Sparse Subspace Tracking in High Dimensions

Le Trung Thanh[1,2], Karim Abed-Meraim[1], Adel Hafiane[1], Nguyen Linh Trung[2]

[1] PRISME Laboratory, University of Orleans/INSA-CVL, France

[2]AVITECH Institute, Vietnam National University, Hanoi, Vietnam

May 2022

Contents

Subspace tracking

- Data Model:

$$\mathbf{x}_t = \mathbf{A}\mathbf{w}_t + \mathbf{n}_t, t = 1, 2, \ldots, T. \quad (1)$$

  - T: number of observed samples.
  - $\mathbf{x}_t \in \mathbb{R}^{n\times 1}$: n-dimensional observation vector
  - $\mathbf{A} \in \mathbb{R}^{n\times r}$: basis matrix (deterministic + full rank + maybe time-varying + sparse components)
  - $\mathbf{w}_t \in \mathbb{R}^{r\times 1}$: coefficient vector (random or sparse)
  - $\mathbf{n}_t$: additive random noise (e.g. $\mathcal{N}(0, \sigma_n^2 \mathbf{I}_n)$)

- Estimate $\mathbf{A}$ or span($\mathbf{A}$) upon the arrival of $\mathbf{x}_t$ at each time t

- Two regimes:
  - Classical: $n/T \to 0$ (e.g. fixed n, $T \to \infty$)
  - Low-sample-size and high dimension: $n/T = c > 0$

Reminder

- True covariance matrix

$$\mathbf{\Sigma} = \mathbb{E}\{\mathbf{x}_t \mathbf{x}_t^\top\} = \mathbf{A}\mathbf{R}_x\mathbf{A}^\top + \sigma_n^2 \mathbf{I}_n. \tag{2}$$

- Sample Covariance Matrix (SCM):

$$\mathbf{C}_T = \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^\top = \frac{1}{T}\mathbf{X}\mathbf{X}^\top. \tag{3}$$

- Two regimes

$$\|\mathbf{C}_T - \mathbf{\Sigma}\| \to \begin{cases} 0 & \text{if } n/T \to 0, \text{ (consistent)} \\ \alpha > 0 & \text{if } n/T \nrightarrow 0, \text{ (inconsistent).} \end{cases} \tag{4}$$

$\Rightarrow$ Most of state-of-the-art subspace trackers are inconsistent, including sparse subspace tracking (SST)!

Consistency of Sparse Subspace/PCA Estimation

For sparse subspace/PCA in high dimension regimes

- Regularized Covariance via Thresholding: under mild conditions[1][2]

$$\|\mathbf{C}_T - \mathbf{C}\| \not\to 0, \text{ but } \|\mathcal{T}(\mathbf{C}_T) - \mathbf{C}\| \to 0, \tag{5}$$

  where $\mathcal{T}(.)$ is a threshold operator (soft, hard, or combined).
- Many good thresholding-based batch algorithms since 2010.

$\rightarrow$ Track the subspace of $\mathcal{T}(\mathbf{C}_T)$ instead of $\mathbf{C}_T$.

[1] P. J. Bickel and E. Levina. "Covariance Regularization by Thresholding" *Ann. Stat.* 6 (2008).

[2] Elizaveta Levina and Roman Vershynin. "Partial Estimation of Covariance Matrices" *Probab. Theory Relat. Fields* 3-4 (2012).

State-of-the-art SST algorithms

Algorithm	Method	Initialization	Convergence Guarantee	Computational Complexity	High Dimension Regime ?	Main Limitations
OIST (ITW 2016)	Oja method + soft-thresholding	random	✓	$\mathcal{O}(n)$ for $(r = 1)$	✓	Support only rank-1 subspace
SPCA (ICML 2015)	Row truncation + QR decomposition	batch	✓	$\mathcal{O}(nr \min(r, s \log n))$	✓	Support row-sparsity only
ℓ_1-PAST (IEEE TSP 2016)	PAST + ℓ_1-norm sample matrix inverse	random	✗	$3nr^2 + 3nr + \mathcal{O}(r^2)$	✗	inconsistent
OVBSL (Elsevier SP 2017)	Bayesian inference + ℓ_2/ℓ_1-norm promotion	random	✗	$\mathcal{O}(nr^2 + nr)$	✗	inconsistent
SS/DS-OPAST (EUSIPCO 2017)	OPAST + ℓ_1-norm approximation	random	✗	$3nr^2 + 3nr + \mathcal{O}(r^3)$/ $3nr + \mathcal{O}(nr^2)$	✗	inconsistent
SS/GSS-FAPI (Elsevier SP 2020)	FAPI + Givens rotations	random	✓	$2nr^2 + 4nr + \mathcal{O}(r^2)$/ $4nr + 4ns + \mathcal{O}(r^2)$	✗	inconsistent

OPIT: Online Power Iteration by Thresholding

- Power Iteration (PI) Method: At ℓ-th iteration
  - Step 1: $\mathbf{S}_\ell \leftarrow \mathbf{C}_t\mathbf{U}_{\ell-1}$
  - Step 2: $\mathbf{U}_\ell \leftarrow \mathrm{QR}(\mathbf{S}_\ell)$

- OPIT: $\ell \leftarrow t$
  - Facts: $\mathbf{R}_t = \mathbf{R}_{t-1} + \mathbf{x}(t)\mathbf{x}(t)^\top$, $\mathbf{C}_t = t^{-1}\mathbf{R}_t$ and $\mathrm{QR}(\mathbf{R}_t\mathbf{U}) = \mathrm{QR}(\mathbf{C}_t\mathbf{U})$
  - Step 1: $\mathbf{S}_t \leftarrow \lambda\mathbf{S}_{t-1}\mathbf{E}_{t-1} + \mathbf{x}_t\mathbf{z}_t^\top$, where
    - $\mathbf{z}_t = \mathbf{U}_{t-1}^\top\mathbf{x}_t$ and $\mathbf{E}_t = \mathbf{U}_{t-2}^\top\mathbf{U}_{t-1}$
    - $0 < \lambda \leq 1$: Forgetting factor
  - Step 2:
    - $\hat{\mathbf{S}}_t \leftarrow \tau(\mathbf{S}_t, k)$ // [keep the k strongest elements in each column of $\mathbf{S}_t$]
    - $\mathbf{U}_t \leftarrow \mathrm{QR}(\hat{\mathbf{S}}_t)$ or $\hat{\mathbf{S}}_t/\|\hat{\mathbf{S}}_t\|_2$

- Computational Complexity: $\mathcal{O}(nr^2)$

Theoretical Analysis: Assumptions

- (A1) $\mathbf{A}_t = \mathbf{A}$, $\lambda = 1$
- (A2) $\mathbf{A} = \mathbf{U} \circledast \mathbf{\Omega}$ where
  - $\mathbf{U} \in \mathcal{U} \triangleq \{\mathbf{U} \in \mathbb{R}^{n \times r}, \|\mathbf{u}_k\|_2 \leq 1, 1 \leq \kappa(\mathbf{U}) < \infty\}$.
  - $\mathbf{\Omega}$: $\omega_{i,j}$ is an i.i.d. Bernoulli variable with probability $1 - \rho$
  - $\rho \geq 1 - \sqrt{(\log n)/n}$
- (A3): For all t,
  - $\|\mathbf{x}_t\|_2^2 < \infty$
  - $\mathbb{E}\{\|\boldsymbol{\ell}_t\|_2^2\} = \sigma_x^2$, $\mathbb{E}\{\|\mathbf{n}_t\|_2^2\} = \sigma_n^2$, $\sigma_n < \sigma_x$
  - $\mathbf{w}_t \in \mathcal{W} \triangleq \{\mathbf{w} \in \mathbb{R}^{r \times 1}, 0 < |\mathbf{w}(i)| < \infty\}$

Theoretical Analysis: Main Result

Lemma

Suppose that assumptions (A1)-(A3) are met, the true basis $\mathbf{A}$ *is deterministic and unchanged over time, and that the initialization matrix* $\mathbf{U}_0$ *and the number of data samples satisfy the following conditions*

$$t \geq \frac{c_\delta}{W\epsilon^2}\left(\sqrt{r} + (2\sigma_n/\sigma_x + \sigma_n^2/\sigma_x^2)\sqrt{n}\right)^2, \tag{6}$$

$$\tan\theta\big(\mathbf{A}, \mathbf{U}_0\big) \leq \frac{\sigma_x^2 + \sigma_n^2}{(1+\sqrt{r}(1+\sqrt{2}))\sigma_x^2 - (2+\sqrt{2})\sigma_n^2}, \tag{7}$$

with a small predefined error ϵ *and a positive number* $c_\delta = C\sqrt{\log(2/\delta)}$ *where* $0 < \delta \ll 1$ *and* C *is a universal positive number. If* $\mathbf{U}_\tau$ *is generated by OPIT at time* t*, then*

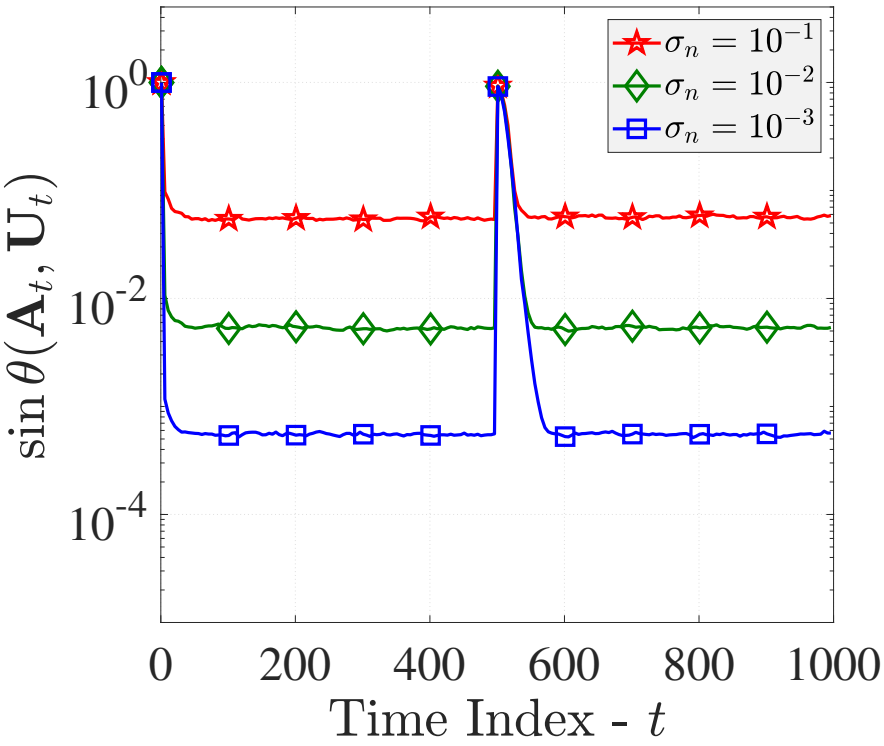
$$\sin\theta(\mathbf{A}, \mathbf{U}_t) \leq \epsilon, \tag{8}$$

with a probability at least $1 - \delta$*.*

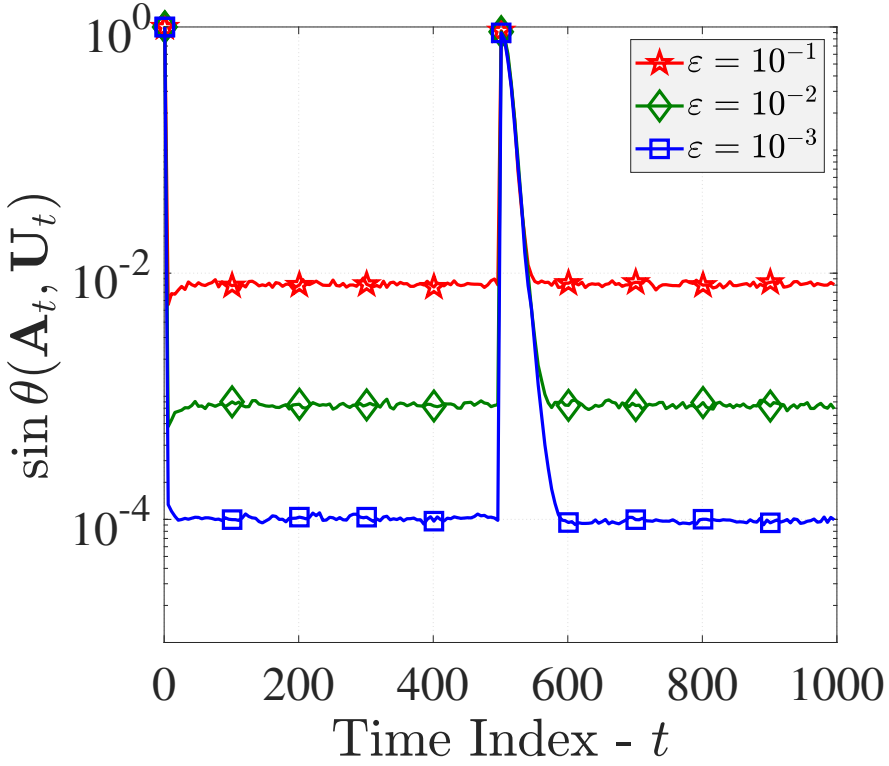
Experiment Setup

- Data model: $\mathbf{x}_t = \mathbf{A}_t \mathbf{w}_t + \sigma_n \mathbf{n}_t$
  - $\mathbf{n}_t \in \mathcal{N}(0, \mathbf{I}_n)$, $\sigma_n > 0$ to control the noise level
  - $\mathbf{w}_t \in \mathcal{N}(0, \mathbf{I}_r)$
  - $\mathbf{A}_t = \mathbf{\Omega} \circledast (\mathbf{A}_{t-1} + \varepsilon \mathbf{N}_t) \in \mathbb{R}^{n \times r}$ where
    - $\mathbf{\Omega}$: Bernoulli random matrix with probability $1 - \rho$
    - $\mathbf{N}_t$: Normalized Gaussian white noise matrix
    - $\varepsilon > 0$ to control the time variation
- Evaluation metric: $d(\mathbf{A}_t, \mathbf{U}_t) = \sin(\mathbf{A}_t, \mathbf{U}_t)$

Effect of noise and time-varying factors



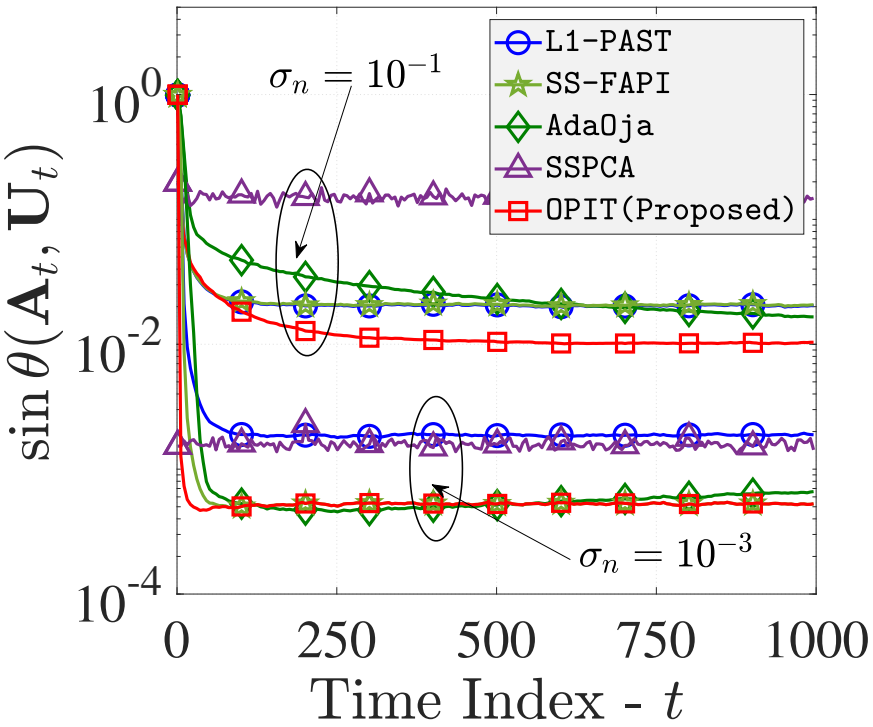
(a) Effect of the noise level σ_n



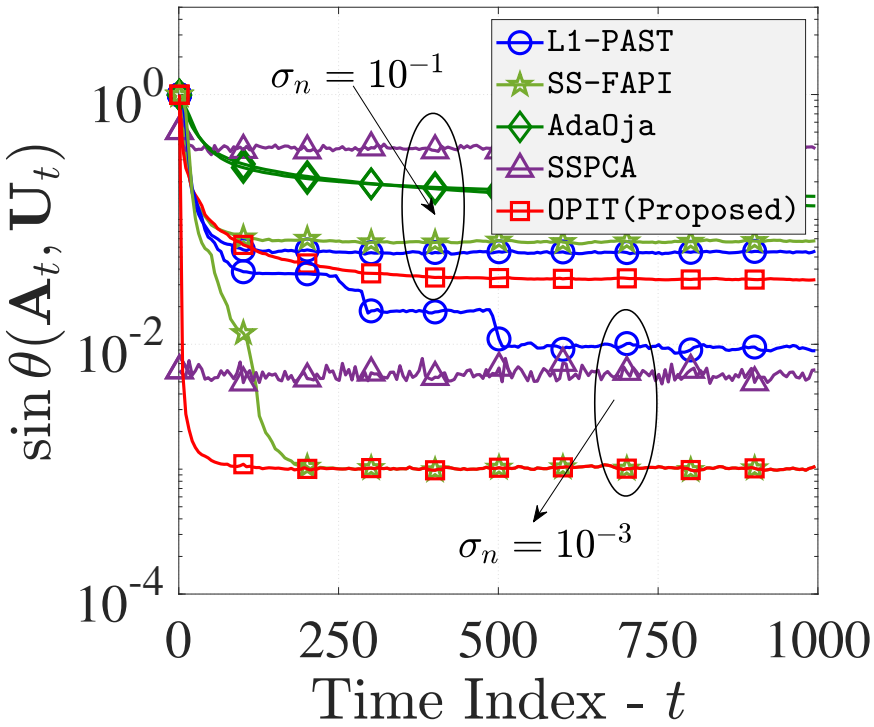
(b) Effect of the time-varying factor ε

Figure: n = 100, r = 5, sparsity level $\rho = 90\%$, forgetting factor $\lambda = 0.9$.

Classical regime



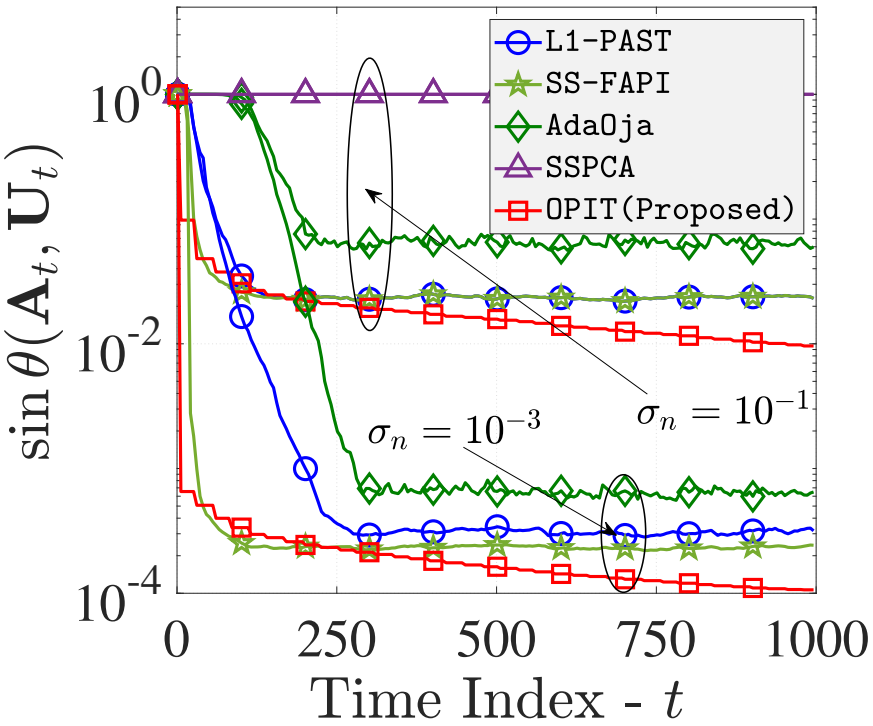
(a) $n = 50,\ r = 2,\ \rho = 50\%$



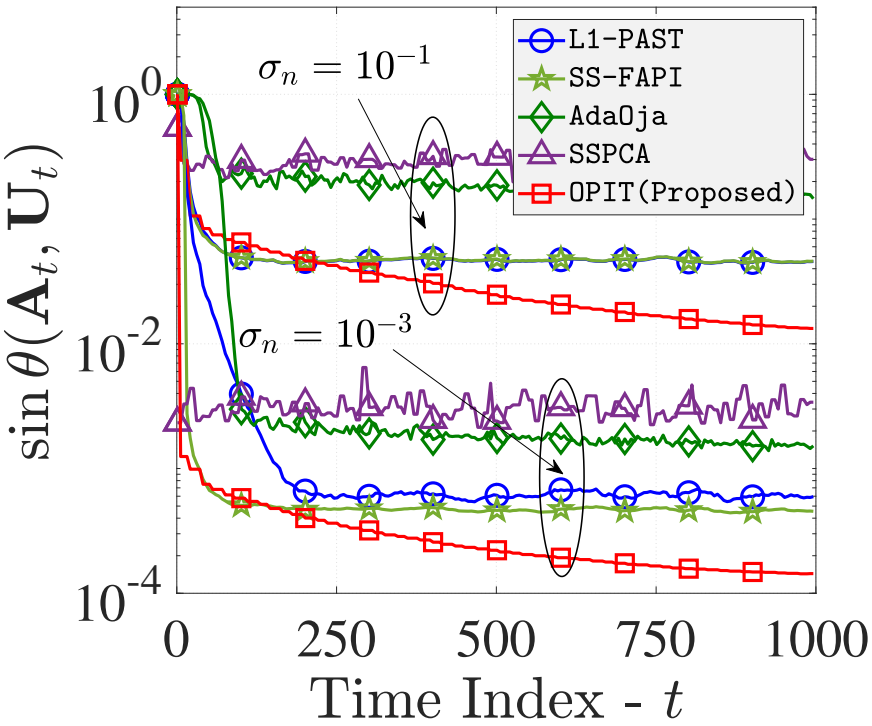
(b) $n = 50,\ r = 2,\ \rho = 90\%$

Figure: Time-varying environments: data samples $T = 1000$ and time-varying factor $\varepsilon = 10^{-3}$.

High-dimensional regime



(a) $n = 10000$, $r = 10$, $\rho = 50\%$



(b) $n = 10000$, $r = 10$, $\rho = 90\%$

Figure: Time-varying environments: data samples $T = 1000$ and time-varying factor $\varepsilon = 10^{-3}$.

Video Tracking

a) Lobby

b) Hall

Figure: Two video sequences used in this paper.

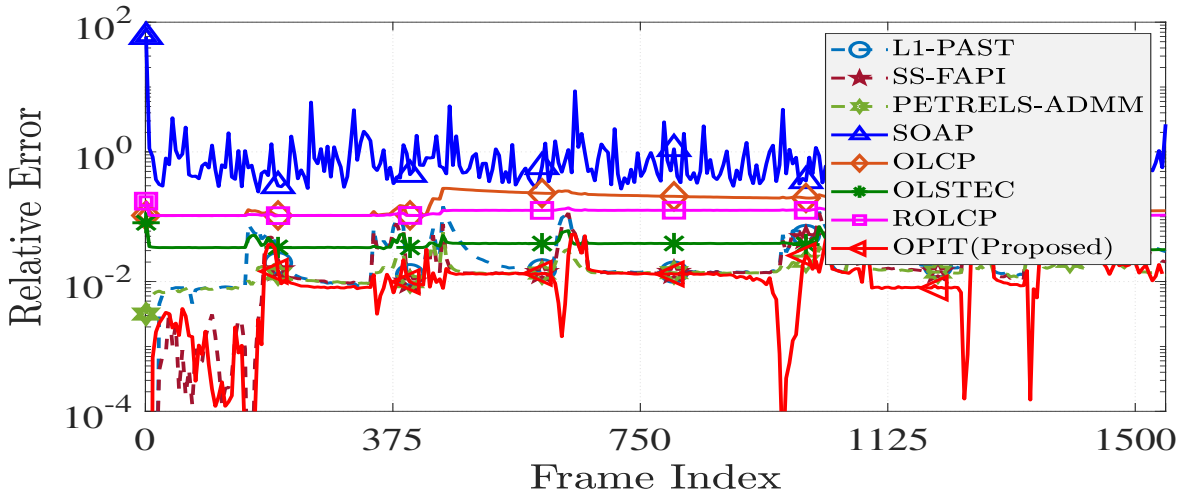


Figure: Tracking ability of algorithms on the "Lobby" data.

Video Tracking (Cont.)

Dataset	"Lobby"		"Hall"	
Tensor size	$128 \times 160 \times 1546$		$174 \times 144 \times 3584$	
Matrix size	20480×1546		25056×3584	
Evaluation metrics	time(s)	error	time(s)	error
SOAP	14.29	0.842	21.72	0.989
OLCP	10.50	0.161	19.98	0.154
OLSTEC	44.25	0.037	92.82	0.041
ROLCP	4.32	0.114	10.74	0.120
PETRELS-ADMM	118.4	0.015	305.5	0.018
ℓ_1-PAST	14.11	0.031	33.73	0.101
SS-FAPI	12.99	0.023	32.72	0.100
OPIT ($W = 1$)	16.32	0.013	50.78	0.056
OPIT ($W = \lfloor \log(IJ) \rceil$)	1.89	0.021	5.62	0.086

Table: Runtime and averaged relative error of adaptive algorithms on tracking the four video sequences.

Conclusions

- Proposed a novel sparse subspace tracking algorithm called OPIT.
- Provide a theoretical result on convergence for OPIT.
- Demonstrated the effectiveness of OPIT with different experiments

Thank you for listening !