Correlative Information Maximization Based Biologically Plausible Neural Networks for Correlated Source Separation

ICLR 2023

Bariscan Bozkurt[1,2]   Ates Isfendiyaroglu[3]   Cengiz Pehlevan[4]   Alper T. Erdogan[1,2]

[1] KUIS AI Center, Koc University, Istanbul, Turkey
[2] Electrical and Electronics Engineering Department, Koc University, Istanbul, Turkey
[3] Uskudar American Academy
[4] John A. Paulson School of Engineering & Applied Sciences and Center for Brain Science, Harvard University, Cambridge, MA 02138, USA

Goal

We introduce a Biologically Plausible Neural Network Framework exploiting information maximization criterion

Goal

We introduce a Biologically Plausible Neural Network Framework exploiting information maximization criterion

- Capable to blindly extract the correlated latent causes

Goal

We introduce a Biologically Plausible Neural Network Framework exploiting information maximization criterion

- Capable to blindly extract the correlated latent causes
- Applicable to a diverse geometric assumptions

Outline

- Blind Source Separation (BSS) Setup

Outline

- Blind Source Separation (BSS) Setup
- Definition of Correlative Information (CorInfo)

Outline

- Blind Source Separation (BSS) Setup
- Definition of Correlative Information (CorInfo)
- Correlative Information Maximization (CorInfoMax) based Blind Source Separation

Outline

- Blind Source Separation (BSS) Setup
- Definition of Correlative Information (CorInfo)
- Correlative Information Maximization (CorInfoMax) based Blind Source Separation
- Online CorInfoMax Formulation ➡ Biologically Plausible Neural Networks

Outline

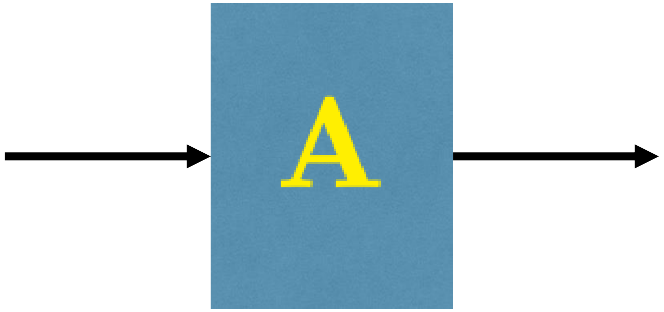
- Blind Source Separation (BSS) Setup
- Definition of Correlative Information (CorInfo)
- Correlative Information Maximization (CorInfoMax) based Blind Source Separation
- Online CorInfoMax Formulation ➡ Biologically Plausible Neural Networks
- Numerical Experiments and Conclusion

Blind Source Separation Setup

$$\underbrace{\begin{bmatrix} | & \cdots & | \\ \mathbf{x}(1) & \cdots & \mathbf{x}(N) \\ | & \cdots & | \end{bmatrix}}_{\mathbf{X}\,\text{(observations)}} = \underbrace{\begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix}}_{\mathbf{A}\,\text{(linear transf.)}} \underbrace{\begin{bmatrix} | & \cdots & | \\ \mathbf{s}(1) & \cdots & \mathbf{s}(N) \\ | & \cdots & | \end{bmatrix}}_{\mathbf{S}\,\text{(latent vectors)}}$$

Sources

$\mathbf{s}(t)$

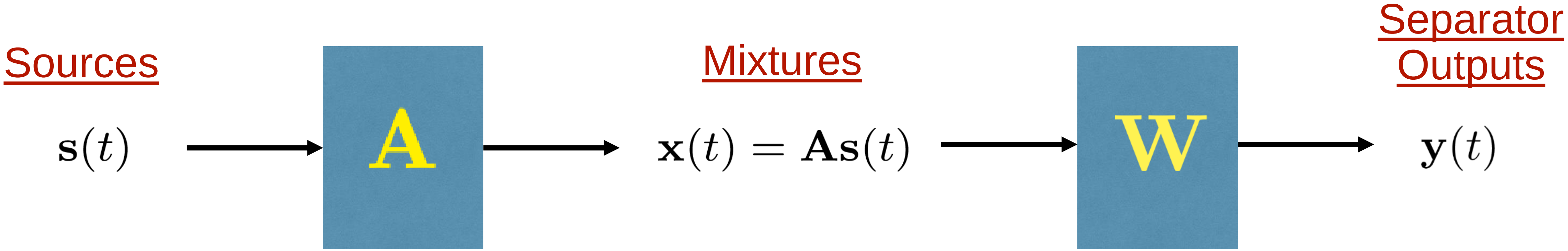


Mixtures

$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$

Blind Source Separation Setup

$$\underbrace{\begin{bmatrix} | & \cdots & | \\ \mathbf{x}(1) & \ldots & \mathbf{x}(N) \\ | & \cdots & | \end{bmatrix}}_{\mathbf{X}\,(\text{observations})} = \underbrace{\begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix}}_{\mathbf{A}\,(\text{linear transf.})} \underbrace{\begin{bmatrix} | & \cdots & | \\ \mathbf{s}(1) & \ldots & \mathbf{s}(N) \\ | & \cdots & | \end{bmatrix}}_{\mathbf{S}\,(\text{ latent vectors})}$$

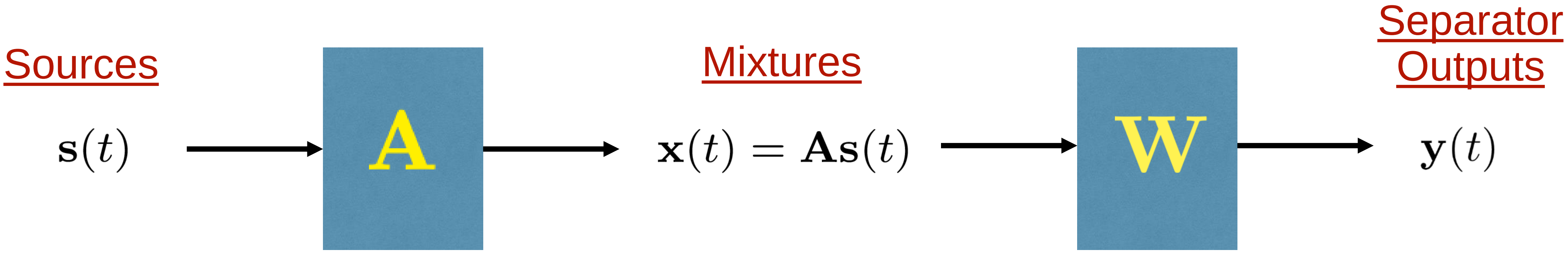


Independent Component Analysis (ICA)

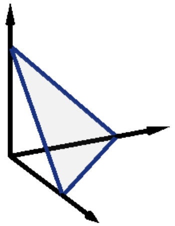
$$\begin{bmatrix} | & \cdots & | \\ \mathbf{x}(1) & \cdots & \mathbf{x}(N) \\ | & \cdots & | \end{bmatrix} = \begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix} \begin{bmatrix} | & \cdots & | \\ \mathbf{s}(1) & \cdots & \mathbf{s}(N) \\ | & \cdots & | \end{bmatrix}$$

The columns are drawn from a separable pdf.

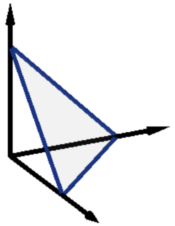
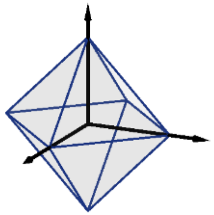
<u>Mutual Independence!</u>



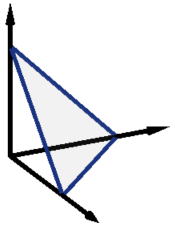
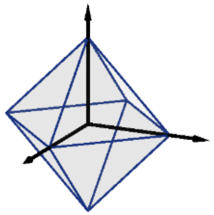
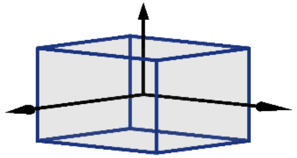
Blind Source Separation Frameworks

Source Domain	Framework
$\mathbf{s}(t) \in \mathcal{P} \subseteq \mathbb{R}^n$	
 Δ – Unit Simplex	**Nonnegative Matrix Factorization (NMF)**

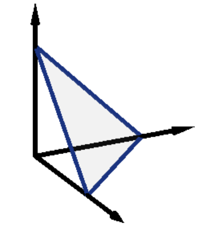
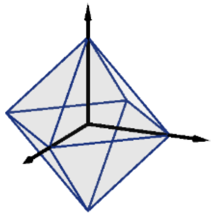
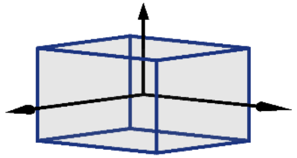
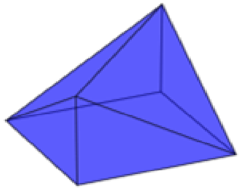
Blind Source Separation Frameworks

Source Domain $\mathbf{s}(t) \in \mathcal{P} \subseteq \mathbb{R}^n$		Framework
	Δ – Unit Simplex	**Nonnegative Matrix Factorization (NMF)**
	ℓ_1 – Norm Ball	**Sparse Component Analysis (SCA)**

Blind Source Separation Frameworks

Source Domain	Framework
$\mathbf{s}(t) \in \mathcal{P} \subseteq \mathbb{R}^n$	
 Δ – Unit Simplex	**Nonnegative Matrix Factorization (NMF)**
 ℓ_1 – Norm Ball	**Sparse Component Analysis (SCA)**
 ℓ_∞ – Norm Ball	**Bounded Component Analysis (BCA)**

Blind Source Separation Frameworks

Source Domain		Framework
$\mathbf{s}(t) \in \mathcal{P} \subseteq \mathbb{R}^n$		
	Δ – Unit Simplex	**Nonnegative Matrix Factorization (NMF)**
	ℓ_1 – Norm Ball	**Sparse Component Analysis (SCA)**
	ℓ_∞ – Norm Ball	**Bounded Component Analysis (BCA)**
	"Identifiable" Polytope	**Polytopic Matrix Factorization (PMF)**

Correlative (Mutual) Information

For finite two sets of vectors (N vectors) in a Euclidean Space,

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}(1) & \boldsymbol{x}(2) & \dots & \boldsymbol{x}(N) \end{bmatrix} \in \mathbb{R}^{m \times N}, \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}(1) & \boldsymbol{y}(2) & \dots & \boldsymbol{y}(N) \end{bmatrix} \in \mathbb{R}^{n \times N}$$

Correlative (Mutual) Information

For finite two sets of vectors (N vectors) in a Euclidean Space,

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}(1) & \boldsymbol{x}(2) & \ldots & \boldsymbol{x}(N) \end{bmatrix} \in \mathbb{R}^{m \times N}, \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}(1) & \boldsymbol{y}(2) & \ldots & \boldsymbol{y}(N) \end{bmatrix} \in \mathbb{R}^{n \times N}$$

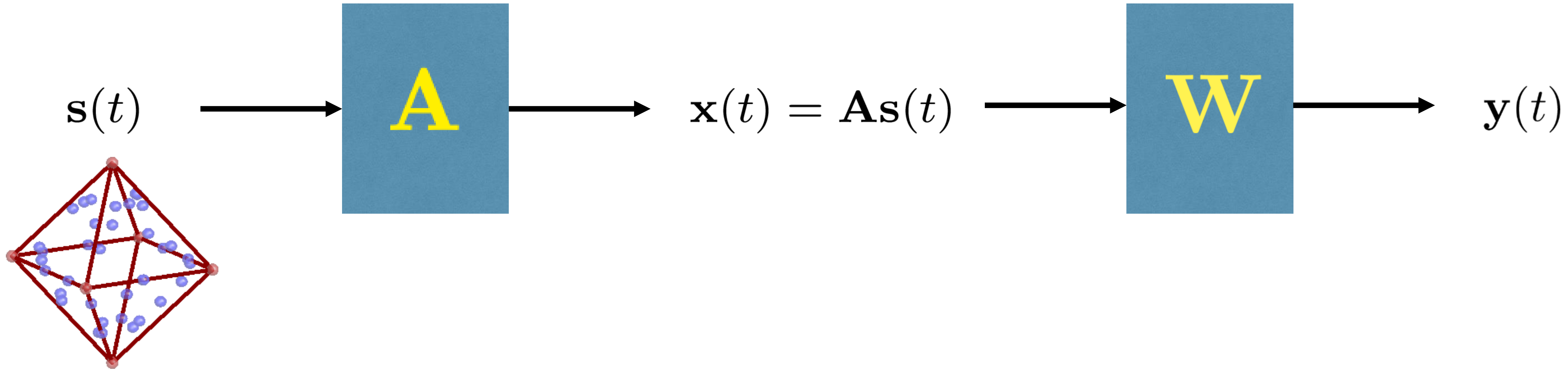
the deterministic correlative mutual information is defined as,

$$\begin{aligned} I^{(\epsilon)}(\boldsymbol{X}, \boldsymbol{Y}) &= H_{\mathrm{LD}}^{(\epsilon)}(\boldsymbol{Y}) - H_{\mathrm{LD}}^{(\epsilon)}(\boldsymbol{Y} |_L \boldsymbol{X}) \\ &= \frac{1}{2} \log\det(\hat{\boldsymbol{R}}_{\boldsymbol{y}} + \epsilon \boldsymbol{I}) - \frac{1}{2} \log\det(\hat{\boldsymbol{R}}_{\boldsymbol{y}} - \hat{\boldsymbol{R}}_{\boldsymbol{xy}}^T (\hat{\boldsymbol{R}}_{\boldsymbol{x}} + \epsilon \boldsymbol{I})^{-1} \hat{\boldsymbol{R}}_{\boldsymbol{xy}} + \epsilon \boldsymbol{I}) \end{aligned}$$

Correlative (Mutual) Information Maximization for BSS

Erdogan (2022) proposes maximizing the correlative information flow from the mixtures to the separator outputs while the outputs are restricted to presumed source domain,

$$\begin{aligned} \underset{\boldsymbol{Y} \in \mathbb{R}^{n \times N}}{\text{maximize}} \quad & I_{\text{LD}}^{(\epsilon)}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2} \log\det(\hat{\boldsymbol{R}}_{\boldsymbol{y}} + \epsilon \boldsymbol{I}) - \frac{1}{2} \log\det(\hat{\boldsymbol{R}}_{\boldsymbol{e}} + \epsilon \boldsymbol{I}) \\ \text{subject to} \quad & \boldsymbol{Y}_{:,i} \in \mathcal{P}, i = 1, \ldots, N, \end{aligned}$$



Online CorInfoMax Optimization

The weighted output and error sample autocorrelation matrices are defined as:

$$\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{\zeta_{\boldsymbol{y}}}(k) = \frac{1-\zeta_{\boldsymbol{y}}}{1-\zeta_{\boldsymbol{y}}^k} \sum_{i=1}^{k} \zeta_{\boldsymbol{y}}^{k-i} \boldsymbol{y}(i)\boldsymbol{y}(i)^T \qquad \hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\zeta_{\boldsymbol{e}}}(k) = \frac{1-\zeta_{\boldsymbol{e}}}{1-\zeta_{\boldsymbol{e}}^k} \sum_{i=1}^{k} \zeta_{\boldsymbol{e}}^{k-i} \boldsymbol{e}(i)\boldsymbol{e}(i)^T$$

Online CorInfoMax Optimization

The weighted output and error sample autocorrelation matrices are defined as:

$$\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{\zeta_{\boldsymbol{y}}}(k) = \frac{1-\zeta_{\boldsymbol{y}}}{1-\zeta_{\boldsymbol{y}}^{k}} \sum_{i=1}^{k} \zeta_{\boldsymbol{y}}^{k-i} \boldsymbol{y}(i)\boldsymbol{y}(i)^T \qquad \hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\zeta_{\boldsymbol{e}}}(k) = \frac{1-\zeta_{\boldsymbol{e}}}{1-\zeta_{\boldsymbol{e}}^{k}} \sum_{i=1}^{k} \zeta_{\boldsymbol{e}}^{k-i} \boldsymbol{e}(i)\boldsymbol{e}(i)^T$$

The online CorInfoMax optimization can be posed as the following:

$$\begin{array}{ll} \underset{\boldsymbol{y}(k)\in\mathbb{R}^n}{\text{maximize}} & \mathcal{J}(\boldsymbol{y}(k)) = \frac{1}{2}\log\det(\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{\zeta_{\boldsymbol{y}}}(k) + \epsilon\boldsymbol{I}) - \frac{1}{2}\log\det(\hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\zeta_{\boldsymbol{e}}}(k) + \epsilon\boldsymbol{I}) \\ \text{subject to} & \boldsymbol{y}(k) \in \mathcal{P}. \end{array}$$

Online CorInfoMax Optimization

The weighted output and error sample autocorrelation matrices are defined as:

$$\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{\zeta_{\boldsymbol{y}}}(k) = \frac{1-\zeta_{\boldsymbol{y}}}{1-\zeta_{\boldsymbol{y}}^k}\sum_{i=1}^{k}\zeta_{\boldsymbol{y}}^{k-i}\boldsymbol{y}(i)\boldsymbol{y}(i)^T \qquad \hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\zeta_{\boldsymbol{e}}}(k) = \frac{1-\zeta_{\boldsymbol{e}}}{1-\zeta_{\boldsymbol{e}}^k}\sum_{i=1}^{k}\zeta_{\boldsymbol{e}}^{k-i}\boldsymbol{e}(i)\boldsymbol{e}(i)^T$$

The online CorInfoMax optimization can be posed as the following:

$$\begin{array}{ll}\underset{\boldsymbol{y}(k)\in\mathbb{R}^n}{\text{maximize}} & \mathcal{J}(\boldsymbol{y}(k)) = \frac{1}{2}\log\det(\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{\zeta_{\boldsymbol{y}}}(k)+\epsilon\boldsymbol{I}) - \frac{1}{2}\log\det(\hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\zeta_{\boldsymbol{e}}}(k)+\epsilon\boldsymbol{I}) \\ \text{subject to} & \boldsymbol{y}(k)\in\mathcal{P}.\end{array}$$

The online regularized least square problem for the best linear MMSE matrix from mixtures to source estimates:

$$\underset{\boldsymbol{W}(k)\in\mathbb{R}^{n\times m}}{\text{minimize}} \quad \mu_{\boldsymbol{W}}\|\boldsymbol{y}(k)-\boldsymbol{W}(k)\boldsymbol{x}(k)\|_2^2 + \|\boldsymbol{W}(k)-\boldsymbol{W}(k-1)\|_F^2.$$

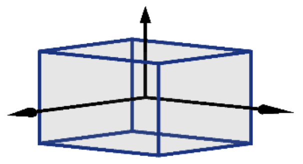
Anti-sparse CorInfoMax Network

Output Updates

$$\boldsymbol{e}(k;\nu) = \boldsymbol{y}(k;\nu) - \boldsymbol{W}(k)\boldsymbol{x}(k),$$

$$\nabla_{\boldsymbol{y}(k)}\mathcal{J}(\boldsymbol{y}(k;\nu)) = \gamma_{\boldsymbol{y}}\boldsymbol{B}_{\boldsymbol{y}}^{\zeta_{\boldsymbol{y}}}(k)\boldsymbol{y}(k;\nu) - \gamma_{\boldsymbol{e}}\boldsymbol{B}_{\boldsymbol{e}}^{\zeta_{\boldsymbol{e}}}(k)\boldsymbol{e}(k;\nu),$$

$$\boldsymbol{y}(k;\nu+1) = \sigma_1\left(\boldsymbol{y}(k;\nu) + \eta_{\boldsymbol{y}}(\nu)\nabla_{\boldsymbol{y}(k)}\mathcal{J}(\boldsymbol{y}(k;\nu))\right),$$



ℓ_∞ – Norm Ball

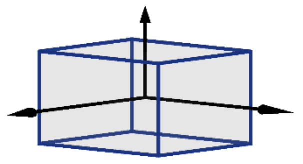
Anti-sparse CorInfoMax Network

Output Updates

$$\boldsymbol{e}(k;\nu) = \boldsymbol{y}(k;\nu) - \boldsymbol{W}(k)\boldsymbol{x}(k),$$

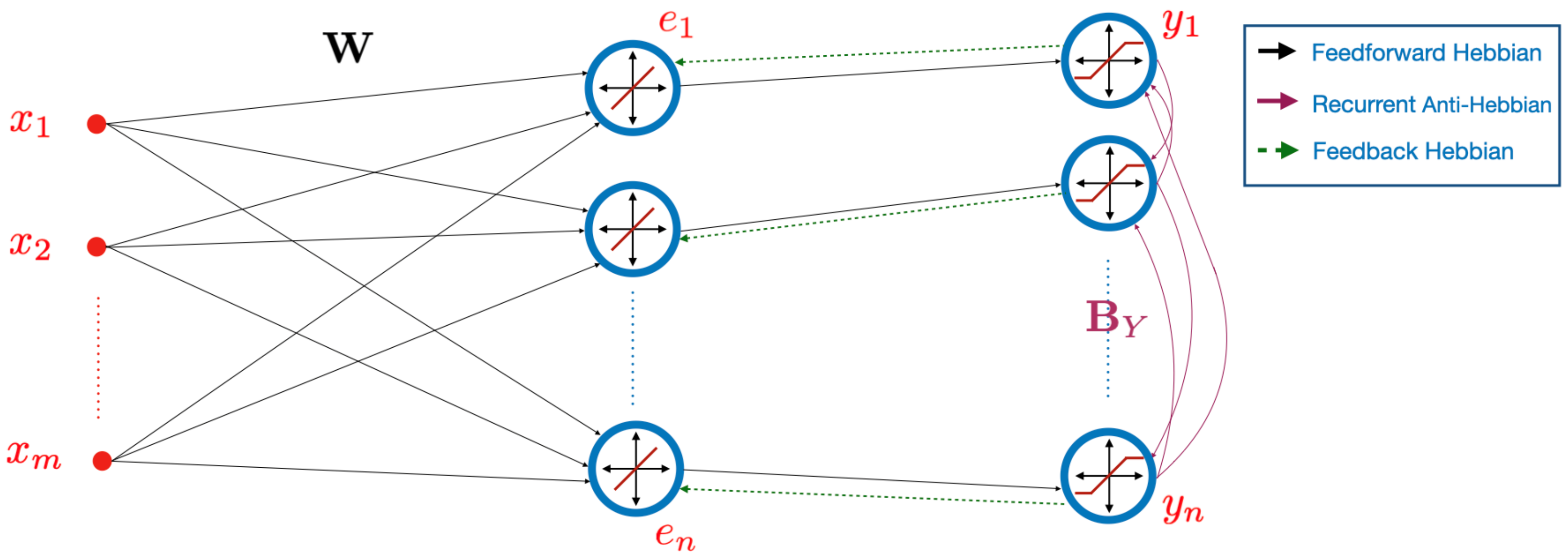
$$\nabla_{\boldsymbol{y}(k)}\mathcal{J}(\boldsymbol{y}(k;\nu)) = \gamma_{\boldsymbol{y}}\boldsymbol{B}_{\boldsymbol{y}}^{\zeta_{\boldsymbol{y}}}(k)\boldsymbol{y}(k;\nu) - \gamma_{\boldsymbol{e}}\boldsymbol{B}_{\boldsymbol{e}}^{\zeta_{\boldsymbol{e}}}(k)\boldsymbol{e}(k;\nu),$$

$$\boldsymbol{y}(k;\nu+1) = \sigma_1\left(\boldsymbol{y}(k;\nu) + \eta_{\boldsymbol{y}}(\nu)\nabla_{\boldsymbol{y}(k)}\mathcal{J}(\boldsymbol{y}(k;\nu))\right),$$



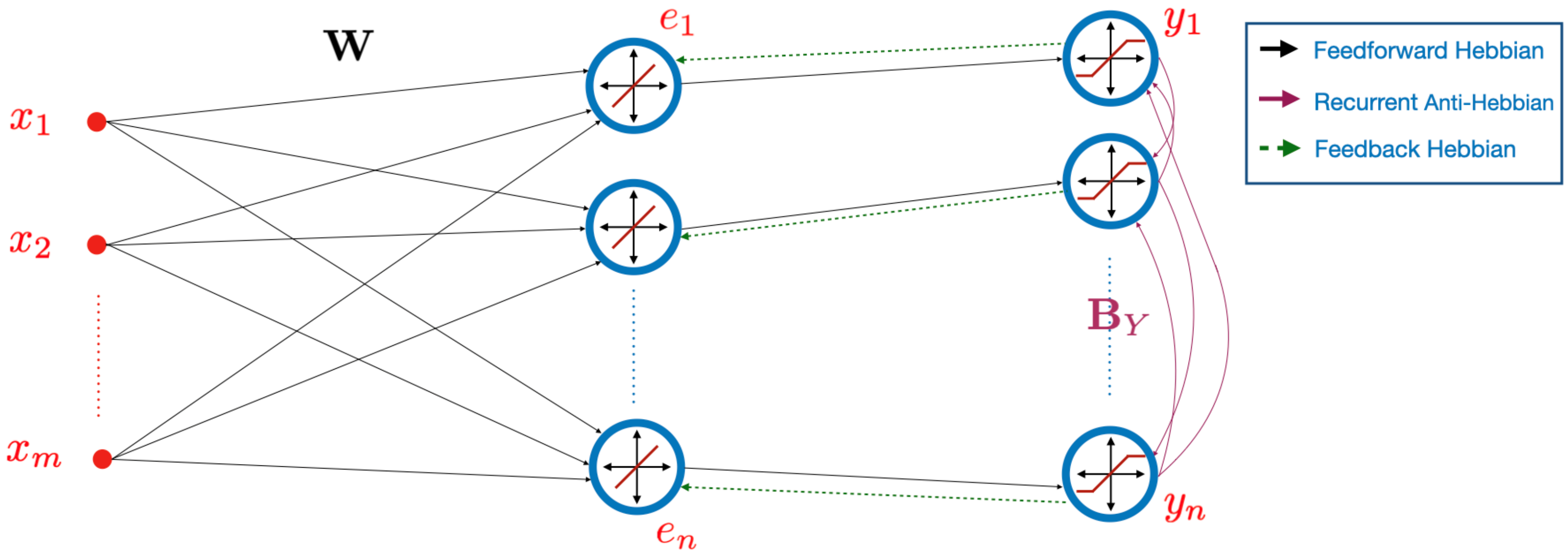
ℓ_∞ – Norm Ball

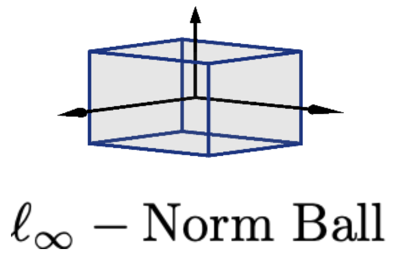
Anti-sparse CorInfoMax Neural Network



Anti-sparse CorInfoMax Network

Anti-sparse CorInfoMax Neural Network





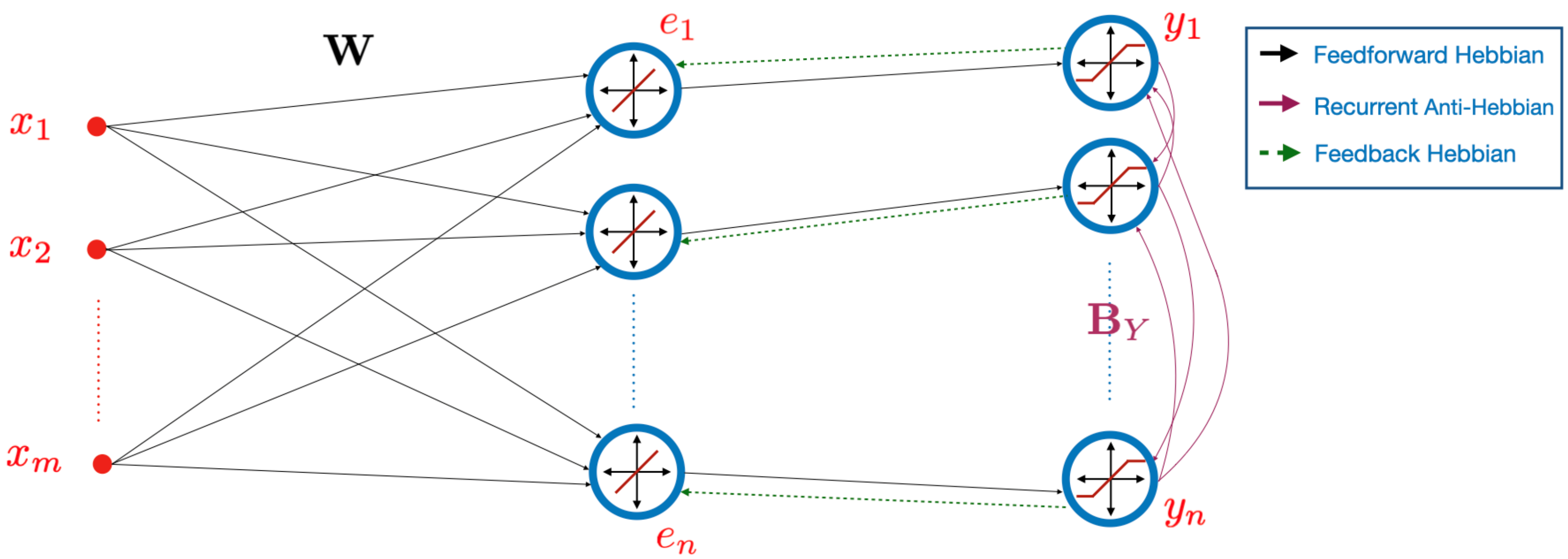
Weight Updates

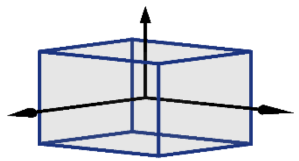
$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + \mu_{\boldsymbol{W}}(k)\boldsymbol{e}(k)\boldsymbol{x}(k)^T,$$

→ **Feed-forward weight**

Anti-sparse CorInfoMax Network

Anti-sparse CorInfoMax Neural Network





ℓ_∞ – Norm Ball

Weight Updates

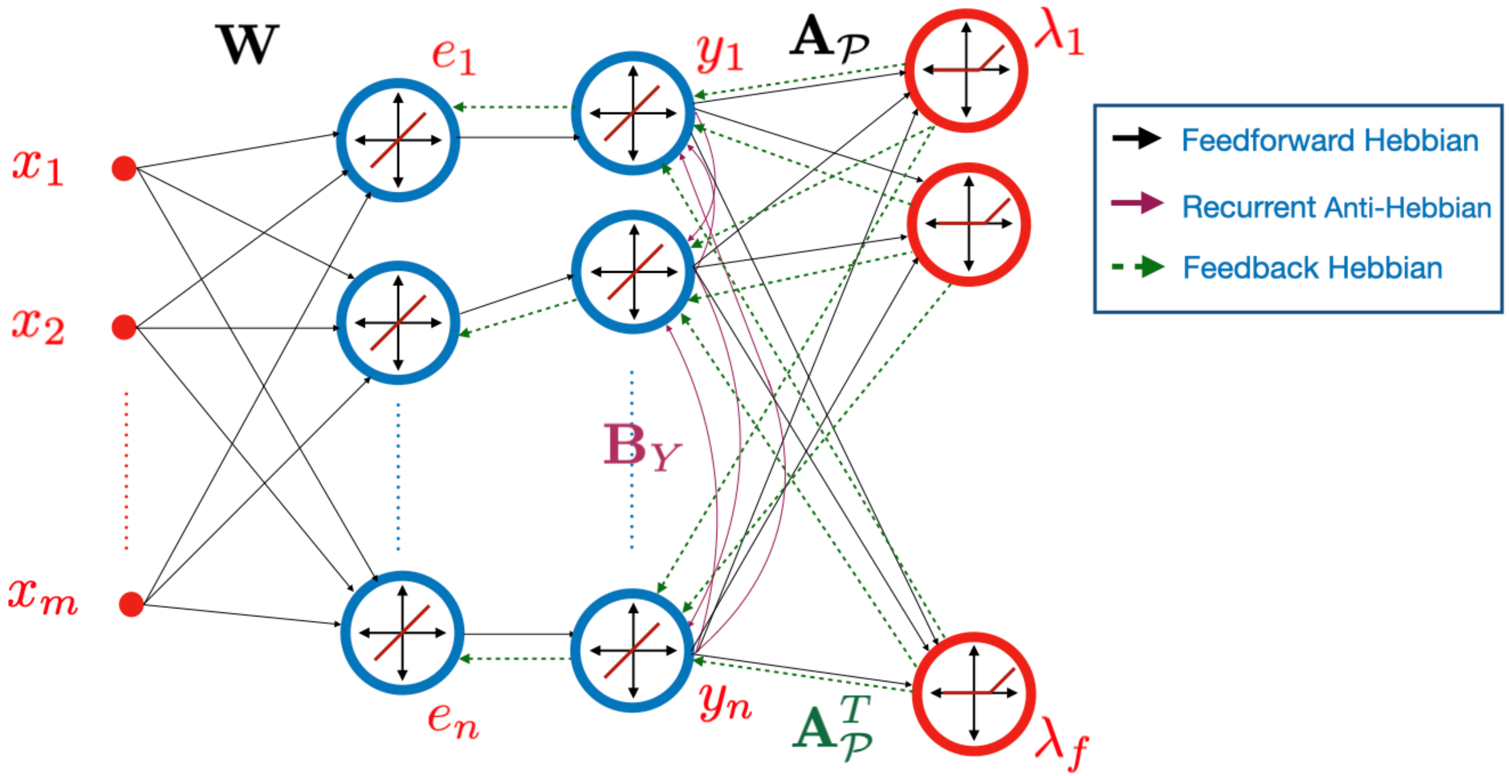
$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + \mu_{\boldsymbol{W}}(k)\boldsymbol{e}(k)\boldsymbol{x}(k)^T,$$

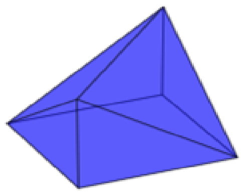
→ **Feed-forward weight**

$$\boldsymbol{B}_{\boldsymbol{y}}^{\zeta_y}(k+1) = \frac{1}{\zeta_{\boldsymbol{y}}}\left(\boldsymbol{B}_{\boldsymbol{y}}^{\zeta_y}(k) - \frac{1-\zeta_{\boldsymbol{y}}}{\zeta_{\boldsymbol{y}}}\boldsymbol{B}_{\boldsymbol{y}}^{\zeta_y}(k)\boldsymbol{y}(k)\boldsymbol{y}(k)^T\boldsymbol{B}_{\boldsymbol{y}}^{\zeta_y}(k)\right).$$

→ **Lateral weight**

Canonical CorInfoMax Network





$$\mathcal{P} = \{\boldsymbol{s} \in \mathbb{R}^n | \boldsymbol{A}_{\mathcal{P}} \boldsymbol{s} \preccurlyeq \boldsymbol{b}_{\mathcal{P}}\}$$

Output Updates

$$\boldsymbol{y}(k;\nu+1) = \boldsymbol{y}(k;\nu) + \eta_{\boldsymbol{y}}(\nu)\nabla_{\boldsymbol{y}(k)}\mathcal{L}(\boldsymbol{y}(k;\nu), \boldsymbol{\lambda}(k;\nu)),$$
$$\boldsymbol{\lambda}(k,\nu+1) = \text{ReLU}\left(\boldsymbol{\lambda}(k,\nu) - \eta_{\boldsymbol{\lambda}}(\nu)(\boldsymbol{b}_{\mathcal{P}} - \boldsymbol{A}_{\mathcal{P}}\boldsymbol{y}(k;\nu))\right).$$

Numerical Experiment: Correlated Anti-sparse Source Separation

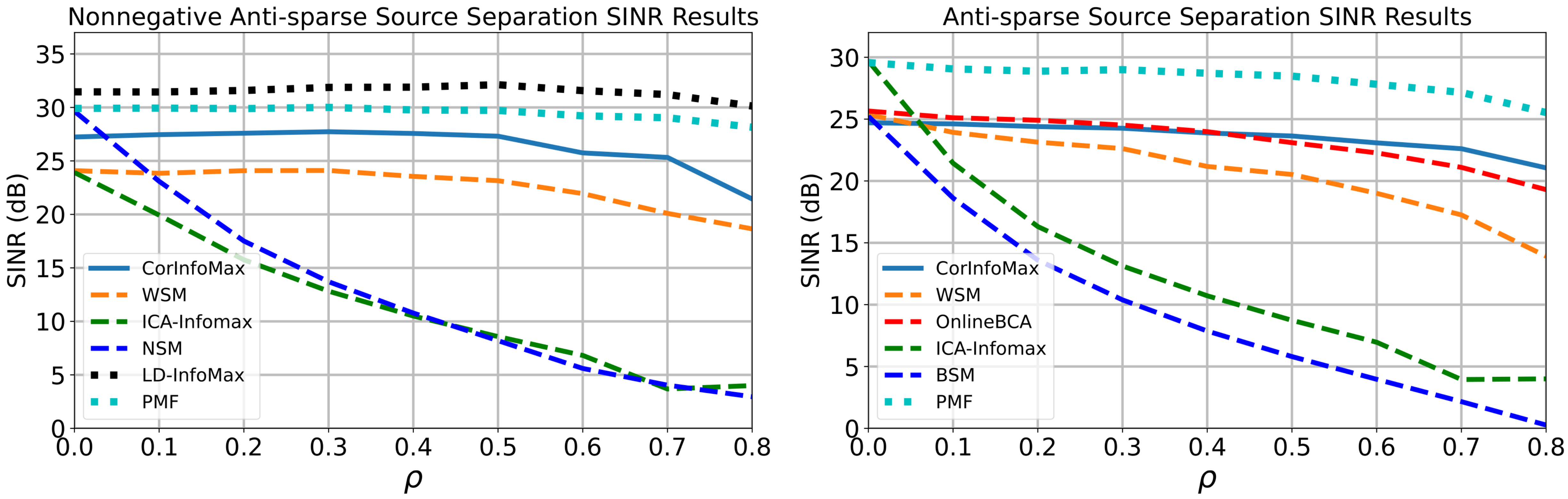


Fig. The SINR performances of CorInfoMax (ours), LD-InfoMax, PMF, ICA-InfoMax, NSM, BSM, and Online-BCA for antisparse source separation experiments. (Averaged over 100 realizations)

Numerical Experiment: Video Separation

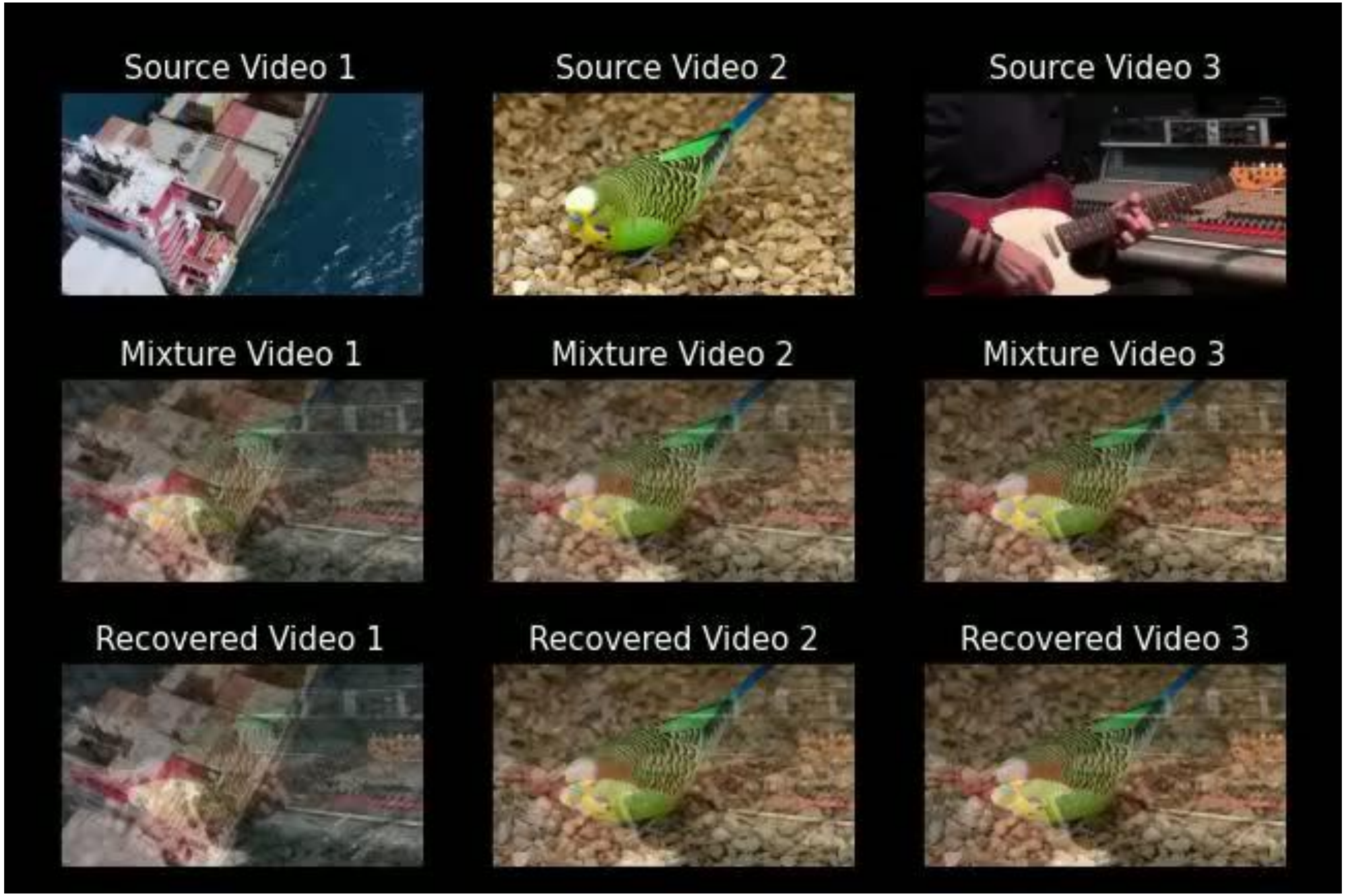


Fig. Video separation experiment of CorInfoMax: (first row) original sources, (second row) mixture videos, (third row) output of CorInfoMax.

Conclusion

- We proposed a novel normative approach for blind source separation problem via information maximization criterion,
- Our online formulation maps to two/three layer recurrent neural networks with local learning rules,
- The resulting framework for generating biologically plausible neural networks are applicable to diverse set of source types,
- We demonstrated diverse numerical experiments on both real and synthetic data for correlated/uncorrelated source separation.
- Our code is publicly available:
  https://github.com/BariscanBozkurt/Biologically-Plausible-Correlative-Information-Maximization-for-Blind-Source-Separation