainder of this subsection, we fix an arbitrary task m and suppress the task index to simplify notation. Specifically, we write $x_t := x_{m,t}$ for the system state at time t, and $T := T_m$ for the length of the observed trajectory in terms of transitions. Thus, the sequence of states $x_0, \dots, x_T$ induces T one-step prediction losses.

We state three results that lead to a bound-inspired objective.

Theorem 1 (**Instantaneous Gaussian Log-Loss**). [1] *Fix $\sigma^2 > 0$ and consider the Gaussian conditional likelihood model $p_\sigma(x_t \mid x_{t-1}, A) = \mathcal{N}(Ax_{t-1}, \sigma^2 I_d)$. The associated instantaneous log-loss is*

$$\ell_t(A) := -\log p_\sigma(x_t \mid x_{t-1}, A) = \frac{1}{2\sigma^2}\|x_t - Ax_{t-1}\|_2^2 + \frac{d}{2}\log(2\pi\sigma^2), \qquad t = 1, \dots, T. \tag{14}$$

Let $\{\mathcal{F}_t\}_{t=0}^T$ be any filtration such that $\ell_t(A)$ is $\mathcal{F}_t$-measurable, and define the centered increment

$$d_t(A) := \mathbb{E}[\ell_t(A) \mid \mathcal{F}_{t-1}] - \ell_t(A). \tag{15}$$

Then for every fixed A, $\{d_t(A)\}_{t=1}^T$ is a martingale difference sequence: $\mathbb{E}[d_t(A) \mid \mathcal{F}_{t-1}] = 0$ for all t.

Theorem 2 (**Exponential Supermartingale**). [2] *Suppose we have a bound on the conditional moment generating function (MGF) such that for all $\lambda \in (0, 1]$ and all $t = 1, \dots, T$,*

$$\mathbb{E}\big[\exp\big(\lambda d_t(A)\big) \mid \mathcal{F}_{t-1}\big] \le \exp\big(\psi_t(\lambda)\big). \tag{16}$$

Let $S_t(A) := \sum_{i=1}^t d_i(A)$ and define

$$Z_t(A; \lambda) := \exp\left(\lambda S_t(A) - \sum_{i=1}^t \psi_i(\lambda)\right), \qquad t = 0, 1, \dots, T, \tag{17}$$

with $Z_0(A; \lambda) = 1$. Then $\{Z_t(A; \lambda)\}_{t=0}^T$ is a nonnegative supermartingale with respect to $\{\mathcal{F}_t\}_{t=0}^T$.

We also suppress the task index and denote it by $\ell_t(\cdot) := \ell_{m,t}(\cdot)$ and $\mathcal{F}_t := \mathcal{F}_{m,t}$.

Theorem 3 (**Martingale PAC-Bayes Bound**). [3] *Let P be any prior distribution over A independent of the trajectory, and let Q be any posterior distribution over A. Under the setting of Theorems 1 and 2, for any confidence $\delta \in (0, 1)$ and any $\lambda \in (0, 1]$, with probability at least $1 - \delta$,*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}_{A\sim Q}[\mathbb{E}[\ell_t(A) \mid \mathcal{F}_{t-1}]] \le \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{A\sim Q}[\ell_t(A)] + \frac{1}{\lambda T}\Big(D_{\mathrm{KL}}(Q\|P) + \log\tfrac{1}{\delta} + \sum_{t=1}^T \psi_t(\lambda)\Big). \tag{18}$$

[1] The proof is included in Appendix A.1

[2] The proof is included in Appendix A.2.

[3] The proof is included in Appendix A.3.

Theorem 1 shows that the gap between the predictive loss $\mathbb{E}[\ell_t(A) \mid \mathcal{F}_{t-1}]$ and the realized loss $\ell_t(A)$ forms a martingale difference sequence along the trajectory. Theorem 2 turns an MGF control on these increments into an exponential supermartingale, which is the standard vehicle for deriving high-probability concentration in dependent settings. Theorem 3 then applies a PAC-Bayes change-of-measure argument to lift a concentration statement from a fixed A to a distribution Q over A, which is exactly the object produced by Bayesian adaptation.

For later reference, define the conditional and empirical trajectory risks

$$L_T(Q) := \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{A\sim Q}[\mathbb{E}[\ell_t(A) \mid \mathcal{F}_{t-1}]], \qquad \widehat{L}_T(Q) := \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{A\sim Q}[\ell_t(A)]. \tag{19}$$

Theorem 3 can be interpreted as follows: with high probability, the predictive risk $L_T(Q)$ is upper bounded by the in-trajectory fit $\widehat{L}_T(Q)$ plus a complexity penalty that scales like $D_{\mathrm{KL}}(Q\|P)/T$, along with additional concentration terms.

The bound holds for any prior P and any posterior Q. In our hierarchical model, we take $P = P_\phi$ to be the learned matrix-normal prior equation 9, and we restrict Q to be the conjugate posterior $Q_{m,\phi}$ in equation 11. The next subsection specializes the bound to this choice and turns it into a practical meta-training objective.

4.3 Meta-Training Objective

The martingale PAC-Bayes bound in Theorem 3 contains the term $\sum_{t=1}^{T}\psi_t(\lambda)$, which captures how heavy-tailed the centered loss increments $d_t(A)$ are. For meta-training, the key point is that under a uniform conditional MGF control, this term becomes explicit and can be treated as independent of the meta-parameters that define the prior, so it affects the bound only through constants and the overall weighting of the KL term.

Assumption 4 (Uniform moment generating function (MGF) control). *Recall $d_t(A) := \mathbb{E}[\ell_t(A) \mid \mathcal{F}_{t-1}] - \ell_t(A)$. We assume there exists a constant $v > 0$, such that for all $t \in \{1,\dots,T\}$, all A, and all $\lambda \in \mathbb{R}$,*

$$\log \mathbb{E}\big[\exp\big(\lambda d_t(A)\big) \,\big|\, \mathcal{F}_{t-1}\big] \le \frac{\lambda^2 v}{2}. \tag{20}$$

Equivalently, equation 16 holds with $\psi_t(\lambda) = \lambda^2 v/2$, so that $\sum_{t=1}^{T}\psi_t(\lambda) \le \lambda^2 vT/2$, which is the standard conditional sub-Gaussian MGF condition for martingale differences used in martingale concentration and self-normalized analyses (Abbasi-Yadkori et al., 2012; Howard et al., 2020).

Under Assumption 4, Theorem 3 yields a bound of the form

$$L_T(Q) \;\le\; \widehat{L}_T(Q) \;+\; \frac{1}{\lambda T} D_{\mathrm{KL}}(Q\|P) \;+\; \text{terms depending only on } (\delta, \lambda, v, T).$$

Consequently, for fixed (δ, λ, v) and T, the dependence on the meta-parameters ϕ enters only through the empirical fit $\widehat{L}_T(Q)$ and the KL term $D_{\mathrm{KL}}(Q\|P)$. The temperature λ therefore acts as a global reweighting of the KL regularization in the training surrogate, while the remaining λ-dependent terms are independent of ϕ. In our implementation, we fix $\lambda = 1$, which yields the canonical tradeoff between empirical fit and the KL term. It preserves the conjugate Bayesian update used for task adaptation and avoids introducing an additional bound-specific tuning parameter in meta-training. We will investigate optimizing λ for the tightest bound in the theoretical analysis in Section 4.5.

For each training task $m \in [M_{\mathrm{tr}}]$, we observe a trajectory $D_m = \{x_{m,t}\}_{t=0}^{T_m}$ and its regression matrices (X_m, Y_m) defined in equation 3. Given the current meta-parameters $\phi = (W, V, \sigma^2)$, we form the conjugate posterior $Q_{m,\phi}$ in equation 11 with parameters M_m, V_m from equation 12 and equation 13. We define the per-task empirical fit term

$$\mathcal{L}_{\mathrm{fit},m}(\phi) := \frac{1}{T_m}\mathbb{E}_{A\sim Q_{m,\phi}}\left[\sum_{t=1}^{T_m} \ell_{m,t}(A)\right], \tag{21}$$

which is exactly $\widehat{L}_{T_m}(Q_{m,\phi})$ under the Gaussian working likelihood; and the per-task complexity term

$$\mathcal{L}_{\mathrm{kl},m}(\phi) := \frac{1}{T_m} D_{\mathrm{KL}}(Q_{m,\phi} \,\|\, P_\phi)\,. \tag{22}$$

While the empirical fit and KL surrogates are mathematically well-defined, they are susceptible to numerical ill-conditioning in the absence of explicit structural constraints. For instance, V can drift toward poorly conditioned matrices, and W can drift toward unstable dynamics. To ensure numerical robustness and promote stability, we incorporate:

$$\mathcal{R}_{\mathrm{hyper}}(W, V) := \frac{1}{2\tau_W^2}\|W\|_F^2 + \lambda_V \left(\frac{1}{2}\|V - I_d\|_F^2 - \log\det(V)\right), \tag{23}$$

where $\tau_W > 0$ controls the degree of shrinkage for W and $\lambda_V \geq 0$ encourages a well-scaled, well-conditioned V while acting as a log-barrier that prevents the loss of positive definiteness.

To encourage stable shared mean dynamics consistent with Assumption 1, we use a soft penalty

$$\mathcal{R}_{\mathrm{stab}}(W; \rho_0) := \big(\max\{0,\ \rho(W) - \rho_0\}\big)^2, \tag{24}$$

where $\rho(W)$ is the spectral radius and $\rho_0 \in (0, 1)$ is a target stability margin. Optionally, after each gradient step, we apply a hard projection $W \leftarrow \textsc{ProjectStable}(W, \rho_0)$ that enforces $\rho(W) \leq \rho_0$.

To ensure V remains symmetric positive definite during optimization, we parameterize

$$V = LL^\top + \epsilon I_d, \tag{25}$$

where L is a learnable lower-triangular matrix and $\epsilon > 0$ is a small diagonal shift.

Finally, we optimize the following surrogate

$$\min_{\phi}\ \frac{1}{M_{\mathrm{tr}}} \sum_{m=1}^{M_{\mathrm{tr}}} \left[\mathcal{L}_{\mathrm{fit},m}(\phi) + \mathcal{L}_{\mathrm{kl},m}(\phi)\right]\ +\ \mathcal{R}_{\mathrm{hyper}}(W, V)\ +\ \mathcal{R}_{\mathrm{stab}}(W; \rho_0), \tag{26}$$

With the objective equation 26 in place, meta-training reduces to repeatedly (i) computing conjugate posteriors for a minibatch of tasks in closed form and (ii) updating ϕ via stochastic gradients of the resulting surrogate loss.

4.4 Task-Level Inference and Adaptation

After meta-training, we perform task-level inference on previously unseen tasks by adapting the learned prior to a short support trajectory. For each new task m with trajectory length T_m, we first select a support prefix of length $S_m = \min\{T_{\mathrm{sup}}, T_m\}$ and form the corresponding regression matrices X_m^{sup} and Y_m^{sup} from the observed state transitions.

Given the learned meta-parameters $\phi = (W, V, \sigma^2)$, adaptation is carried out via a closed-form Bayesian update. Specifically, the matrix-normal prior is combined with the support data to produce a task-specific posterior $Q_{m,\phi} = \mathcal{MN}(M_m, I_d, V_m)$, where the posterior mean M_m and covariance V_m are given by equation 12 and equation 13. We use the posterior mean $\widehat{A}_m := M_m$ as a point estimate of the task dynamics, while the posterior covariance quantifies task-specific uncertainty and governs the strength of shrinkage toward the meta-prior. We summarize the overall optimization procedure with task adaptation inference Algorithm 1.

4.5 Theoretical Analysis

The previous section showed how the martingale PAC-Bayes framework motivates the PAC-Bayes meta-training objective. In this section, we make this connection explicit by deriving a concrete generalization

Algorithm 1 PBML-LTI Meta-Training and Task Adaptation

Meta-training

Require: Training tasks $\mathcal{T}_{\text{train}}$; steps S; batch size B.

for $s = 1, \dots, S$ **do**
  Sample minibatch $\mathcal{B} \subset \mathcal{T}_{\text{train}}$ with $|\mathcal{B}| = \min(B, |\mathcal{T}_{\text{train}}|)$.
  for each task $m \in \mathcal{B}$ **do**
    Form (X_m, Y_m) from the trajectory as in equation 2.
    Compute posterior parameters (M_m, V_m) via equation 12 and equation 13.
    Evaluate $\mathcal{L}_{\text{fit},m}$ and $\mathcal{L}_{\text{kl},m}$ according to equation 21 and equation 22.
  end for
  $L_{\text{tr}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{m \in \mathcal{B}} \left(\mathcal{L}_{\text{fit},m} + \mathcal{L}_{\text{kl},m} \right) + \mathcal{R}_{\text{hyper}}(W, V) + \mathcal{R}_{\text{stab}}(W)$.
  Adam update of (W, V, σ^2) using ∇L_{tr}.
end for
Save learned $\phi = (W, V, \sigma^2)$.

Task adaptation on new tasks

Require: New tasks $\mathcal{T}_{\text{test}}$; support horizon T_{sup}; (optional) query horizon T_{qry}.

for each task $m \in \mathcal{T}_{\text{test}}$ **do**
  Choose $S_m \leftarrow \min\{T_{\text{sup}}, T_m\}$.
  Form $(X_m^{\text{sup}}, Y_m^{\text{sup}})$ from the first S_m transitions.
  Compute (M_m, V_m) using $(X_m, Y_m) = (X_m^{\text{sup}}, Y_m^{\text{sup}})$ via equation 12 and equation 13.
  Set point estimate $\widehat{A}_m \leftarrow M_m$.
  Rollout over $Q_m = \min\{T_{\text{qry}}, T_m - S_m\}$ using $\widehat{x}_{t+1} = \widehat{A}_m \widehat{x}_t$.
end for
return $\{\widehat{A}_m\}_{m \in \mathcal{T}_{\text{test}}}$ and rollout predictions $\{\widehat{x}_{m,t}\}$.

bound obtained by specializing Theorem 3 under Assumption 4 and examining the effect of the PAC-Bayes temperature λ. In particular, we show that an appropriate choice of λ balances the empirical fit and complexity terms, ensuring that the resulting excess-risk bound remains non-vacuous and decays with trajectory length, thereby justifying the use of a fixed and well-scaled temperature in practice.

Corollary 1 (**Sub-Gaussian martingale PAC-Bayes excess risk**). [4] *Under Assumption 4, for any prior P independent of the trajectory and any (possibly data-dependent) posterior Q, for any $\delta \in (0,1)$ and any $\lambda \in (0,1]$, with probability at least $1-\delta$,*

$$L_T(Q) \le \widehat{L}_T(Q) + \frac{D_{\text{KL}}(Q\|P) + \log(1/\delta)}{\lambda T} + \frac{\lambda v}{2}. \tag{27}$$

Optimizing the right-hand side over $\lambda \in (0,1]$ gives

$$\lambda^\star = \min\left\{1,\ \sqrt{\frac{2\big(D_{\text{KL}}(Q\|P) + \log(1/\delta)\big)}{vT}}\right\}. \tag{28}$$

In the regime where $\lambda^\star < 1$, this simplifies to

$$L_T(Q) \le \widehat{L}_T(Q) + \sqrt{\frac{2v\big(D_{\text{KL}}(Q\|P) + \log(1/\delta)\big)}{T}}. \tag{29}$$

The excess-risk term in equation 29 decays at the canonical rate $O(1/\sqrt{T})$, ensuring that the PAC-Bayes bound remains non-vacuous and exhibits the expected high-probability generalization behavior under sub-Gaussian concentration. The bound also makes explicit the role of meta-learning: for a fixed task posterior Q, a higher-quality prior P reduces the divergence $D_{\text{KL}}(Q|P)$, thereby tightening the gap between empirical in-trajectory fit and the conditional predictive risk. This highlights how learning a transferable prior directly improves generalization to new tasks by lowering the complexity penalty incurred during task adaptation.

[4] The proof is included in Appendix A.4.

5 Performance Evaluation

We evaluate the proposed framework through a comprehensive series of experiments on both synthetic LTI systems and real-world data sets. We begin by detailing the experimental protocol, including competitive baselines and evaluation metrics. Subsequently, we present a comparative analysis demonstrating the advantages of our approach in terms of parameter identification accuracy, multi-step predictive stability, and data efficiency.

5.1 Baselines

Per-task ordinary least squares (OLS). The OLS baseline fits each task independently using the closed-form least squares estimator

$$\widehat{A}_m^{\mathrm{ols}} = Y_m^{\mathrm{sup}}(X_m^{\mathrm{sup}})^\top \Big(X_m^{\mathrm{sup}}(X_m^{\mathrm{sup}})^\top \Big)^{-1}. \tag{30}$$

To ensure numerical stability, we include a small diagonal jitter and employ a pseudoinverse fallback whenever the Gram matrix is ill-conditioned. Note that OLS has no regularization hyperparameter to tune.

Per-task ridge (Ridge). The Ridge baseline fits each task by Tikhonov-regularized least squares

$$\widehat{A}_m^{\mathrm{ridge}}(\lambda_{\mathrm{reg}}) = Y_m^{\mathrm{sup}}(X_m^{\mathrm{sup}})^\top \Big(X_m^{\mathrm{sup}}(X_m^{\mathrm{sup}})^\top + \lambda_{\mathrm{reg}} I_d \Big)^{-1}. \tag{31}$$

We select λ_{reg} once using only the training tasks by grid search over a predefined candidate set, using a stability-oriented criterion. Specifically, we pick the smallest λ_{reg} for which the fitted matrices have mean spectral radius at most a target threshold

$$\mathbb{E}_{m\in\mathcal{T}_{\mathrm{tr}}}\big[\rho(\widehat{A}_m^{\mathrm{ridge}}(\lambda_{\mathrm{reg}}))\big] \le \rho_{\mathrm{target}}, \tag{32}$$

If no candidate satisfies this constraint, we choose the λ_{reg} that minimizes the mean stability violation

$$\mathbb{E}_{m\in\mathcal{T}_{\mathrm{tr}}}\Big[\max\big\{0, \rho(\widehat{A}_m^{\mathrm{ridge}}(\lambda_{\mathrm{reg}})) - \rho_{\mathrm{target}}\big\}\Big]. \tag{33}$$

The selected λ_{reg} is then fixed and reused for all test tasks.

Pooled-prior Ridge. The Pooled-prior Ridge baseline incorporates cross-task information through a pooled mean dynamics estimate. It first computes a pooled matrix

$$\bar{A} := \frac{1}{M_{\mathrm{tr}}} \sum_{m=1}^{M_{\mathrm{tr}}} \widehat{A}_m^{\mathrm{ols,full}}, \tag{34}$$

where $\widehat{A}_m^{\mathrm{ols,full}}$ is the OLS fit using the full training trajectory of task m. Given $\bar{A}$, the task-specific estimate is obtained in closed form as

$$\widehat{A}_m^{\mathrm{pool}}(\lambda_{\mathrm{reg}}) = \Big(Y_m^{\mathrm{sup}}(X_m^{\mathrm{sup}})^\top + \lambda_{\mathrm{reg}}\bar{A} \Big) \Big(X_m^{\mathrm{sup}}(X_m^{\mathrm{sup}})^\top + \lambda_{\mathrm{reg}} I_d \Big)^{-1}. \tag{35}$$

We tune λ_{reg} using the same stability-based grid-search procedure as Ridge. The chosen value is then fixed for evaluation on validation and test tasks.

Shared Subspace. The Shared Subspace baseline posits that the task matrices share a low-dimensional structure in vectorized form. It first computes OLS estimates on the training tasks and vectorizes them to form $\mathrm{vec}(\widehat{A}_m^{\mathrm{ols,full}})$. It then computes the mean vector a_0 and a rank-k PCA basis $U \in \mathbb{R}^{d^2\times k}$ from the centered vectors. For a new task, we restrict the estimate to the form

$$\mathrm{vec}(A) = a_0 + Uc, \tag{36}$$

and fit the coefficients $c \in \mathbb{R}^k$ via ridge-regularized least squares on the support data. Here, λ_{reg} regularizes the coefficient estimation and is selected using the same stability-based grid-search criterion as Ridge. The subspace dimension k is treated as an additional method hyperparameter.

5.2 Evaluation Metrics

Transition matrix estimation error. To evaluate the accuracy the estimated transition matrix $\widehat{A}_m$, we report the Frobenius squared error

$$E_A := \|\widehat{A}_m - A_m\|_F^2, \tag{37}$$

which quantifies the aggregate squared deviation between the estimated and true transition matrices across all entries. This metric treats all matrix elements equally and provides a convenient global measure of parameter recovery (Ziemann et al., 2023).

Trajectory rollout error. To evaluate the practical utility of the estimated dynamics $\widehat{A}_m$, we measure the open-loop multi-step rollout error on a query window by recursively applying the identified transition operator. This metric assesses the model's capacity for long-term trajectory prediction in the absence of intermediate state corrections. It represents a significantly more stringent test than the Frobenius norm of the parameter error ($E_A := \|\widehat{A}_m - A_m\|_F^2$); since inaccuracies in $\widehat{A}_m$ compound over the prediction horizon, even marginal errors can lead to exponential divergence, particularly along unstable or marginally stable eigenmodes.

Starting from the last support state x_{m,S_m}, we generate predictions by repeated next-step updates:

$$\widehat{x}_{m,S_m} = x_{m,S_m}, \qquad \widehat{x}_{m,t+1} = \widehat{A}_m \widehat{x}_{m,t}, \qquad t = S_m, \dots, S_m + Q_m - 1. \tag{38}$$

To ensure rollout result stability and reproducibility, we does not consider the noise term in We then compare the predicted states to the ground-truth query states in open loop. The per-task trajectory rollout error is defined as the sum of squared prediction errors over the query horizon

$$E_{\text{traj}} := \sum_{t=1}^{Q_m} \left\| \widehat{x}_{m,S_m+t} - x_{m,S_m+t} \right\|_2^2. \tag{39}$$

In the result of the experiment, we set $Q_m = 5$ for both following synthetic and real-world experiment, meaning the reported trajectory rollout error is the accumulative error of five consecutive rollout iteration.

Average support length. Additionally, we allow each task to adaptively choose its task-specific support prefix length S_m to form the corresponding regression matrices X_m^{sup} and Y_m^{sup} from the observed state transitions. We report the average selected support length $\overline{S}$ across tasks as an indicator of few-shot capability, with smaller values corresponding to more data-efficient adaptation

$$\overline{S} := \frac{1}{|\mathcal{T}_{\text{test}}|} \sum_{m \in \mathcal{T}_{\text{test}}} S_m, \tag{40}$$

5.3 Synthetic Data Experiment

5.3.1 Task Generation Mechanism

For synthetic experiments, we generate a meta-dataset by first fixing environment-level parameters and then sampling multiple independent LTI tasks from a shared environment distribution, consistent with Assumption 3.

We fix an environment hyper-parameter $\phi_\star = (W_\star, V_\star, \sigma_\star^2)$ and sample tasks i.i.d. as

$$A_m \mid \phi_\star \sim \mathcal{MN}(W_\star,\ I_d,\ V_\star)\,, \qquad m = 1, 2, \dots \tag{41}$$

which matches the prior family equation 9, and we generate trajectories according to equation 1 with Gaussian process noise

$$w_{m,t+1} \sim \mathcal{N}(0, \sigma_\star^2 I_d), \qquad t = 0, \dots, T_m - 1. \tag{42}$$

We construct a stable meta-mean matrix $W_\star \in \mathbb{R}^{d\times d}$ by drawing a Gaussian random matrix and scaling it so that its spectral radius is at most $0.9\,\rho_0$. We set the true column covariance to $V_\star = v_{\text{true}} I_d$ and the process noise variance to $\sigma_\star^2 = \sigma_{\text{true}}^2$.

For each task m, we sample A_m from equation 41, then rollout a trajectory from an initial state $x_{m,0} \sim \mathcal{N}(0, I_d)$, and generate a trajectory according to equation 1

$$x_{m,t+1} = A_m x_{m,t} + w_{m,t+1}, \qquad t = 0, \ldots, T_m - 1. \tag{43}$$

From each trajectory, we form the regression matrices X_m and Y_m as in equation 3 and store them together with the ground-truth dynamics A_m.

5.3.2 Experiment Setup

We consider a synthetic high-dimensional identification setting in which the state dimension exceeds the available trajectory length. We fix the environment parameters to $v_{\text{true}} = 0.5$ and $\sigma_{\text{true}}^2 = 0.01$, and study two dynamical regimes by varying the spectral-radius parameter ρ_0: a near-stable regime with $\rho_0 = 0.95$ and an unstable regime with $\rho_0 = 4.95$. The stability threshold ρ_{target} used to tune the ridge regularization parameter λ_{reg} in equation 32 and equation 33, is set to match the target spectral radius ρ_0. The unstable setting is inherently more challenging, as errors in the estimated transition matrix are amplified under multi-step rollouts, potentially leading to severe prediction instability. Furthermore, we consider three state dimensions $d \in \{10, 25, 50\}$ with a fixed trajectory length $T_m = 25$. In the medium- and high-dimensional cases, the number of unknown parameters in A_m grows as d^2 and far exceeds the number of observed transitions, placing the problem in a few-shot regime that stresses both estimation accuracy and predictive stability.

First, we uniformly sample 100 tasks to compose the meta-training set. We then construct two distinct evaluation sets. The *common-case* test set consists of tasks where the entrywise mean dynamics fall within one standard deviation of the population mean, representing typical in-distribution systems. In contrast, the *edge-case* test set is formed by selecting tasks from the extremes of the overall parameter distribution, representing out-of-distribution systems intended to stress-test robustness under atypical dynamics. Figure 1 depicts the resulting task distributions.

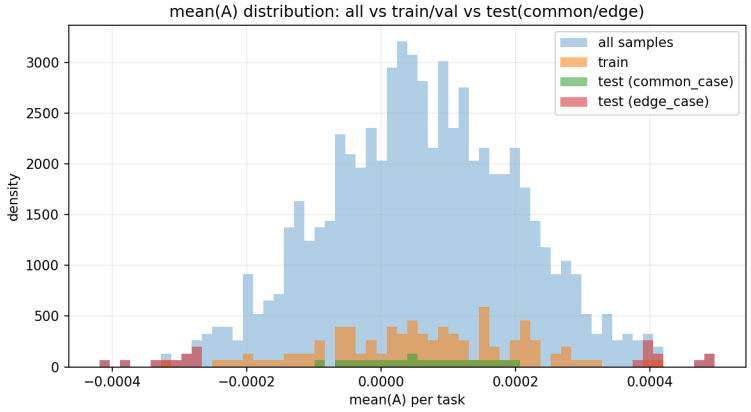

Figure 1: This figure illustrates the distribution of sampled tasks as a function of the entrywise mean of the dynamics matrix A_m. The blue histogram shows all sampled tasks in the initial pool. The yellow region indicates the subset selected for training and validation. The green region corresponds to the in-distribution test tasks, while the red markers denote the edge-case test tasks chosen from the extremes of the distribution.

5.3.3 Experiment Result

Stable-system results Table 1 reports results on the stable synthetic environment with $\rho_0 = 0.95$, where all tasks are generated with a spectral radius below one and therefore do not exhibit unstable rollouts.

Table 1: Performance comparison across methods in common and edge case settings on stable systems ($\rho_0 = 0.95$). E_A is the mean ± standard deviation of the transition matrix error, E_{traj} is the mean ± standard deviation of the rollout error, and $\overline{S}$ reports the average selected support length. Lower is better; best results are in **bold**.

Dimension	Setting	Metric	OLS	Ridge	Pool-prior Ridge	Shared Subspace	PBML-LTI
50	Common-case	E_A	0.8411±.015	0.8410±.015	0.6743±.011	0.8572±.050	**0.2068±.0052**
		E_{traj}	0.0258±.0018	0.0258±.0018	0.0258±.0018	0.0258±.0018	**0.0257±.0018**
		$\overline{S}$	2.00	2.00	2.55	10.25	**1.35**
	Edge-case	E_A	0.8382±.010	0.8381±.010	0.6695±.009	0.8811±.073	**0.2071±.0042**
		E_{traj}	0.0254±.0018	0.0254±.0018	0.0254±.0018	0.0254±.0018	**0.0253±.0018**
		$\overline{S}$	2.00	2.00	2.85	11.05	**1.40**
25	Common-case	E_A	0.7140±.0181	0.7140±.0181	0.8544±.0186	0.7731±.0189	**0.0394±.0020**
		E_{traj}	0.0133±.0013	0.0133±.0013	0.0134±.0018	0.0134±.0013	**0.0121±.0013**
		$\overline{S}$	2.00	2.00	2.00	13.95	**1.20**
	Edge-case	E_A	0.7196±.0181	0.7196±.0181	0.8504±.0230	0.7892±.0191	**0.0399±.00251**
		E_{traj}	0.0135±.0018	0.0135±.0018	0.0122±.0012	0.0121±.0013	**0.0121±.00128**
		$\overline{S}$	2.00	2.00	2.00	13.95	**1.10**
10	Common-case	E_A	0.4449±.0472	0.4447±.0471	0.0887±.0079	0.1088±.0532	**0.0053±.0007**
		E_{traj}	0.0056±.0011	0.0056±.0011	0.0055±.0009	0.0055±.0009	**0.0055±.0009**
		$\overline{S}$	2.65	2.65	1.80	7.95	**1.05**
	Edge-case	E_A	0.4501±.0479	0.4498±.0479	0.0910±.0070	0.0930±.0307	**0.0054±.0008**
		E_{traj}	0.0056±.0011	0.0056±.0011	0.0056±.0011	0.0057±.0009	**0.0056±.0010**
		$\overline{S}$	2.80	2.80	1.90	10.40	**1.10**

A consistent pattern across all dimensions is that the proposed method achieves the lowest transition-matrix error while requiring the shortest support prefixes. Across all three dimensionalities, the posterior-mean estimator induced by the learned prior substantially reduces E_A relative to single-task estimators, such as OLS and Ridge. It also outperforms cross-task baselines, including Pooled-prior Ridge and Shared Subspace methods. Importantly, these gains are achieved with markedly smaller average support lengths $\overline{S}$. This indicates that the learned prior effectively concentrates probability mass on a narrower set of plausible dynamics, enabling more data-efficient adaptation in the few-shot regime.

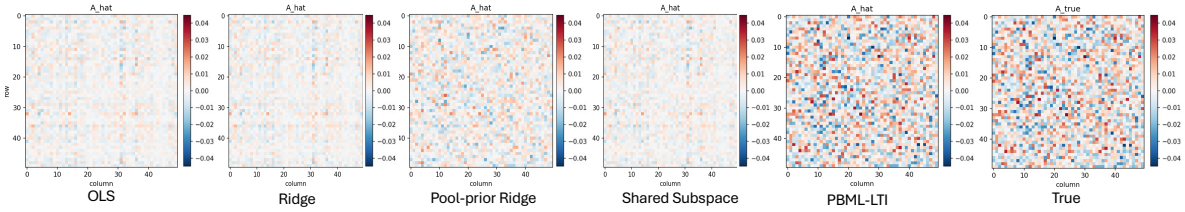

Figure 2: Heatmap comparison of the estimated and ground-truth transition matrices for a representative stable test system in the common-case setting. An extended version is provided in the Appendix, Figure 6.

Figure 2 further illustrates these estimation patterns. Competing baselines tend to produce conservative estimates, where entries are often attenuated or shrunk toward the spectral center. In contrast, our approach yields transition matrices that adhere to stability constraints while more accurately capturing the true spectral structure of the system.

Conversely, differences in rollout error are comparatively small in this stable setting. Because the systems are stable over the query horizon (i.e., $\rho(A) < 1$), open-loop errors do not amplify rapidly. Consequently, many methods attain similar trajectory errors E_{traj} despite large discrepancies in parameter error E_A. notably, baselines like OLS and Ridge can exhibit significantly higher transition-matrix error while still achieving comparable rollout performance, as small per-step prediction errors remain bounded. This highlights that,

Table 2: Performance comparison across methods in common and edge case settings on unstable systems $3(\rho_0 = 4.95)$. E_A is mean $\pm$ std of the transition matrix error, E_{traj} is mean $\pm$ std of the trajectory rollout error, and $\overline{S}$ reports the average of selected support length. Scores are lower and better; best results are in **bold**.

Dimension	Setting	Metric	OLS	Ridge	Pool-prior Ridge	Shared Subspace	PBML-LTI
50	Common-case	E_A	16.923±.345	15.902±.340	7.538±.102	6.419±.201	**0.156±.005**
		E_{traj}	0.040±.004	0.040±.003	0.038±.003	0.038±.003	**0.037±.003**
		$\overline{S}$	9.75	15	15	14.2	**4.15**
	Edge-case	E_A	16.776±.393	15.874±.349	7.532±.079	6.455±.406	**0.157±.003**
		E_{traj}	0.040±.004	0.039±.004	0.038±.003	0.038±.003	**0.036±.003**
		$\overline{S}$	10.15	15	15	13.65	**4.25**
25	Common-case	E_A	10.344±.659	9.907±.619	1.587±.050	35.268±12.856	**0.029±.001**
		E_{traj}	0.088±.058	0.064±.025	**0.053±.024**	27.151±28.807	0.080±.052
		$\overline{S}$	14.1	14.9	14.75	14.35	**6.05**
	Edge-case	E_A	11.023±.938	10.445±.856	1.586±.057	33.889±7.925	**0.029±.001**
		E_{traj}	0.086±.049	0.071±.034	**0.042±.009**	28.933±47.100	0.086±.062
		$\overline{S}$	13.5	14.55	14.9	14.3	**6**
10	Common-case	E_A	0.845±.764	0.697±.581	**0.007±.002**	0.013±.005	0.139±.046
		E_{traj}	193.704±459.303	**57.628±154.718**	21935.364±69327.156	721888.809±1044209.839	57.919±144.808
		$\overline{S}$	13.35	14.3	9.25	**3.2**	14.35
	Edge-case	E_A	0.572±.630	0.515±.566	**0.008±.001**	0.014±.003	0.132±.040
		E_{traj}	296.091±755.920	131.024±389.381	16433.662±49301.429	709906.413±1155741.042	**31.400±38.879**
		$\overline{S}$	13.9	14.15	10.1	**3.2**	14.45

in stable regimes, multi-step rollout metrics are often less sensitive to parameter-estimation quality than they are in unstable or near-explosive regimes.

Finally, performance on the edge-case test tasks closely tracks the common-case results across all three dimensions. The ranking of methods remains largely unchanged, with our approach minimizing E_A while maintaining competitive rollout error. This suggests that, under stable dynamics, the learned prior transfers reliably to both typical and atypical systems, supporting robust few-shot identification.

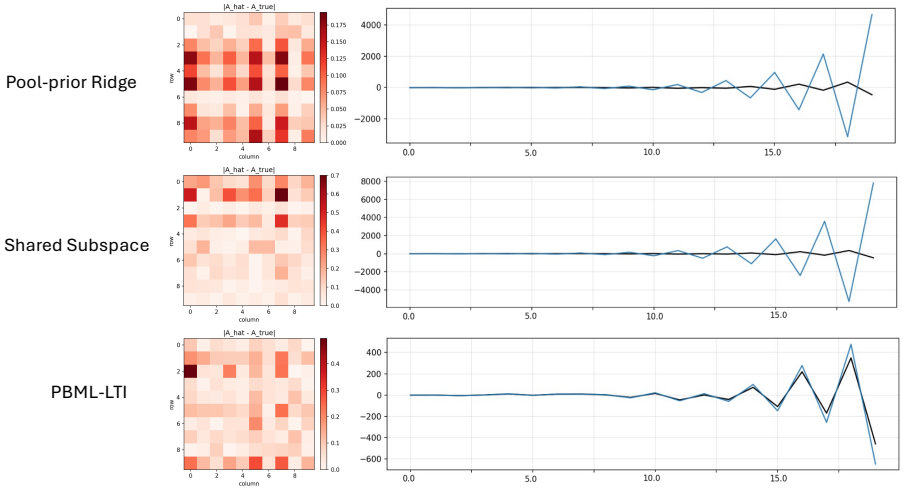

Figure 3: The left panels show heatmaps of the absolute differences between the estimated and ground-truth transition matrices for a representative unstable test system with state dimension $d = 10$ in the common-case setting. The right panels display open-loop trajectory rollouts generated using the estimated transition matrix (black) alongside the corresponding ground-truth trajectories (blue) for the same system. An extended visualization of the estimated transition matrix, the ground-truth transition matrix, and their absolute differences is provided in Figure 8 in the Appendix, while the corresponding open-loop rollout trajectories are shown in Figure 9 in the Appendix.

Unstable-system results. Table 2 presents results for the unstable environment, where $\rho_0 = 4.95$ and prediction errors can amplify rapidly. In this regime, accurate long-horizon rollout requires not only minimizing average matrix error, but also strictly controlling errors along unstable modes. Consequently, some methods may achieve small parameter error E_A, yet still suffer from large trajectory error E_{traj}.

In the high-dimensional setting ($d = 50$), increasing the support length reduces rollout error across all baselines; nevertheless, significant disparities persist in transition-matrix recovery. OLS and Ridge necessitate average support prefixes of approximately 10–15 transitions yet still suffer from substantial parameter error. While Pooled-prior Ridge and Shared Subspace improve E_A relative to single-task baselines, they fall short of optimal performance. In contrast, the proposed method achieves the lowest E_A and superior rollout error using significantly shorter prefixes, demonstrating robust few-shot adaptation even under unstable dynamics.

For $d = 25$, the proposed method again yields the best parameter recovery, requiring substantially fewer support transitions than competing approaches. Regarding rollout accuracy, Pooled-prior Ridge attains the lowest E_{traj} in this specific regime, likely by leveraging longer prefixes and aggressive shrinkage toward the pooled mean. However, the proposed method remains competitive in rollout performance while utilizing less support data, highlighting a favorable trade-off in sample efficiency. Conversely, the Shared Subspace baseline performs poorly, exhibiting very large E_A and correspondingly high rollout error; this suggests that imposing a fixed low-dimensional constraint on $\text{vec}(A_m)$ is brittle in the presence of heterogeneous, unstable dynamics.

For $d = 10$, the behavior shifts notably: cross-task baselines, specifically Pooled-prior Ridge and Shared Subspace, can achieve extremely low parameter error E_A while simultaneously exhibiting catastrophic rollout error. This discrepancy highlights a fundamental limitation of Frobenius-type metrics: $\|\widehat{A}_m - A_m\|_F$ measures an average entrywise deviation, whereas open-loop prediction is governed principally by the spectral structure of the dynamics. Specifically, when the system contains unstable or weakly damped modes, a seemingly negligible perturbation in $\widehat{A}_m$ can shift the dominant eigenvalue or distort its associated eigenvector. Such spectral errors are amplified multiplicatively under repeated iteration of $\widehat{A}_m$, leading to massive deviations in long-horizon rollouts even when E_A appears negligible.

This phenomenon is especially pronounced for Pooled-prior Ridge and Shared Subspace. Although these methods produce transition estimates that closely align with the true matrix in the Frobenius norm, their imposed structural biases—whether shrinkage toward a pooled mean or restriction to a fixed low-dimensional subspace—can result in slight mis-estimation of the dominant spectral mode. When this mode lies near or beyond the stability boundary, even a minute spectral shift precipitates rapid trajectory divergence.

In contrast, Ridge and the proposed method demonstrate the most reliable rollout performance. Our approach matches Ridge in common-case rollout accuracy and attains superior performance on edge cases; however, it no longer utilizes the shortest prefixes on average. This suggests that in low-dimensional settings—where long-horizon behavior is dominated by a small number of unstable directions—additional support transitions are requisite to accurately resolve these critical spectral modes. Figure 3 illustrates representative examples of this phenomenon, showing how transition estimates that are nearly indistinguishable in the Frobenius norm can yield dramatically different rollout behaviors due to spectral misalignment.

Overall, the unstable-system experiments highlight that the benefits of a learned prior are most pronounced in the high-dimensional, few-shot regime. Here, the proposed method achieves strong parameter recovery and stable rollout using substantially smaller support budgets. Conversely, in low dimensions, rollout stability becomes the dominant challenge and can effectively decouple from the parameter error E_A.

5.4 Real Data Experiment

5.4.1 Functional Magnetic Resonance Imaging (fMRI) Data set

We construct a real-data LTI benchmark from the OpenNeuro data set ds000244 (Pinho et al., 2018). Raw fMRI volumes are converted into multivariate Region-of-Interest (ROI) time series by averaging voxel signals within atlas-defined parcels using the Schaefer parcellation (Schaefer et al., 2018). The resulting ROI signals

Table 3: Performance comparison on the fMRI dataset. Results are reported as mean ± standard deviation across tasks. Lower values indicate better performance. The row $\overline{S}$ reports the average selected support prefix length.

Metric	OLS	Ridge	Pool-prior Ridge	Shared Subspace	PBML-LTI
E_A	141.766 ± 259.513	142.493 ± 259.174	129.643 ± 263.244	147.876 ± 256.798	**124.734 ± 268.471**
E_{traj}	134.126 ± 221.939	134.078 ± 221.941	125.892 ± 209.232	679.799 ± 968.563	**123.663 ± 206.714**
$\overline{S}$	59.42	61.21	60.85	24.00	**21.21**

are further preprocessed with standardization, motion-confound regression when available, and per-run mean centering.

To obtain many related identification tasks from each run, we segment the ROI time series into overlapping windows of length L with a stride of s. Each window is treated as one LTI task instance m with states $\{x_{m,t}\}_{t=0}^{L-1}$, and we form the one-step regression matrices

$$X_m := [x_{m,0}, \dots, x_{m,L-2}] \in \mathbb{R}^{d\times(L-1)}, \qquad Y_m := [x_{m,1}, \dots, x_{m,L-1}] \in \mathbb{R}^{d\times(L-1)}. \tag{44}$$

These (X_m, Y_m) pairs are saved in the same task format as the synthetic experiments (one task per window).

Unlike synthetic data, fMRI does not provide a physical ground-truth transition matrix. We therefore define a reference transition matrix A_m^{ref} for each window by ridge regression on the same window

$$A_m^{\text{ref}} \;:=\; \arg\min_{A\in\mathbb{R}^{d\times d}} \; \big\|Y_m - AX_m\big\|_F^2 \;+\; \lambda_{\text{ref}}\|A\|_F^2, \tag{45}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $\lambda_{\text{ref}} = 0.0001$ is a fixed ridge weight used only for constructing the reference label. For consistency, the ridge regularization parameter λ_{reg} in equation 32 and equation 33 is set equal to λ_{ref}. This optimization has a closed-form solution

$$A_m^{\text{ref}} \;=\; Y_m X_m^\top \Big(X_m X_m^\top + \lambda_{\text{ref}} I_d\Big)^{-1}. \tag{46}$$

Importantly, the all method itself is trained from trajectories (X_m, Y_m) and the reference matrices A_m^{ref} are only a convenient target for evaluation.

In our experiments, we use a predefined train–test split constructed at the window level. Each task corresponds to a short temporal window extracted from a single fMRI run, and windows from different subjects, sessions, and experimental conditions are distributed across the training and test sets according to this split. The original BIDS task labels are retained only as metadata for grouping and reporting and are not used by the learning algorithms. Importantly, each window is treated as an independent task instance for meta-learning, so that both the training and test sets contain multiple tasks drawn from the same heterogeneous pool of experimental conditions.

In total, the training set contains 141 task instances spanning 17 distinct task labels, while the test set contains 14 task instances spanning 8 task labels. Each task instance has state dimensionality $d = 200$ and trajectory length $T = 100$. [5]

5.4.2 Experiment Result

Table 3 summarizes identification and prediction performance on the fMRI benchmark, reporting mean ± standard deviation across window-level tasks. Overall, PBML-LTI achieves the best average performance on both the transition-matrix metric E_A and the open-loop rollout metric E_{traj}, while also selecting the shortest average support prefix length.

[5] More technical details of the dataset processing in Appendix A.5.

In terms of multi-step prediction, PBML-LTI attains the lowest rollout error, improving over OLS and Ridge and Pool-prior Ridge, while the Shared Subspace baseline performs poorly in rollout, indicating severe instability under open-loop iteration on this real-data benchmark. The improvement in transition-matrix error is comparatively modest but consistent: PBML-LTI achieves the lowest E_A, slightly better than Pool-prior Ridge and noticeably better than OLS and Ridge. These results are consistent with the fact that fMRI windows are only approximately LTI and the reference matrix is itself defined via a ridge fit, so transition-matrix recovery is inherently noisy, whereas rollout error more directly reflects practical predictive stability.

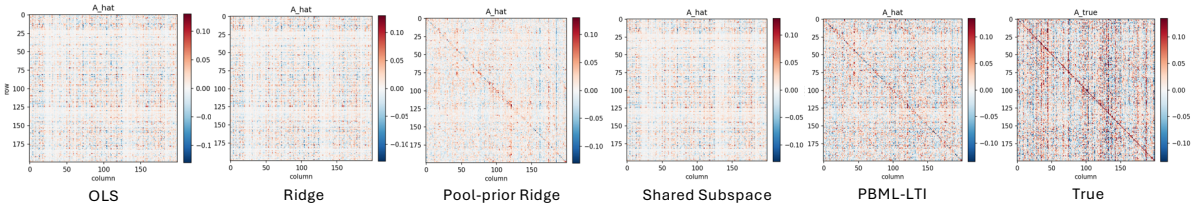

Figure 4: This figure visualizes the estimated and ground-truth transition matrices for a representative system from the fMRI dataset. For a more detailed analysis, including an extended set of heatmaps and the corresponding open-loop rollout trajectories, please refer to Figure 11 in the Appendix.

The average selected support length further highlights data efficiency. PBML-LTI achieves the best errors while using the shortest prefixes, closely followed by Shared Subspace , whereas OLS, Ridge, and Pool-prior Ridge require substantially longer prefixes. Notably, Shared Subspace uses short prefixes but does not translate this into accurate prediction, suggesting that its fixed low-dimensional constraint can be mismatched to the heterogeneous dynamics across subjects and task conditions, leading to unstable rollouts. Finally, the large standard deviations across all methods indicate substantial variability across windows, which is expected in this real-data setting and underscores the importance of methods that remain robust when adapting from limited support data.

6 Discussion

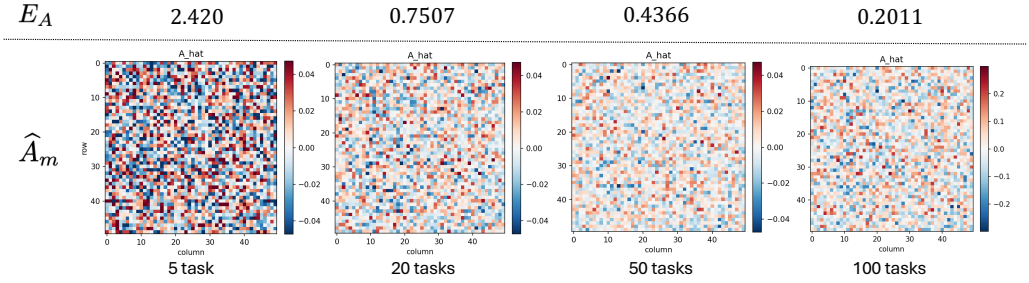

Figure 5: This figure visualizes the estimated transition matrices and the corresponding estimation errors for varying numbers of available training tasks. All tasks are sampled from stable systems with spectral radius $\rho_0 = 0.95$ and dimension d = 50.

In PBML-LTI, task-level uncertainty is quantified by the posterior column covariance V_m equation 12. Consequently, the fidelity of these uncertainty estimates relies heavily on the meta-training stage's ability to accurately recover the shared meta-parameters (W, V, σ^2). When only a small number of training tasks is available, the learned prior covariance V is often poorly scaled or ill-conditioned, resulting in posterior distributions that are either overly confident or insufficiently regularized.

To investigate this effect, we vary the number of training tasks used during meta-training. As illustrated in Figure 5, with only five training tasks, the resulting estimates $\widehat{A}_m$ exhibit overly aggressive values, indicating

that the inferred posterior does not adequately reflect uncertainty. As the number of training tasks increases, the learned prior covariance V becomes a more reliable estimate of cross-task variability, yielding posterior covariances V_m that better capture remaining uncertainty after observing the support trajectory. Quantitatively, increasing the number of training tasks from five to one hundred reduces the transition-matrix estimation error by approximately 90%, highlighting the importance of sufficient meta-level data for reliable uncertainty quantification.

Overall, we studied few-shot identification of linear time-invariant dynamical systems in a meta-learning setting, where each task provides only a short, temporally dependent trajectory and reliable adaptation to previously unseen systems is required. Our proposed PBML-LTI framework learns a transferable prior over system dynamics from multiple related tasks and performs task-level adaptation through closed-form Bayesian inference, yielding both accurate point estimates and principled uncertainty quantification while remaining computationally efficient.

On the theoretical side, we developed PAC-Bayesian generalization guarantees tailored to trajectory data by leveraging martingale concentration tools. These results extend existing PAC-Bayes analyses for meta-learning to settings with within-task temporal dependence and clarify how empirical trajectory fit and a KL-based complexity penalty jointly govern generalization to new systems. The analysis also provides a formal justification for the PAC-Bayes surrogate training objective used in practice.

Empirically, we demonstrated that PBML-LTI consistently improves data efficiency and robustness in challenging regimes, including high-dimensional synthetic systems and real fMRI-derived LTI tasks, where linear dynamics only hold approximately. In particular, the proposed method achieves strong multi-step prediction performance while adapting from shorter support trajectories than competing baselines, highlighting its suitability for few-shot system identification.

There are several future research directions. Extending the framework to accommodate controlled systems, partial observability, or nonlinear dynamics would significantly broaden its applicability. Additionally, incorporating richer structured prior distributions represent promising avenues for further research.

Code and Data Availability The code repository is available at https://anonymous.4open.science/r/PAC_Bayesian_Meta_Learning_LTI. The synthetic dataset generation code is included in the code repository. The fMRI dataset is available https://openneuro.org/datasets/ds000244/versions/1.0.0, provided by Pinho et al. (2018).

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Online least squares estimation with self-normalized processes: An application to bandit problems. *arXiv preprint arXiv:1102.2670*, 2011. doi: 10.48550/arXiv.1102.2670.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 1–9. PMLR, 2012.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

John T. Bosworth. Linearized aerodynamic and control law models of the x-29a airplane and comparison with flight data. NASA Technical Memorandum NASA-TM-4356, National Aeronautics and Space Administration, 1992.

Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

Yiting Chen, Ana M Ospina, Fabio Pasqualetti, and Emiliano Dall'Anese. Multi-task system identification of similar linear time-invariant dynamical systems. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 7342–7349. IEEE, 2023.

A. P. Dawid. Some matrix-variate distribution theory: Notational considerations and a bayesian application. *Biometrika*, 68(1):265–274, 1981. doi: 10.1093/biomet/68.1.265.

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Finite time identification in unstable linear systems. *Automatica*, 96:342–353, 2018.

Alec Farid and Anirudha Majumdar. Generalization bounds for meta-learning via pac-bayes and uniform stability. *Advances in neural information processing systems*, 34:2173–2186, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018. doi: 10.48550/arXiv.1801.08930.

Jiechao Guan and Zhiwu Lu. Fast-rate PAC-bayesian generalization bounds for meta-learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7930–7948. PMLR, 2022.

Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*. Chapman and Hall/CRC, 2018.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17:257–317, 2020. doi: 10.1214/18-PS321.

Yan Ji, Jian Liu, and Haibo Liu. An identification algorithm of generalized time-varying systems based on the taylor series expansion and applied to a pH process. *Journal of Process Control*, 128:103007, 2023. doi: 10.1016/j.jprocont.2023.103007.

Jiahe Lin, Huitian Lei, and George Michailidis. A VAE-based framework for learning multi-level neural granger-causal connectivity. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=kNCZ95mw7N.

Tianyu Liu, Jie Lu, Zheng Yan, and Guangquan Zhang. PAC-bayes bounds for meta-learning with data-dependent prior. *arXiv preprint arXiv:2102.03748*, 2021. doi: 10.48550/arXiv.2102.03748.

Daniel Ludvig and Eric J. Perreault. System identification of physiological systems using short data segments. *IEEE Transactions on Biomedical Engineering*, 59(12):3541–3549, 2012. doi: 10.1109/TBME.2012.2220767.

Andreas Maurer. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6:967–994, 2005.

David A. McAllester. PAC-bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pp. 164–170. ACM, 1999. doi: 10.1145/307400.307435.

Aditya Modi, Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Joint learning of linear time-invariant dynamical systems. *Automatica*, 164:111635, 2024. doi: 10.1016/j.automatica.2024.111635.

Deepan Muthirayan, Dileep Kalathil, and Pramod P Khargonekar. Meta-learning online control for linear dynamical systems. *IEEE Transactions on Automatic Control*, 2025.

Massimiliano Patacchiola, Jack Turner, Elliot J. Crowley, Michael O'Boyle, and Amos Storkey. Bayesian meta-learning for the few-shot setting via deep kernels. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16108–16118, 2020.

M. Hashem Pesaran. *Time Series and Panel Data Econometrics*. Oxford University Press, 2015.

Ana Luisa Grilo Pinho, Lucie Hertz-Pannier, and Bertrand Thirion. "individual brain charting". OpenNeuro, 2018.

Arezou Rezazadeh. A unified view on PAC-bayes bounds for meta-learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18576–18595. PMLR, 2022.

Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5610–5618. PMLR, 2019.

Alexander Schaefer, Ru Kong, Evan M. Gordon, Timothy O. Laumann, Xi-Nian Zuo, Avram J. Holmes, Simon B. Eickhoff, and B. T. Thomas Yeo. Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity mri. *Cerebral Cortex*, 28(9):3095–3114, 2018. doi: 10.1093/cercor/bhx179.

Matthias Seeger. PAC-bayesian generalisation error bounds for gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I. Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Proceedings of the 31st Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 439–473. PMLR, 2018.

James H. Stock and Mark W. Watson. Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In John B. Taylor and Harald Uhlig (eds.), *Handbook of Macroeconomics*, volume 2, pp. 415–525. Elsevier, 2016.

Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, 2 edition, 1997. doi: 10.1007/b98971.

Ingvar Ziemann, Anastasios Tsiamis, Bruce Lee, Yassir Jedra, Nikolai Matni, and George J Pappas. A tutorial on the non-asymptotic theory of system identification. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 8921–8939. IEEE, 2023.

A Technical Appendix

A.1 Proof of Theorem 1

Proof. Under the Gaussian conditional model

$$p_\sigma(x_t \mid x_{t-1}, A) = \mathcal{N}(Ax_{t-1}, \sigma^2 I_d), \tag{47}$$

the conditional density is given by

$$p_\sigma(x_t \mid x_{t-1}, A) = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{1}{2\sigma^2}\|x_t - Ax_{t-1}\|_2^2\right). \tag{48}$$

Taking the negative of the logarithm yields

$$\ell_t(A) = \frac{1}{2\sigma^2}\|x_t - Ax_{t-1}\|_2^2 + \frac{d}{2}\log(2\pi\sigma^2), \tag{49}$$

which proves equation 14.

Next, fix any A and define

$$d_t(A) := \mathbb{E}[\ell_t(A) \mid \mathcal{F}_{t-1}] - \ell_t(A). \tag{50}$$

By assumption, $\ell_t(A)$ is $\mathcal{F}_t$-measurable and integrable. Therefore,

$$\mathbb{E}[d_t(A) \mid \mathcal{F}_{t-1}] = 0, \tag{51}$$

which shows that $\{d_t(A)\}_{t=1}^T$ is a martingale difference sequence. □

A.2 Proof of Theorem 2

Proof. Fix A and $\lambda \in (0, 1]$. Define

$$S_t(A) := \sum_{i=1}^{t} d_i(A), \qquad S_0(A) = 0, \tag{52}$$

and

$$Z_t(A; \lambda) := \exp\left(\lambda S_t(A) - \sum_{i=1}^{t} \psi_i(\lambda)\right), \qquad Z_0(A; \lambda) = 1. \tag{53}$$

Since $S_t(A) = S_{t-1}(A) + d_t(A)$,

$$Z_t(A; \lambda) = Z_{t-1}(A; \lambda) \exp(\lambda d_t(A) - \psi_t(\lambda)). \tag{54}$$

Taking its conditional expectation yields

$$\mathbb{E}[Z_t(A; \lambda) \mid \mathcal{F}_{t-1}] \le Z_{t-1}(A; \lambda), \tag{55}$$

so $\{Z_t(A; \lambda)\}$ is a nonnegative supermartingale. □

Lemma 1 (Donsker–Varadhan change of measure)**.** *Let P and Q be probability measures on the same measurable space. If $Q \ll P$, then for any measurable function f (for which the expressions are well-defined),*

$$\mathbb{E}_{A\sim Q}[f(A)] \;\le\; D_{\mathrm{KL}}(Q\|P) + \log \mathbb{E}_{A\sim P}\left[e^{f(A)}\right]. \tag{56}$$

If $Q \not\ll P$, then $D_{\mathrm{KL}}(Q\|P) = +\infty$ and equation 56 holds trivially.

Proof. Assume $Q \ll P$ and let $r := \frac{dQ}{dP}$. Then

$$D_{\mathrm{KL}}(Q\|P) = \mathbb{E}_Q[\log r]. \tag{57}$$

Using $r^{-1} = \frac{dP}{dQ}$ and Jensen's inequality,

$$\begin{aligned}\mathbb{E}_Q[f(A)] - D_{\mathrm{KL}}(Q\|P) = \mathbb{E}_Q[f(A) - \log r(A)] &= \mathbb{E}_Q\left[\log\left(\frac{e^{f(A)}}{r(A)}\right)\right] \\ &\le \log \mathbb{E}_Q\left[\frac{e^{f(A)}}{r(A)}\right] = \log \int \frac{e^{f(a)}}{r(a)}\, dQ(a) = \log \int e^{f(a)}\, dP(a) = \log \mathbb{E}_P[e^{f(A)}].\end{aligned} \tag{58}$$

Rearranging gives equation 56. □

A.3 Proof of Theorem 3

Proof. Fix $\lambda \in (0, 1]$. Define the mixture supermartingale

$$\overline{Z}_t(\lambda) := \mathbb{E}_{A\sim P}[Z_t(A;\lambda)]. \tag{59}$$

Then

$$\mathbb{E}[\overline{Z}_t(\lambda) \mid \mathcal{F}_{t-1}] \le \overline{Z}_{t-1}(\lambda), \tag{60}$$

and hence

$$\mathbb{E}[\overline{Z}_T(\lambda)] \le 1. \tag{61}$$

By Markov's inequality, with probability at least $1 - \delta$,

$$\overline{Z}_T(\lambda) \le \frac{1}{\delta}. \tag{62}$$

Applying Lemma 1 yields

$$\mathbb{E}_{A\sim Q}[S_T(A)] \le \frac{1}{\lambda}\left(D_{\mathrm{KL}}(Q\|P) + \log\frac{1}{\delta} + \sum_{t=1}^{T}\psi_t(\lambda)\right). \tag{63}$$

Substituting

$$S_T(A) = \sum_{t=1}^{T}\Big(\mathbb{E}[\ell_t(A) \mid \mathcal{F}_{t-1}] - \ell_t(A)\Big) \tag{64}$$

and rearranging gives equation 18. □

A.4 Proof of Corollary 1

Proof. Under Assumption 4,

$$\sum_{t=1}^{T}\psi_t(\lambda) \le \frac{\lambda^2 vT}{2}. \tag{65}$$

Plugging into Theorem 3 yields

$$L_T(Q) \le \widehat{L}_T(Q) + \frac{D_{\mathrm{KL}}(Q\|P) + \log(1/\delta)}{\lambda T} + \frac{\lambda v}{2}. \tag{66}$$

Define

$$g(\lambda) := \frac{a}{\lambda T} + \frac{\lambda v}{2}, \tag{67}$$

with $a = D_{\mathrm{KL}}(Q\|P) + \log(1/\delta)$. The minimizer satisfies

$$\lambda^\star = \sqrt{\frac{2a}{vT}}. \tag{68}$$

Substitution yields

$$L_T(Q) \le \widehat{L}_T(Q) + \sqrt{\frac{2va}{T}}, \tag{69}$$

which completes the proof. □

A.5 Functional Magnetic Resonance Imaging (fMRI) Data set

We construct a real-data LTI benchmark from the OpenNeuro dataset ds000244 (Individual Brain Charting) (Pinho et al., 2018). The dataset is publicly available at https://openneuro.org/datasets/ds000244/versions/1.0.0. The raw data follow the Brain Imaging Data Structure (BIDS) convention, together with metadata and tabular confound files.

Schaefer parcellation To obtain a multivariate state sequence from each fMRI run, we convert voxel-level Blood-Oxygen-Level-Dependent (BOLD) signals into a d-dimensional ROI time series by averaging voxel signals within atlas-defined parcels using the Schaefer parcellation (Schaefer et al., 2018). Let $y_v(t)$ denote the BOLD signal at voxel v and time index t. For parcel (ROI) j with voxel set Ω_j, we define the parcel-averaged signal

$$x_t[j] \;:=\; \frac{1}{|\Omega_j|} \sum_{v \in \Omega_j} y_v(t), \qquad j = 1, \dots, d, \tag{70}$$

yielding a multivariate state vector $x_t \in \mathbb{R}^d$ at each time t. In our experiments we use a Schaefer atlas with $d = 200$ parcels, so each fMRI run becomes a length-T trajectory $\{x_t\}_{t=0}^{T}$ in $\mathbb{R}^{200}$.

Run-level preprocessing. For each run-level ROI time series, we apply standard time-series preprocessing to improve comparability across runs and to reduce nuisance variation. Concretely, we (i) standardize each ROI time series over time, (ii) regress out motion confounds when a confounds TSV is available, and (iii) mean-center the resulting ROI signals within each run. If we write the ROI matrix as $Z \in \mathbb{R}^{T \times d}$ with rows $x_t^\top$, and let $C \in \mathbb{R}^{T \times q}$ denote a subspace spanned by C:

$$Z \;\leftarrow\; \big(I - C(C^\top C)^\dagger C^\top\big) Z, \tag{71}$$

followed by per-run mean-centering $Z \leftarrow Z - \mathbf{1}\mu^\top$, where $\mu = \frac{1}{T}\sum_{t=1}^{T} x_t$.

Window-level LTI task construction. To obtain many related identification tasks from each run, we segment each ROI time series into overlapping temporal windows of fixed length. Each window is treated as a separate task instance m with states $\{x_{m,t}\}_{t=0}^{T_m}$, and we form the one-step regression matrices

$$X_m := [x_{m,0}, \dots, x_{m,T_m-1}] \in \mathbb{R}^{d \times T_m}, \qquad Y_m := [x_{m,1}, \dots, x_{m,T_m}] \in \mathbb{R}^{d \times T_m}. \tag{72}$$

These matrices define the window-level matrix regression view $Y_m \approx A_m X_m$ used throughout the paper. In our released split, each window has $d = 200$ and $T_m = 100$ transitions.

Reference transition matrix for evaluation. Unlike synthetic data, fMRI does not provide a physical ground-truth transition matrix. We therefore define a reference transition matrix A_m^{ref} for each window by fitting a ridge-regularized one-step linear map:

$$A_m^{\mathrm{ref}} \;:=\; \arg\min_{A \in \mathbb{R}^{d \times d}} \; \|Y_m - A X_m\|_F^2 \;+\; \lambda_{\mathrm{ref}} \|A\|_F^2, \tag{73}$$

which has the closed-form solution

$$A_m^{\mathrm{ref}} \;=\; Y_m X_m^\top \Big(X_m X_m^\top + \lambda_{\mathrm{ref}} I_d \Big)^{-1}. \tag{74}$$

We emphasize that A_m^{ref} is used only as a consistent evaluation reference to compute E_A; all methods are trained and adapted from trajectories (X_m, Y_m).

Train–test split and task labels. We use a predefined train–test split constructed at the window level. Windows from different subjects, sessions, and experimental conditions are distributed across training and test according to this split. The original BIDS task labels are retained only as metadata for grouping and reporting, and are not used by the learning algorithms; each window is treated as an independent task instance for meta-learning.

In the split used in our experiments, the training set contains 141 window-level task instances spanning 17 task labels, while the test set contains 14 window-level instances spanning 8 task labels. Across the full split, the task labels include:

- **Archi:** ArchiEmotional, ArchiSocial, ArchiSpatial, ArchiStandard.
- **HCP-style:** HcpEmotion, HcpGambling, HcpLanguage, HcpMotor, HcpRelational, HcpSocial, HcpWm.
- **RSVP language:** RSVPLanguage00, RSVPLanguage01, RSVPLanguage02, RSVPLanguage03, RSVPLanguage04, RSVPLanguage05.

B Additional Visualization

For qualitative evaluation, we provide an extended version of the visualizations shown in the main experimental section. Specifically, we visualize a representative test system by presenting heatmaps of the estimated transition matrix, the ground-truth transition matrix, and their absolute differences. In addition, we plot open-loop rollout trajectories for a system of the same dimensionality to illustrate how errors in transition-matrix estimation propagate to trajectory-level behavior over time.

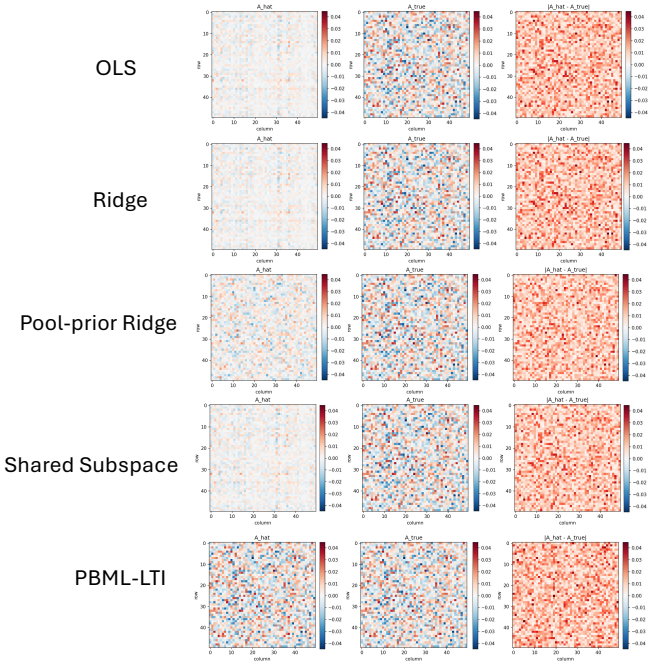

Figure 6: Extended version of Figure 2: it includes heatmaps of the estimated transition matrix, the ground-truth transition matrix, and their absolute differences, providing a more detailed comparison of estimation accuracy across methods.

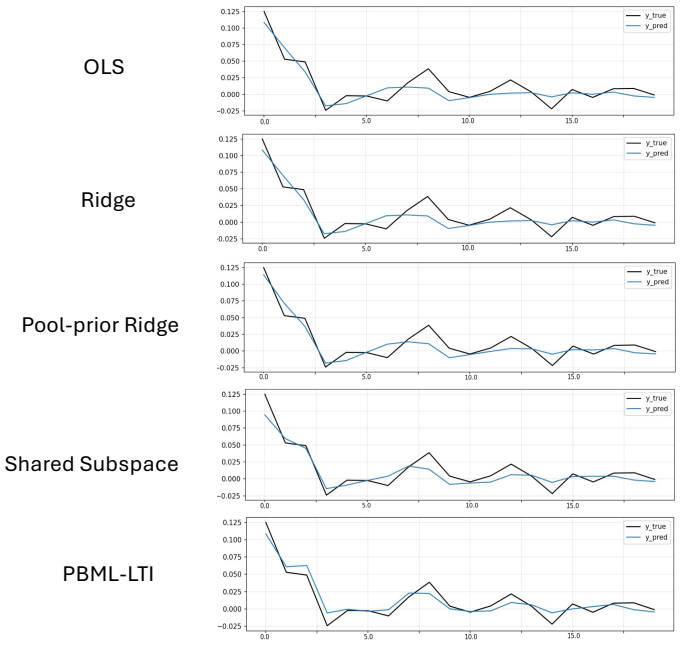

Figure 7: Open-loop rollout trajectories for the system shown in Figure 6 with the same dimensionality. The ground-truth trajectory is shown in black, and the trajectory predicted using the estimated transition matrix is shown in blue.

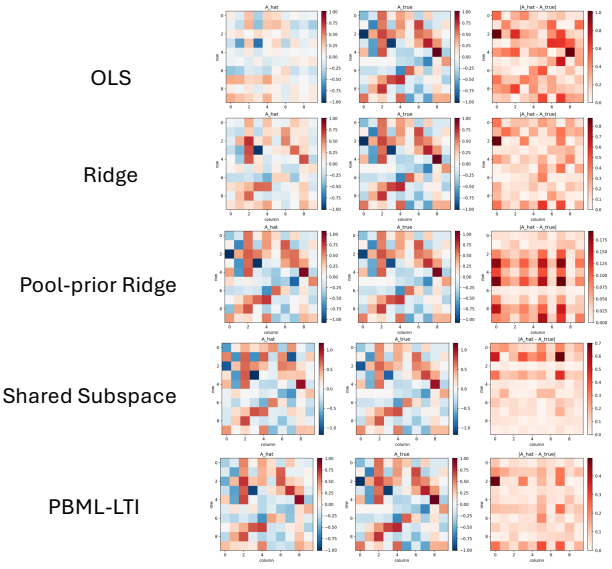

Figure 8: Extended version of Figure 3: it includes heatmaps of the estimated transition matrices, the ground-truth transition matrix, and their absolute differences. These visualizations provide a more detailed comparison of estimation accuracy across all methods.

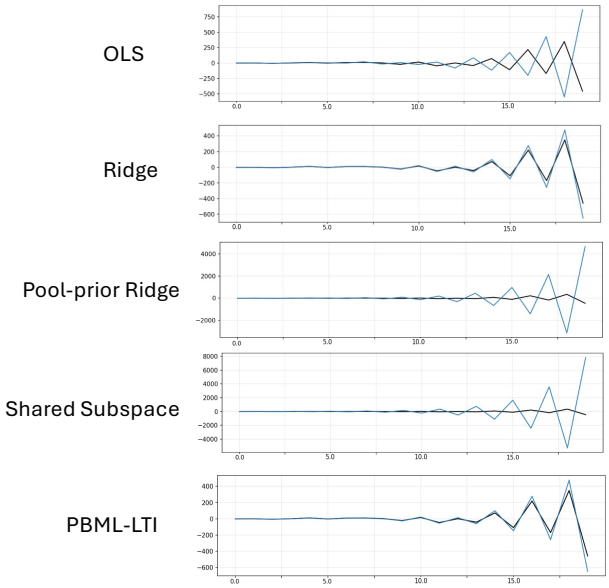

Figure 9: Open-loop rollout trajectories for the system shown in Figure 8 with the same dimensionality. The ground-truth trajectory is shown in black, and the trajectory predicted using the estimated transition matrix is shown in blue.

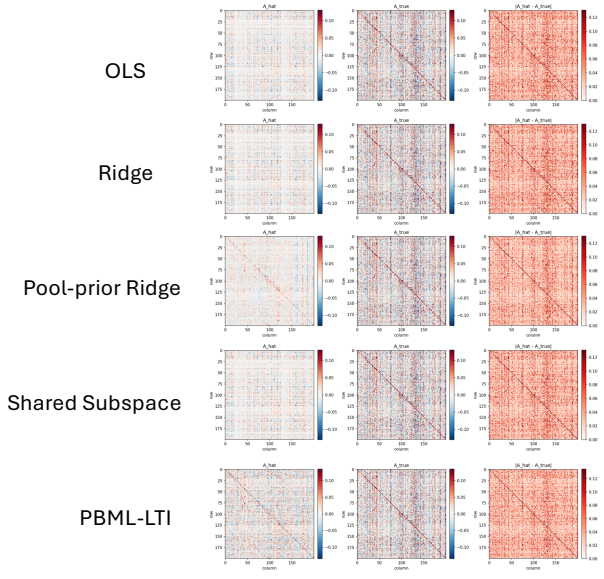

Figure 10: Extended version of Figure 4: it includes heatmaps of the estimated transition matrices, the ground-truth transition matrix, and their absolute differences. These visualizations provide a more detailed comparison of estimation accuracy across all methods

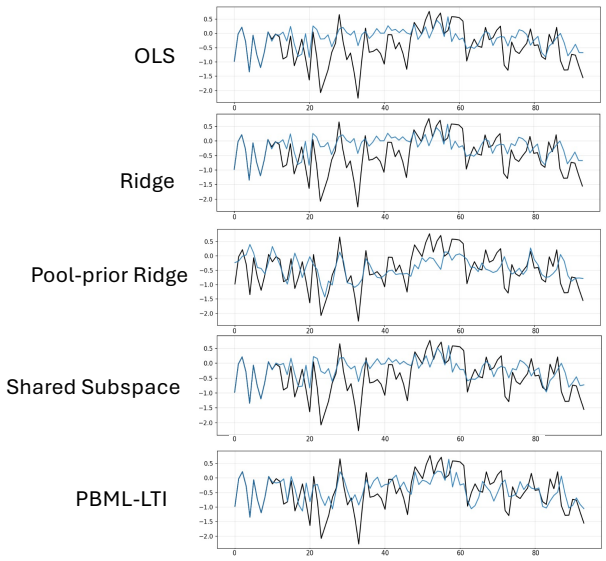

Figure 11: Open-loop rollout trajectories for the system shown in Figure 10 with the same dimensionality. The ground-truth trajectory is shown in black, and the trajectory predicted using the estimated transition matrix is shown in blue.