Yale SCHOOL OF ENGINEERING AND APPLIED SCIENCE

WU TSAI INSTITUTE
YALE UNIVERSITY

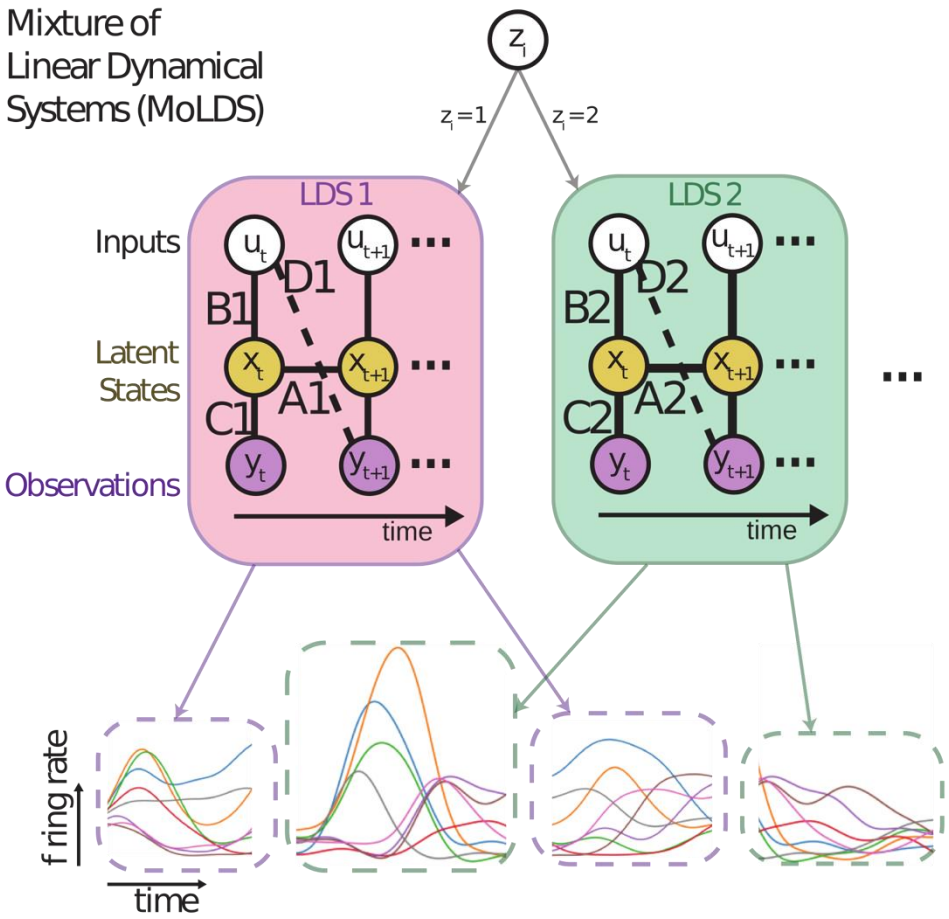
Learning Mixtures of Linear Dynamical Systems via Hybrid Tensor-EM Method

Lulu Gong, Shreya Saxena

Department of Biomedical Engineering
Center for Neurocomputation and Machine Intelligence, Wu Tsai Institute
Yale University

ICLR
2026.04
Rio de Janeiro, Brazil

MoLDS Overview



- **Data**: N input-output trajectories

$$\{u_{i,0:T-1}, y_{i,0:T-1}\}_{i=1}^{N}$$

- **MoLDS model**: each trajectory i is generated by one of K **LDS**

$$\Pr(z_i = k) = p_k, \quad k = 1, ..., K$$

conditioned on $z_i = k$

$$x_{t+1} = A_k x_t + B_k u_t + w_t, \quad w_t \sim \mathcal{N}(0, Q_k),$$

$$y_{t+1} = C_k x_t + D_k u_t + v_t, \quad v_t \sim \mathcal{N}(0, R_k).$$

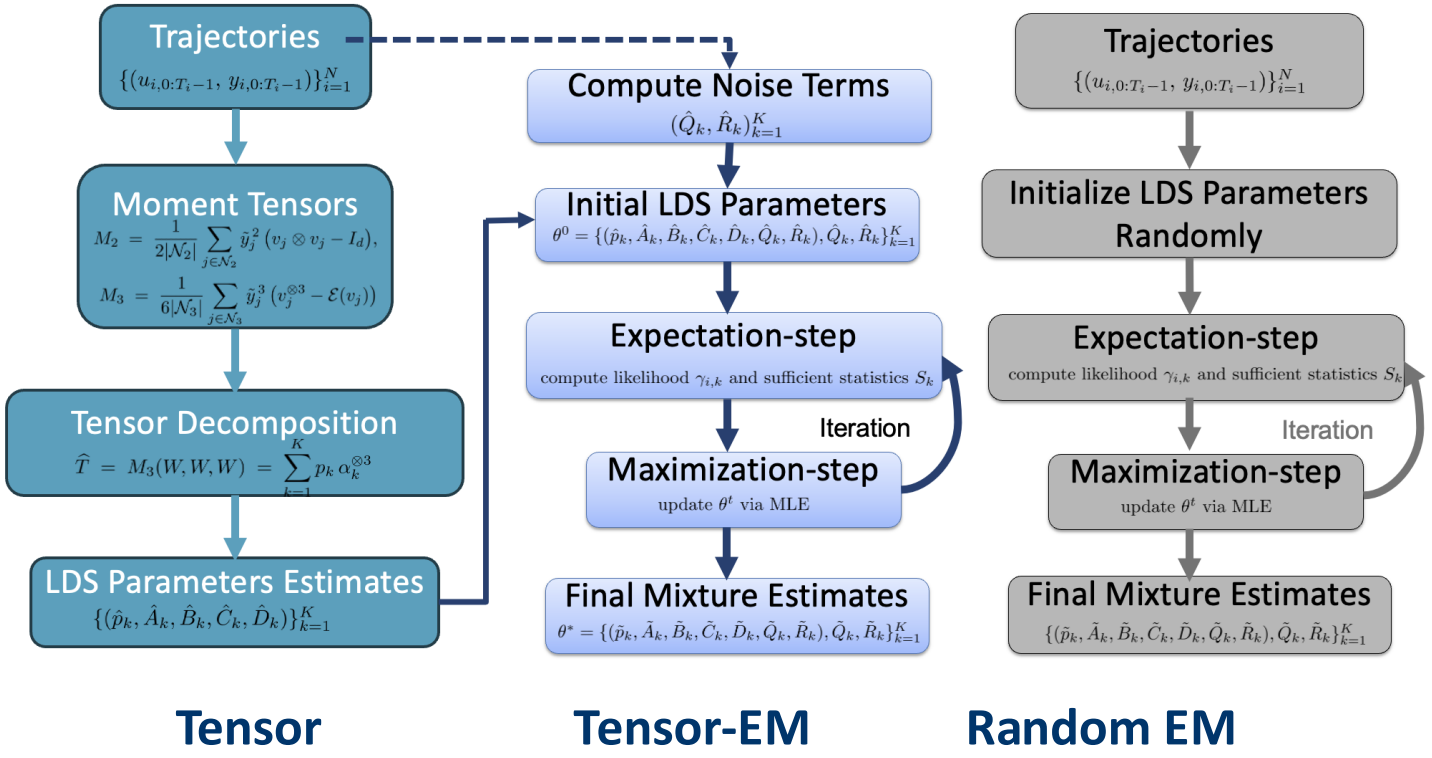
- **Goal:** learn the mixture weights and component system parameters

$$\{p_k, A_k, B_k, C_k, D_k, Q_k, R_k\}_{k=1}^{K}$$

Learning Methods of MoLDS

Different learning methods:

- **Tensor approach:** Decomposition of moment tensors provides globally consistent estimates of mixture weights and system parameters, not accurate in complex and noisy conditions.
- **Random EM approach:** EM initialized with random parameters, iteratively optimizing likelihood but susceptible to local optima.
- **Tensor-EM approach:** Tensor approach provides reliable initialization, followed by Kalman-EM refinement via likelihood optimization.



Tensor-EM for MoLDS

Tensor-EM algorithm: Tensor decomposition provides reliable initialization, followed by Kalman-EM refinement via local optimization of likelihood.

Algorithm 1 Tensor-EM Pipeline for MoLDS

Require: Trajectories $\{(u_{i,0:T_i-1}, y_{i,0:T_i-1})\}_{i=1}^{N}$, truncation length L, LDS order n, #components K

Ensure: Mixture weights $\{\hat{p}_k\}_{k=1}^{K}$ and LDS parameters $\{(\hat{A}_k, \hat{B}_k, \hat{C}_k, \hat{D}_k, \hat{Q}_k, \hat{R}_k)\}_{k=1}^{K}$

Stage 1: Tensor Initialization (Appendix Alg. 3)

Transform MoLDS to MLR via lagged inputs; construct moment tensors M_2, M_3

Apply whitening and SMD to recover mixture weights $\{\hat{p}_k\}$ and Markov parameters

Realize LDS matrices $\{(\hat{A}_k, \hat{B}_k, \hat{C}_k, \hat{D}_k)\}$ via Ho-Kalman algorithm

Initialize noise parameters $\{(\hat{Q}_k^{(0)}, \hat{R}_k^{(0)})\}$ from residual covariances (Appendix D)

Stage 2: EM Refinement (Appendix Alg. 4)

repeat

    E-step: Compute trajectory responsibilities $\gamma_{i,k}$ via Kalman filter likelihoods

    Run Kalman smoother to obtain responsibility-weighted sufficient statistics

    M-step: Update mixture weights and all LDS parameters via closed-form MLE

until convergence in log-likelihood

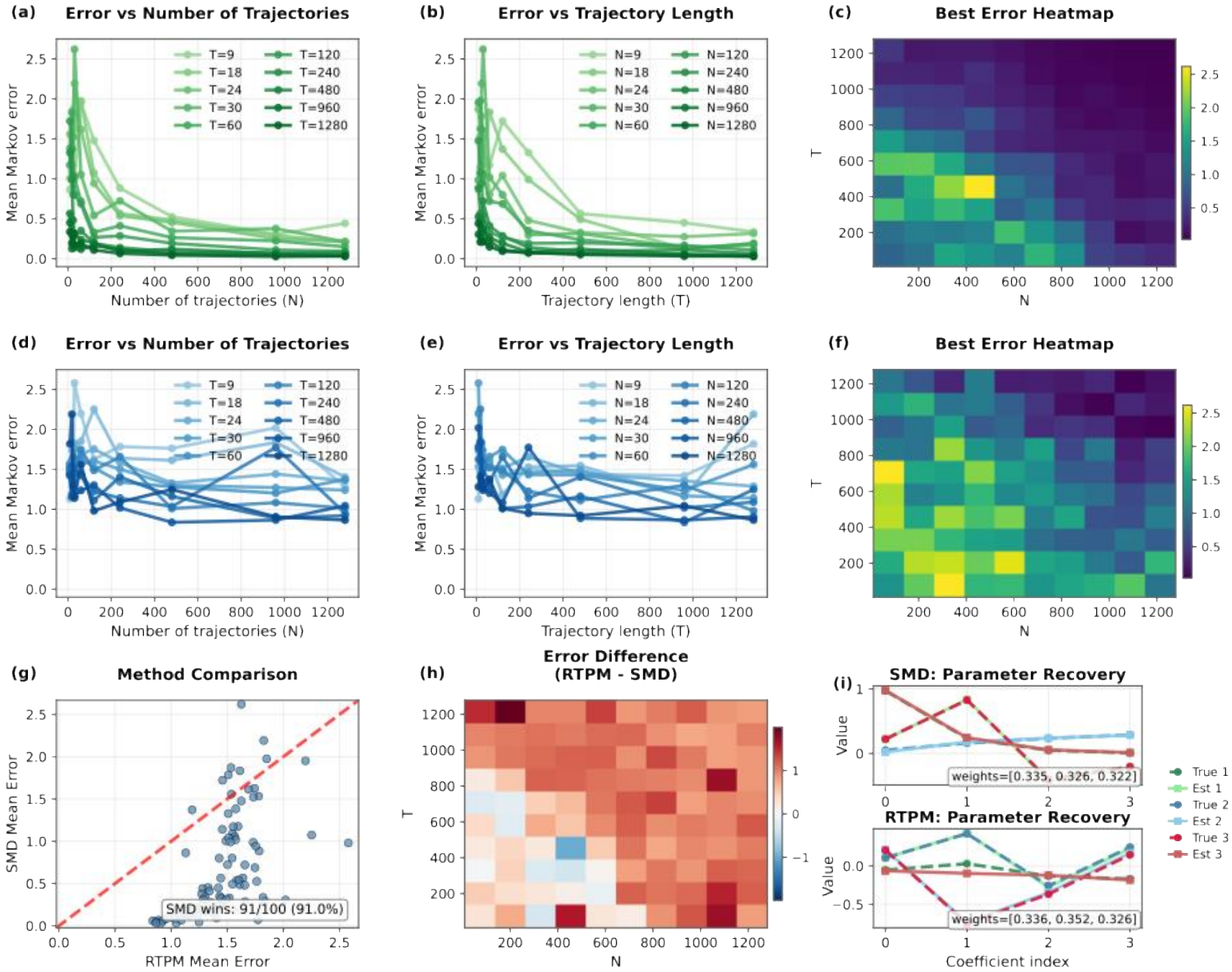
return Refined parameters $\{\hat{p}_k, (\hat{A}_k, \hat{B}_k, \hat{C}_k, \hat{D}_k, \hat{Q}_k, \hat{R}_k)\}_{k=1}^{K}$

Synthetic data I

Tensor-only methods:

1. Setup: K=3, n=3, m=p=1; vary N, T
2. Methods compared: RTPM-tensor, SMD-tensor
3. Evaluation metrics: recovery error of Markov parameters, mixture weights.

- SMD has better empirical performance and enables accurate recovery in low-dim simulated MoLDS

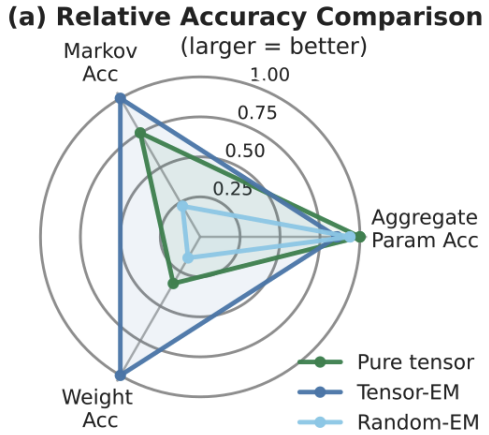


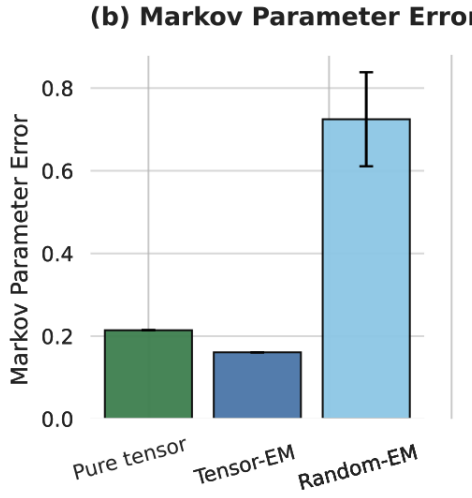
Synthetic data II

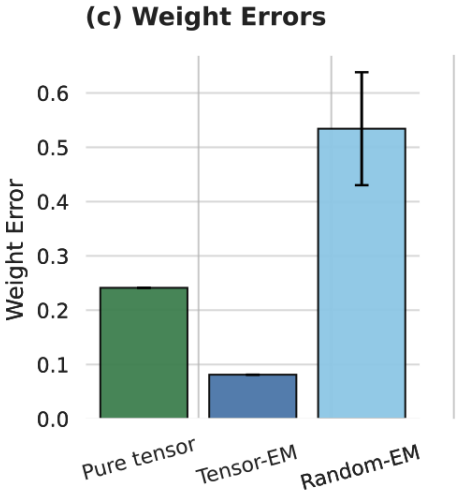
Tensor, Tensor-EM, and Random EM

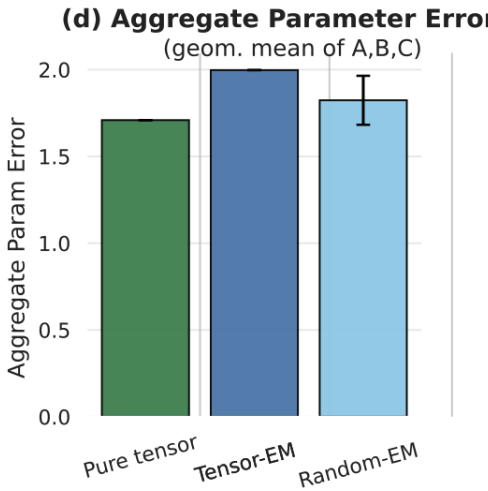
1. Setup: K=6, n=3, m=p=2; N=2000 trajectories with length T=600.
2. Methods compared: pure Tensor, Tensor-EM, and randomly initialized EM.
3. Evaluation metrics: recovery accuracy of Markov parameters, LDS system parameters, and mixture weights.

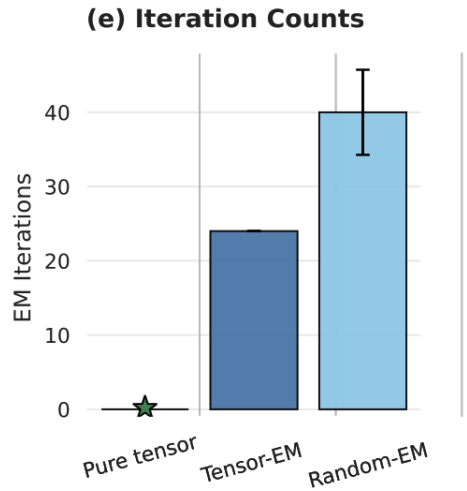
- Tensor-EM achieves more accurate, stable and efficient parameter recovery than pure tensor or randomly initialized EM.

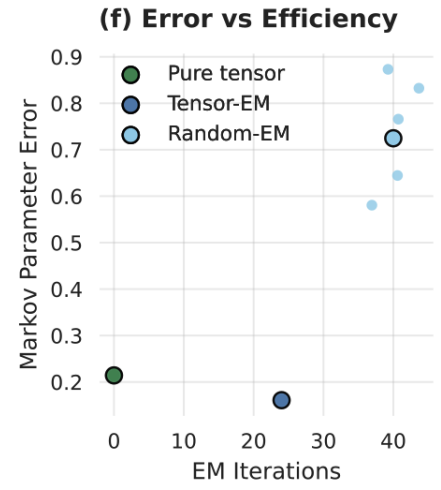








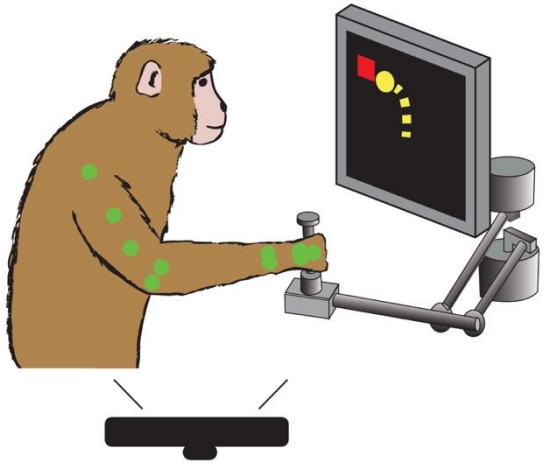


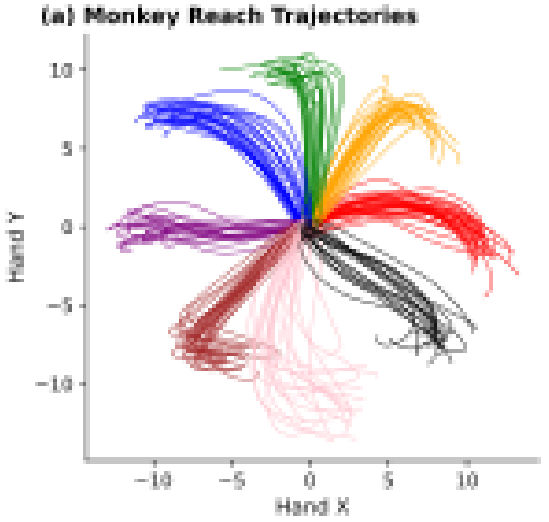


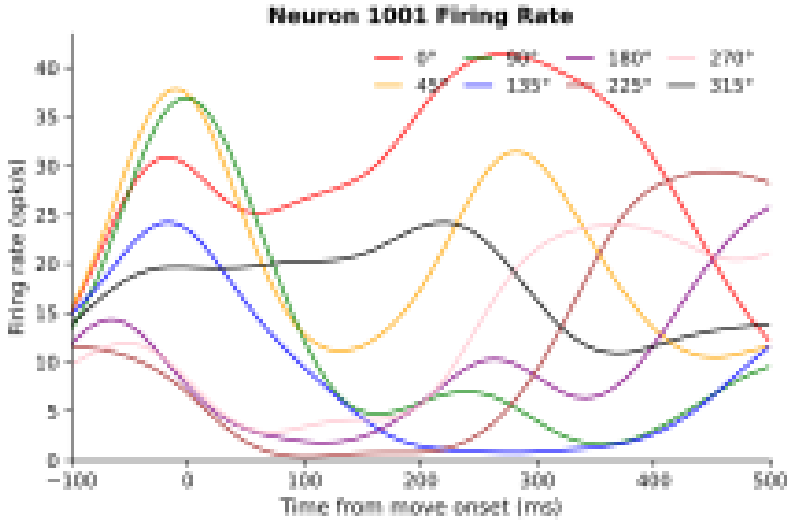
Neural data I - Area2

Area2 dataset:

1. Task: macaque performing center-out reaching movements toward 8 directions; neural activity recorded from somatosensory cortex (Area2).
2. Data: movement-related neural activity; outputs: neural firing rates, inputs: hand velocity.
3. Model selection: models trained using train/validation/test splits; best model selected by BIC, NLL, and RMSE, yielding K=3 components, latent dimension n=5.
4. Analysis: trials clustered by their dominant LDS component, and compared with SLDS, per-direction LDS, and single-trial LDS baselines

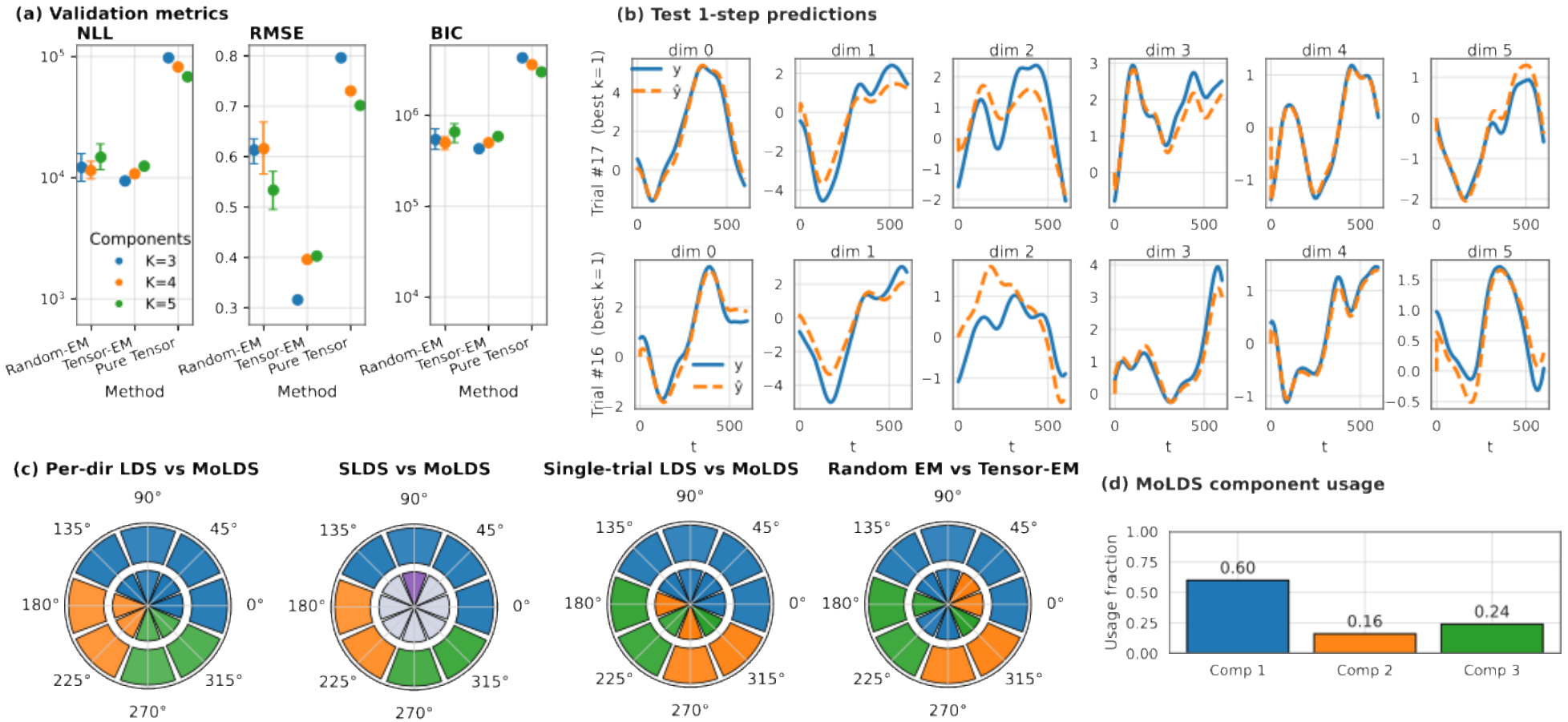




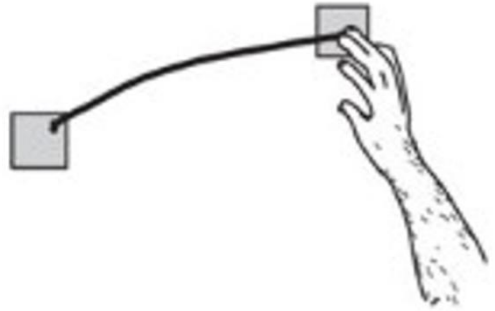


Neural data I - Area2

- Tensor-EM method identifies more meaningful clusters of condition-specific dynamics across trials, outperforming SLDS that struggles to capture cross-trial dynamical heterogeneity.

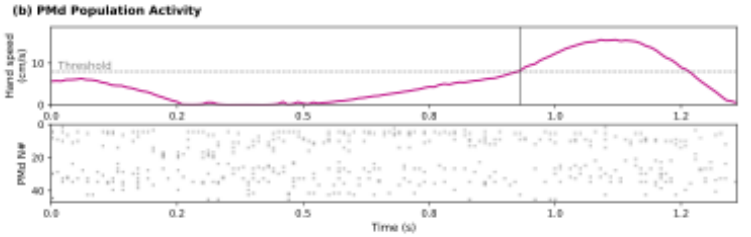


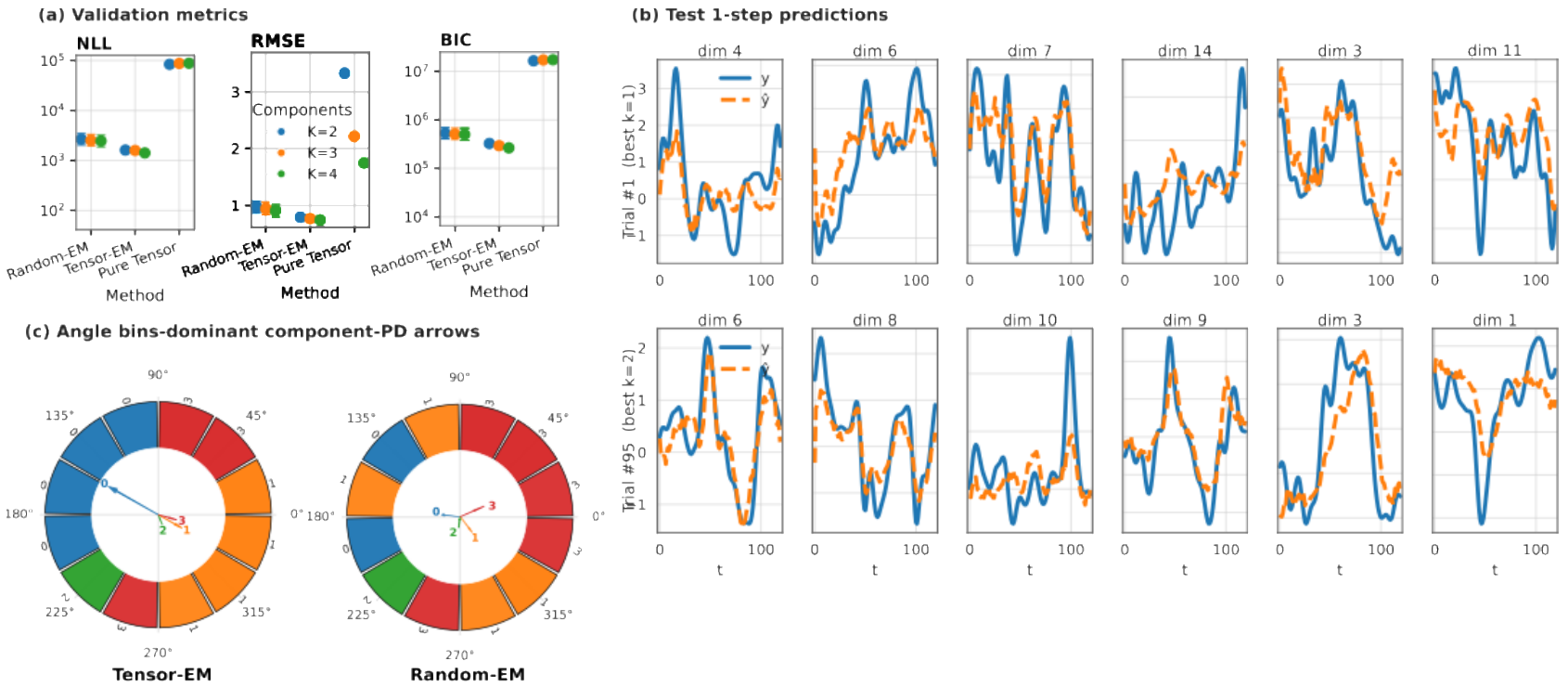
Neural data II - PMd



PMd dataset:

1. Monkey performing sequential reaching movements; neural activity recorded in dorsal premotor cortex (PMd)
2. Best model selected with K=4 components, latent dimension n=5
3. Trials clustered by dominant LDS component





Summary:

- MoLDS provides an interpretable framework for modeling heterogeneous neural population dynamics across trials and conditions.
- Tensor-EM enables stable and efficient learning of MoLDS, combining globally consistent initialization with likelihood-based refinement.
- Applied to two primate neural datasets, the method successfully recovers and clusters condition-specific neural dynamics.

Future work:

1. Linear dynamics → nonlinearity?
2. Tensor-EM is principled but could be intractable→ provable guarantees?
3. Different datasets and tasks …

Thank you!