Biologically-Plausible Determinant Maximization Neural Networks for Blind Separation of Correlated Sources

Bariscan Bozkurt[1,2] **Cengiz Pehlevan**[3] **Alper T. Erdogan**[1,2]
[1]KUIS AI Center, Koc University, Turkey [2]EEE Department, Koc University, Turkey
[3]John A. Paulson School of Engineering & Applied Sciences and Center for Brain Science, Harvard University, Cambridge, 02138 MA, USA
{bbozkurt15, alperdogan}@ku.edu.tr cpehlevan@seas.harvard.edu

Abstract

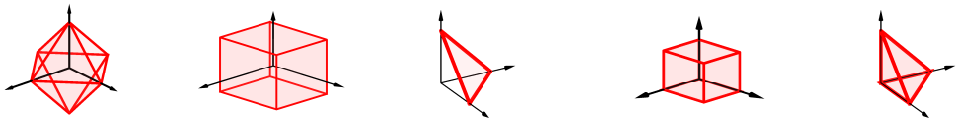
Extraction of latent sources of complex stimuli is critical for making sense of the world. While the brain solves this blind source separation (BSS) problem continuously, its algorithms remain unknown. Previous work on biologically-plausible BSS algorithms assumed that observed signals are linear mixtures of statistically independent or uncorrelated sources, limiting the domain of applicability of these algorithms. To overcome this limitation, we propose novel biologically-plausible neural networks for the blind separation of potentially dependent/correlated sources. Differing from previous work, we assume some general geometric, not statistical, conditions on the source vectors allowing separation of potentially dependent/correlated sources. Concretely, we assume that the source vectors are sufficiently scattered in their domains which can be described by certain polytopes. Then, we consider recovery of these sources by the Det-Max criterion, which maximizes the determinant of the output correlation matrix to enforce a similar spread for the source estimates. Starting from this normative principle, and using a weighted similarity matching approach that enables arbitrary linear transformations adaptable by local learning rules, we derive two-layer biologically-plausible neural network algorithms that can separate mixtures into sources coming from a variety of source domains. We demonstrate that our algorithms outperform other biologically-plausible BSS algorithms on correlated source separation problems.

1 Introduction

Our brains constantly and effortlessly extract latent causes, or sources, of complex visual, auditory or olfactory stimuli sensed by sensory organs [1–11]. This extraction is mostly done without any instruction, in an unsupervised manner, making the process an instance of the blind source separation (BSS) problem [12, 13]. Indeed, visual and auditory cortical receptive fields were argued to be the result of performing BSS on natural images [1, 2] and sounds [4]. The wide-spread use of BSS in the brain suggests the existence of generic circuit motifs that perform this task [14]. Consequently, the literature on biologically-plausible neural network algorithms for BSS is growing [15–19].

Because BSS is an underdetermined inverse problem, BSS algorithms make generative assumptions on observations. In most instances of the biologically-plausible BSS algorithms, complex stimuli are assumed to be linear mixtures of latent sources. This assumption is particularly fruitful and is used to model, for example, natural images [1, 20], and responses of olfactory neurons to complex odorants [21–23]. However, linear mixing by itself is not sufficient for source identifiability; further assumptions are needed. Previous work on biologically-plausible algorithms for BSS of linear mixtures

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) $\mathcal{B}_{\ell_1}$ (sparse) (b) $\mathcal{B}_{\ell_\infty}$ (anti-sparse) (c) Δ (normalized nonnegative) (d) $\mathcal{B}_{\ell_\infty,+}$ (nonnegative anti-sparse) (e) $\mathcal{B}_{\ell_1,+}$ (nonnegative sparse)

Figure 1: Examples of source domains leading to identifiable generative models.

assumed sources to be statistically independent [17, 19, 24] or uncorrelated [16, 18]. However, these assumptions are very limiting when considering real data where sources can themselves be correlated.

In this paper, we address the limitation imposed by independence assumptions and provide biologically-plausible BSS neural networks that can separate potentially correlated sources. We achieve this by considering various general geometric identifiability conditions on sources instead of statistical assumptions like independence or uncorrelatedness. In particular, 1) we make natural assumptions on the domains of source vectors–like nonnegativity, sparsity, anti-sparsity or boundedness (Figure 1)–and 2) we assume that latent source vectors are sufficiently spread in their domain [25, 26]. Because these identifiability conditions are not stochastic in nature, our neural networks are able to separate both independent and dependent sources.

We derive our biologically-plausible algorithms from a normative principle. A common method for exploiting our geometric identifiability conditions is to disperse latent vector estimates across their presumed domain by maximizing the determinant of their sample correlation matrix, i.e., the Det-Max approach [25, 27–30]. Starting from a Det-Max objective function with constraints that specify the domain of source vectors, and using mathematical tools introduced for mapping optimization algorithms to adaptive Hebbian neural networks [18, 31, 32], we derive two-layered neural networks that can separate potentially correlated sources from their linear mixtures (Figure 2). These networks contain feedforward, recurrent and feedback synaptic connections updated via Hebbian or anti-Hebbian update rules. The domain of latent sources determines the structure of the output layer of the neural network (Figure 2, Table 1 and Appendix D).

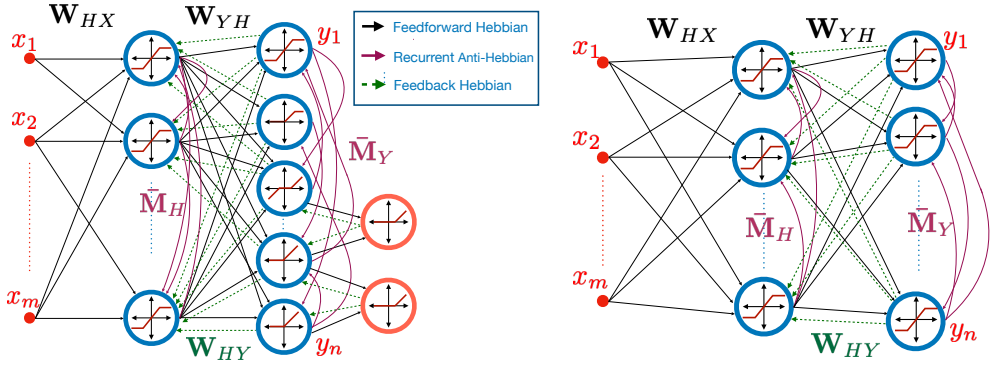
In summary, our main contributions in this article are the following:

- We propose a normative framework for generating biologically plausible neural networks that are capable of separating correlated sources from their mixtures by deriving them from a Det-Max objective function subject to source domain constraints.
- Our framework can handle infinitely many source types by exploiting their source domain topology.
- We demonstrate the performance of our networks in simulations with synthetic and realistic data.

1.1 Other related work

Several algorithms for separation of linearly mixed and correlated sources have been proposed outside the domain of biologically-plausible BSS. These algorithms make other forms of assumptions on the latent sources. Nonnegative matrix factorization (NMF) assumes that the latent vectors are nonnegative [13, 33–35]. Simplex structured matrix factorization (SSMF) assumes that the latent vectors are members of the unit-simplex [25, 36, 37]. Sparse component analysis (SCA) often assumes that the latent vectors lie in the unity ℓ_1-norm-ball [30, 38–42]. Antisparse bounded component analysis (BCA) assumes latent vectors are in the ℓ_∞-norm-ball [28, 29, 43]. Recently introduced polytopic matrix factorization (PMF) extends the identifiability-enabling domains to infinitely many polytopes obeying a particular symmetry restriction [26, 44, 45].

The mapping of optimization algorithms to biologically-plausible neural networks have been formalized in the similarity matching framework [31, 32, 46, 47]. Several BSS algorithms were proposed within this framework: 1) Nonnegative Similarity Matching (NSM) [16, 48] separates linear mixtures of uncorrelated nonnegative sources, 2) [19] separates independent sources, and 3) Bounded Similarity Matching (BSM) separates uncorrelated anti-sparse bounded sources from ℓ_∞-norm-ball



(a) General source domain with sparse components

(b) Antisparse sources

Figure 2: Det-Max WSM neural network for blind source separation. The network takes a mixed input $\mathbf{x}$ and produces latent components $\mathbf{y}$ at the output. The output layer depends on the choice of source domain. Mutually sparse components are connected by inhibitory neurons at the output layer.

[18]. BSM introduced a weighted inner product-based similarity criterion, referred to as the weighted similarity matching (WSM). Compared to these algorithm, the neural network algorithms we propose in this article 1) cover more general source domains, 2) handle potentially correlated sources, 3) use a two-layer WSM architecture (relative to single layer WSM architecture of BSM, which is not capable of generating arbitrary linear transformations) and 4) offer a general framework for neural-network-based optimization of the Det-Max criterion.

2 Problem statement

2.1 Sources

We assume that there are n real-valued sources, represented by the vector $\mathbf{s} \in \mathcal{P}$, where $\mathcal{P}$ is a particular subset of $\mathbb{R}^n$. Our algorithms will address a wide range of source domains. We list some examples before giving a more general criterion:

- *Bounded sparse sources*: A natural convex domain choice for sparse sources is the unit ℓ_1 norm ball $\mathcal{B}_{\ell_1} = \{\mathbf{s} \mid \|\mathbf{s}\|_1 \leq 1\}$ (Figure 1.(a)). The use of ℓ_1-norm as a convex (non)sparsity measure has been quite successful with various applications including sparse dictionary learning/component analysis [30, 39, 41, 49, 50] and modeling of V1 receptive fields [2].
- *Bounded anti-sparse sources*: A common domain choice for anti-sparse sources is the unit ℓ_∞-norm-ball: $\mathcal{B}_{\ell_\infty} = \{\mathbf{s} \mid \|\mathbf{s}\|_\infty \leq 1\}$ (Figure 1.(b)). If vectors drawn from $\mathcal{B}_{\ell_\infty}$ are well-spread inside this set, some samples would contain near-peak magnitude values simultaneously at all their components. The potential equal spreading of values among the components justifies the term "anti-sparse" [51] or "democratic" [52] component representations. This choice is well-suited for both applications in natural images and digital communication constellations [28, 43].
- *Normalized nonnegative sources*: Simplex structured matrix factorization [25, 36, 37] uses the unit simplex [35, 53] $\Delta = \{\mathbf{s} \mid \mathbf{s} \geq 0, \mathbf{1}^T\mathbf{s} = 1\}$ (Figure 1.(c)) as the source domain. Nonnegativity of sources naturally arises in biological context, for example in demixing olfactory mixtures [54].
- *Nonnegative bounded anti-sparse sources*: A non-degenerate polytopic choice of the nonnegative sources can be obtained through the combination of anti-sparseness and nonnegativity constraints. This corresponds to the intersection of $\mathcal{B}_{\ell_\infty}$ with the nonnegative orthant $\mathbb{R}^n_+$, represented as $\mathcal{B}_{\ell_\infty,+} = \mathcal{B}_{\ell_\infty} \cap \mathbb{R}^n_+$ [26] (Figure 1.(d)).
- *Nonnegative bounded sparse sources*: Another polytopic choice for nonnegative sources can be obtained through combination of the sparsity and nonnegativity constraints which yields the intersection of $\mathcal{B}_{\ell_1}$ with the nonnegative orthant $\mathbb{R}_+$, [26]: $\mathcal{B}_{\ell_1,+} = \mathcal{B}_{\ell_1} \cap \mathbb{R}^n_+$ (Figure 1.(e)).

Except the unit simplex Δ, all the examples above are examples of an infinite set of identifiable polytopes whose symmetry groups are restricted to the combinations of component permutations and sign alterations as formalized in PMF framework for BSS [44]. Further, in-

stead of a homogeneous choice of features, such as sparsity and nonnegativity, globally imposed on all elements of the component vector, we can assign these attributes at the subvector level and still obtain identifiable polytopes. For example, the reference [26] provides the set $\mathcal{P}_{ex} = \left\{ \mathbf{s} \in \mathbb{R}^3 \,\middle|\, s_1, s_2 \in [-1,1],\, s_3 \in [0,1],\, \left\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \right\|_1 \leq 1,\, \left\| \begin{bmatrix} s_2 \\ s_3 \end{bmatrix} \right\|_1 \leq 1 \right\}$, as a simple illustration of such polytopes with heterogeneous structure where s_3 is nonnegative, s_1, s_2 are signed, and $\begin{bmatrix} s_1 & s_2 \end{bmatrix}^T, \begin{bmatrix} s_2 & s_3 \end{bmatrix}^T$ are sparse subvectors, while sparsity is not globally imposed. In this article, we concentrate on particular source domains including the unit simplex, and the subset of identifiable polytopes for which the attributes such as sparsity and nonnegativity are defined at the subvector level in the general form

$$\mathcal{P} = \left\{ \mathbf{s} \in \mathbb{R}^n \,\middle|\, s_i \in [-1,1]\, \forall i \in \mathcal{I}_s,\, s_i \in [0,1]\, \forall i \in \mathcal{I}_+,\, \|\mathbf{s}_{\mathcal{J}_k}\|_1 \leq 1,\, \mathcal{J}_k \subseteq \mathbb{Z}_n,\, k \in \mathbb{Z}_L \right\}, \quad (1)$$

where $\mathcal{I}_+ \subseteq \mathcal{Z}_n$ is the index set for nonnegative sources, and I_s is its complement, $\mathbf{s}_{\mathcal{J}_k}$ is the subvector constructed from the elements with indices in $\mathcal{J}_k$, and L is the number of sparsity constraints imposed in the subvector level.

The Det-Max criterion for BSS is based on the assumption that the source samples are well-spread in their presumed domain. The references [55] and [26] provide precise conditions on the scattering of source samples which guarantee their identifiability for the unit simplex and polytopes, respectively. Appendix A provides a brief summary of these conditions.

We emphasize that our assumptions about the sources are deterministic. Therefore, our proposed algorithms do not exploit any stochastic assumptions such as independence or uncorrelatedness, and can separate both independent and dependent (potentially correlated) sources.

2.2 Mixing

The sources $\mathbf{s}_t$ are mixed through a mixing matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$.

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad t \in \mathbb{Z}. \quad (2)$$

We only consider the (over)determined case with $m \geq n$ and assume that the mixing matrix is full-rank. While we consider noiseless mixtures to achieve perfect separability, the optimization setting proposed for the online algorithm features a particular objective function that safeguards against potential noise presence. We use $\mathbf{S}(t) = \begin{bmatrix} \mathbf{s}_1 & \dots & \mathbf{s}_t \end{bmatrix} \in \mathbb{R}^{n \times t}$ and $\mathbf{X}(t) = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_t \end{bmatrix} \in \mathbb{R}^{m \times t}$ to represent data snapshot matrices, at time t, for sources and mixtures, respectively.

2.3 Separation

The goal of the source separation is to obtain an estimate of $\mathbf{S}(t)$ from the mixture measurements $\mathbf{X}(t)$ when the mixing matrix $\mathbf{A}$ is unknown. We use the notation $\mathbf{y}_t$ to refer to source estimates, which are linear transformations of observations, i.e., $\mathbf{y}_i = \mathbf{W}\mathbf{x}_i$, where $\mathbf{W} \in \mathbb{R}^{n \times m}$. We define $\mathbf{Y}(t) = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_t \end{bmatrix} \in \mathbb{R}^{n \times t}$ as the output snapshot matrix. "Ideal separation" is defined as the condition where the outputs are scaled and permuted versions of original sources, i.e., they satisfy $\mathbf{y}_t = \mathbf{P}\mathbf{\Lambda}\mathbf{s}_t$, where $\mathbf{P}$ is a permutation matrix, and $\mathbf{\Lambda}$ is a full rank diagonal matrix.

3 Determinant maximization based blind source separation

Among several alternative solution methods for the BSS problem, the determinant-maximization (Det-Max) criterion has been proposed within the NMF, BCA, and PMF frameworks, [26–28, 30, 35, 44]. Here, the separator is trained to maximize the (log)-determinant of the sample correlation matrix for the separator outputs, $J(\mathbf{W}) = \log(\det(\hat{\mathbf{R}}_y(t)))$, where $\hat{\mathbf{R}}_y(t)$ is the sample correlation matrix $\hat{\mathbf{R}}_y(t) = \frac{1}{t}\sum_{i=1}^{t} \mathbf{y}_i\mathbf{y}_i^T = \frac{1}{t}\mathbf{Y}(t)\mathbf{Y}(t)^T$. Further, during the training process, the separator outputs are constrained to lie inside the presumed source domain, i.e. $\mathcal{P}$. As a result, we can pose the corresponding optimization problem as [26, 35]

$$\underset{\mathbf{Y}(t)}{\text{maximize}} \quad \log(\det(\mathbf{Y}(t)\mathbf{Y}(t)^T)) \quad (3a)$$

$$\text{subject to} \quad \mathbf{y}_i \in \mathcal{P}, i = 1, \dots, t, \quad (3b)$$

where we ignored the constant $\frac{1}{t}$ term. Here, the determinant of the correlation matrix acts as a spread measure for the output samples. If the original source samples $\{\mathbf{s}_1, \ldots, \mathbf{s}_t\}$ are sufficiently scattered inside the source domain $\mathcal{P}$, as described in Section 2.1 and Appendix A, then the global solution of this optimization can be shown to achieve perfect separation [26, 35, 55].

4 An alternative optimization formulation of determinant-maximization based on weighted similarity matching

Here, we reformulate the Det-Max problem 3 described above in a way that allows derivation of a biologically-plausible neural network for the linear BSS setup in Section 2. Our formulation applies to all source types discusses in 2.1.

We propose the following optimization problem:

$$\underset{\substack{\mathbf{Y}(t),\mathbf{H}(t),D_{1,11}(t),\ldots D_{1,nn}(t),\mathbf{D}_1(t)\\ D_{2,11}(t),\ldots D_{2,nn}(t),\mathbf{D}_2(t)}}{\text{minimize}} \quad \sum_{i=1}^{n} \log(D_{1,ii}(t)) + \sum_{i=1}^{n} \log(D_{2,ii}(t)) \tag{4a}$$

$$\text{subject to} \quad \mathbf{X}(t)^T\mathbf{X}(t) - \mathbf{H}(t)^T\mathbf{D}_1(t)\mathbf{H}(t) = 0, \tag{4b}$$

$$\mathbf{H}(t)^T\mathbf{H}(t) - \mathbf{Y}(t)^T\mathbf{D}_2(t)\mathbf{Y}(t) = 0, \tag{4c}$$

$$\mathbf{y}_i \in \mathcal{P}, i = 1, \ldots, n, \tag{4d}$$

$$\mathbf{D}_l(t) = \text{diag}(D_{l,11}(t), \ldots, D_{l,nn}(t)), \quad l = 1, 2, \tag{4e}$$

$$D_{l,11}(t), D_{l,22}(t), \ldots, D_{l,nn}(t) > 0, \quad l = 1, 2 \tag{4f}$$

Here, $\mathbf{X}(t) \in \mathbb{R}^{m\times t}$ is the matrix containing input (mixture) vectors, $\mathbf{Y}(t) \in \mathbb{R}^{n\times t}$ is the matrix containing output vectors, $\mathbf{H}(t) \in \mathbb{R}^{n\times t}$ is a slack variable containing an intermediate signal $\{\mathbf{h}_i \in \mathbb{R}^n, i = 1, \ldots, t\}$, corresponding to the hidden layer of the neural network implementation in Section 5, in its columns $\mathbf{H}(t) = [\ \mathbf{h}_1 \quad \mathbf{h}_2 \quad \ldots \quad \mathbf{h}_t\ ]$. $D_{l,11}(t), D_{l,22}(t), \ldots, D_{l,nn}(t)$ for $l = 1, 2$ are nonnegative slack variables to be described below, and $\mathbf{D}_l$ is the diagonal matrix containing weights $D_{l,ii}$ for $i = 1, \ldots, n$ and $l = 1, 2$. The constraint (4d) ensures that the outputs lie in the presumed domain of sources.

This problem is related to the weighted similarity matching (WSM) objective introduced in [18]. Constraints (4b) and (4c) define two separate WSM conditions. In particular, the equality constraint in (4b) is a WSM constraint between inputs and the intermediate signal $\mathbf{H}(t)$. This constraint imposes that the pairwise weighted correlations of the signal $\{\mathbf{h}_i, i = 1, \ldots, t\}$ are the same as correlations among the elements of the input signal $\{\mathbf{x}_i, i = 1, \ldots, t\}$, i.e., $\mathbf{x}_i^T\mathbf{x}_j = \mathbf{h}_i^T\mathbf{D}_1(t)\mathbf{h}_j, \quad \forall i, j \in \{1, \ldots, t\}$. $D_{1,11}(t), D_{1,22}(t), \ldots, D_{1,nn}(t)$ correspond to inner product weights used in these equalities. Similarly, the equality constraint in (4c) defines a WSM constraint between the intermediate signal and outputs. This equality can be written as $\mathbf{h}_i^T\mathbf{h}_j = \mathbf{y}_i^T\mathbf{D}_2(t)\mathbf{y}_j, \quad i, j \in \{1, \ldots, t\}$, and $D_{2,11}(t), D_{2,22}(t), \ldots, D_{2,nn}(t)$ correspond to the inner product weights used in these equalities. The optimization involves minimizing the logarithm of the determinant of the weighting matrices.

Now we state the relation between our WSM-based objective and the original Det-Max criterion (3).

Theorem 1. *If $\mathbf{X}(t)$ is full column-rank, then global optimal $\mathbf{Y}(t)$ solutions of* (3) *and* (4) *coincide.*

Proof of Theorem 1. See Appendix B for the proof. The proof relies on a lemma that states that the optimization constraints enforce inputs and outputs to be related by an arbitrary linear transformation. □

5 Biologically-plausible neural networks for WSM-based BSS

The optimization problems we considered so far were in an offline setting, where all inputs are observed together and all outputs are produced together. However, biology operates in an online fashion, observing an input and producing the corresponding output, before seeing the next input. Therefore, in this section, we first introduce an online version of the batch WSM-problem (4). Then we show that the corresponding gradient descent algorithm leads to a two-layer neural network with biologically-plausible local update rules.

5.1 Online optimization setting for WSM-based BSS

We first propose an online extension of WSM-based BSS (4). In the online setting, past outputs cannot be altered, but past inputs and outputs still carry valuable information about solving the BSS problem. We will write down an optimization problem whose goal is to produce the sources $\mathbf{y}_t$ given a mixture $\mathbf{x}_t$, while exploiting information from all the fixed previous inputs and outputs.

We first introduce our notation. We consider exponential weighting of the signals as a recipe for dynamical adjustment to potential nonstationarity in the data. We define the weighted input data snapshot matrix by time t as, $\boldsymbol{\mathcal{X}}(t) = \begin{bmatrix} \gamma^{t-1}\mathbf{x}_1 & \dots & \gamma\mathbf{x}_{t-1} & \mathbf{x}_t \end{bmatrix} = \mathbf{X}(t)\boldsymbol{\Gamma}(t)$, where γ is the forgetting factor and $\boldsymbol{\Gamma}(t) = \text{diag}(\gamma^{t-1}, \dots, \gamma, 1)$. The exponential weighting emphasizes recent mixtures by reducing the impact of past samples. Similarly, we define the corresponding weighted output snapshot matrix for output as $\boldsymbol{\mathcal{Y}}(t) = \begin{bmatrix} \gamma^{t-1}\mathbf{y}_1 & \dots & \gamma\mathbf{y}_{t-1} & \mathbf{y}_t \end{bmatrix} = \mathbf{Y}(t)\boldsymbol{\Gamma}(t)$, and the hidden layer vectors as $\boldsymbol{\mathcal{H}}(t) = \begin{bmatrix} \gamma^{t-1}\mathbf{h}_1 & \dots & \gamma\mathbf{h}_{t-1} & \mathbf{h}_t \end{bmatrix} = \mathbf{H}(t)\boldsymbol{\Gamma}(t)$. We further define $\tau = \lim_{t\to\infty} = \sum_{k=0}^{t-1}\gamma^{2k} = \frac{1}{1-\gamma^2}$as a measure of the effective time window length for sample correlation calculations based on the exponential weights.

In order to derive an online cost function, we first converted equality constraints in (4b) and (4c) to similarity matching cost functions $J_1(\mathbf{H}(t), \mathbf{D}_1(t)) = \frac{1}{2\tau^2}\|\boldsymbol{\mathcal{X}}(t)^T\boldsymbol{\mathcal{X}}(t) - \boldsymbol{\mathcal{H}}(t)^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t)\|_F^2$, $J_2(\mathbf{H}(t), \mathbf{D}_2(t), \mathbf{Y}(t)) = \frac{1}{2\tau^2}\|\boldsymbol{\mathcal{H}}(t)^T\boldsymbol{\mathcal{H}}(t) - \boldsymbol{\mathcal{Y}}(t)^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t)\|_F^2$. Then, a weighted combination of similarity matching costs and the objective function in (4a) yields the final cost function

$$\begin{aligned}\mathcal{J}(\mathbf{H}(t), \mathbf{D}_1(t), \mathbf{D}_2(t), \mathbf{Y}(t)) \quad = \quad & \lambda_{SM}[\beta J_1(\mathbf{H}(t), \mathbf{D}_1(t)) + (1-\beta)J_2(\mathbf{H}(t), \mathbf{D}_2(t), \mathbf{Y}(t))] \\ & +(1-\lambda_{SM})[\sum_{k=1}^{n}\log(D_{1,kk}(t)) + \sum_{k=1}^{n}\log(D_{2,kk}(t))]. \end{aligned} \tag{5}$$

Here, $\beta \in [0,1]$ and $\lambda_{SM} \in [0,1]$ are parameters that convexly combine similarity matching costs and the objective function. Finally, we can state the online optimization problem for determining the current output $\mathbf{y}_t$, the corresponding hidden state $\mathbf{h}_t$ and for updating the gain parameters $\mathbf{D}_l(t)$ for $l = 1, 2$, as

$$\underset{\substack{\mathbf{y}_t, \mathbf{h}_t, D_{1,11}(t), \dots D_{1,nn}(t), \mathbf{D}_1(t) \\ D_{2,11}(t), \dots D_{2,nn}(t), \mathbf{D}_2(t)}}{\text{minimize}} \quad \mathcal{J}(\mathbf{H}(t), \mathbf{D}_1(t), \mathbf{D}_2(t), \mathbf{Y}(t)) \tag{6a}$$

$$\text{subject to} \quad \mathbf{y}_t \in \mathcal{P}, \tag{6b}$$

$$\mathbf{D}_l(t) = \text{diag}(D_{l,11}(t), \dots, D_{l,nn}(t)), \quad l = 1, 2, \tag{6c}$$

$$D_{l,11}(t), D_{l,22}(t), \dots, D_{l,nn}(t) > 0, \quad l = 1, 2 \tag{6d}$$

As shown in Appendix C.1, part of $\mathcal{J}$ that depends on $\mathbf{h}_t$ and $\mathbf{y}_t$ can be written as

$$\begin{aligned} C(\mathbf{h}_t, \mathbf{y}_t) \quad = \quad & 2\mathbf{h}_t^T\mathbf{D}_1\mathbf{M}_H(t)\mathbf{D}_1(t)\mathbf{h}_t - 4\mathbf{h}_t^T\mathbf{D}_1(t)\mathbf{W}_{HX}(t)\mathbf{x}_t \\ & +2\mathbf{y}_t^T\mathbf{D}_2(t)\mathbf{M}_Y(t)\mathbf{D}_2(t)\mathbf{y}_t - 4\mathbf{y}_t^T\mathbf{D}_2(t)\mathbf{W}_{YH}(t)\mathbf{h}_t + 2\mathbf{h}_t^T\mathbf{M}_H(t)\mathbf{h}_t, \end{aligned} \tag{7}$$

where the dependence on past inputs and outputs appear in the weighted correlation matrices:

$$\begin{aligned} \mathbf{M}_H(t) &= \tfrac{1}{\tau}\textstyle\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{h}_k\mathbf{h}_k^T, \quad \mathbf{W}_{HX}(t) = \tfrac{1}{\tau}\textstyle\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{h}_k\mathbf{x}_k^T, \\ \mathbf{W}_{YH}(t) &= \tfrac{1}{\tau}\textstyle\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{y}_k\mathbf{h}_k^T, \quad \mathbf{M}_Y(t) = \tfrac{1}{\tau}\textstyle\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{y}_k\mathbf{y}_k^T. \end{aligned} \tag{8}$$

5.2 Description of network dynamics for bounded anti-sparse sources

We now show that the gradient-descent minimization of the online WSM cost function in (6) can be interpreted as the dynamics of a neural network with local learning rules. The exact network architecture is determined by the presumed identifiable source domain $\mathcal{P}$, which can be chosen in infinitely many ways. In this section, we concentrate on the domain choice $\mathcal{P} = \mathcal{B}_\infty$ as an illustrative example. In Section 5.3, we discuss how to generalize the results of this section by modifying the output layer for different identifiable source domains. We start by writing the update expressions for the optimization variables based on the gradients of $\mathcal{J}(\mathbf{h}_t, \mathbf{y}_t, \mathbf{D}_1(t), \mathbf{D}_2(t))$:

Update dynamics for $\mathbf{h}_t$: Following previous work [48, 56], and using the gradient of (7) in (A.8) with respect to $\mathbf{h}_t$, we can write down an update dynamics for $\mathbf{h}_t$ in the form

$$\begin{aligned} \frac{d\mathbf{v}(\tau)}{d\tau} &= -\mathbf{v}(\tau) - \lambda_{SM}[((1-\beta)\bar{\mathbf{M}}_H(t) + \beta\mathbf{D}_1(t)\bar{\mathbf{M}}_H(t)\mathbf{D}_1(t))\mathbf{h}(\tau) \\ &\quad +\beta\mathbf{D}_1(t)\mathbf{W}_{HX}(t)\mathbf{x}(\tau) + (1-\beta)\mathbf{W}_{YH}(t)^T\mathbf{D}_2(t)\mathbf{y}(\tau)] \qquad (9) \\ \mathbf{h}_{t,i}(\tau) &= \sigma_A\left(\frac{\mathbf{v}_i(\tau)}{\lambda_{SM}\Gamma_{Hii}(t)((1-\beta)+\beta D_{1,ii}(t)^2)}\right), \quad \text{for } i = 1,\dots n, \qquad (10) \end{aligned}$$

where $\mathbf{\Gamma}_H(t)$ is a diagonal matrix containing diagonal elements of $\mathbf{M}_H(t)$ and $\bar{\mathbf{M}}_H(t) = \mathbf{M}_H(t) - \mathbf{\Gamma}_H(t)$, $\sigma(\cdot)$ is the clipping function, defined as $\sigma_A(x) = \begin{cases} x & -A \le x \le A, \\ A\text{sign}(x) & \text{otherwise.} \end{cases}$. This dynamics can be shown to minimize (7) [56]. Here $\mathbf{v}(\tau)$ is an internal variable that could be interpreted as the voltage dynamics of a biological neuron, and is defined based on a linear transformation of $\mathbf{h}_t$ in (A.9). Equation (9) defines $\mathbf{v}(\tau)$ dynamics from the gradient of (7) with respect to $\mathbf{h}_t$ in (A.8). Due to the positive definite linear map in (A.9), the expression in (A.8) also serves as a descent direction for $\mathbf{v}(\tau)$. Furthermore, $\sigma(\cdot)$ function is the projection onto $A\mathcal{B}_\infty$, where $[-A, A]$ is the presumed dynamic range for the components of $\mathbf{h}_t$. We note that there is no explicit constraint set for $\mathbf{h}_t$ in the online optimization setting of Section 5.1, and therefore, A can be chosen as large as desired in the actual implementation. We included the nonlinearity in (10) to model the limited dynamic range of an actual (biological) neuron.

Update dynamics for output $\mathbf{y}_t$: We write the update dynamics for the output $\mathbf{y}_t$, based on (A.11) as

$$\begin{aligned} \frac{d\mathbf{u}(\tau)}{d\tau} &= -\mathbf{u}(\tau) + \mathbf{W}_{YH}(t)\mathbf{h}(\tau) - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}(\tau), \qquad (11) \\ \mathbf{y}_{t,i}(\tau) &= \sigma_1\left(\frac{\mathbf{u}_i(\tau)}{\Gamma_{Yii}(t)D_{2,ii}(t)}\right), \qquad \text{for } i = 1,\dots n, \qquad (12) \end{aligned}$$

which is derived using the same approach for $\mathbf{h}_t$, where we used the descent direction expression in (A.11), and the substitution in (A.12). Here, $\mathbf{\Gamma}_Y(t)$ is a diagonal matrix containing diagonal elements of $\mathbf{M}_Y(t)$ and $\bar{\mathbf{M}}_Y(t) = \mathbf{M}_Y(t) - \mathbf{\Gamma}_Y(t)$. Note that the nonlinear mapping $\sigma_1(\cdot)$ is the projection onto the presumed domain of sources, i.e., $\mathcal{P} = \mathcal{B}_\infty$, which is elementwise clipping operation.

The state space representations in (9)-(10) and (11)-(12) correspond to a two-layer recurrent neural network with input $\mathbf{x}_t$, hidden layer activation $\mathbf{h}_t$, output layer activation $\mathbf{y}_t$, $\mathbf{W}_{HX}$ ($\mathbf{W}_{HX}^T$) and $\mathbf{W}_{YH}$ ($\mathbf{W}_{YH}^T$) are the feedforward (feedback) synaptic weight matrices for the first and the second layers, respectively, and $\bar{\mathbf{M}}_H$ and $\bar{\mathbf{M}}_Y$ are recurrent synaptic weight matrices for the first and the second layers, respectively. The corresponding neural network schematic is provided in Figure 2.(b). The gain and synaptic weight dynamics below describe the learning mechanism for this network:

Update dynamics for gains $D_{l,ii}$: Using the derivative of the cost function with respect to $D_{1,ii}$ in (A.13), we can write the dynamics corresponding to the gain variable $D_{1,ii}$ as

$$\mu_{D_1}\frac{dD_{1,ii}(t)}{dt} = -(\lambda_{SM}\beta)(\|\mathbf{M}_{Hi,:}\|^2_{\mathbf{D}_1(t)} - \|\mathbf{W}_{HXi,:}\|_2^2) - (1-\lambda_{SM})\frac{1}{D_{1,ii}(t)}, \qquad (13)$$

where μ_{D_1} corresponds to the learning time-constant. Similarly, for the gain variable $D_{2,ii}$, the corresponding coefficient dynamics expression based on (A.14) is given by

$$\mu_{D_2}\frac{dD_{2,ii}(t)}{dt} = -(\lambda_{SM}\beta)(\|\mathbf{M}_{Yi,:}\|^2_{\mathbf{D}_2(t)} - \|\mathbf{W}_{YHi,:}\|_2^2) - (1-\lambda_{SM})\frac{1}{D_{2,ii}(t)}, \qquad (14)$$

where μ_{D_2} corresponds to the learning time-constant.

The inverses of the inner product weights $D_{l,ii}$ correspond to homeostatic gain parameters. The inspection of the gain updates in (13) and (14) leads to an interesting observation: whether the corresponding gain is going to increase or decrease depends on the balance between the norms of the recurrent and the feedforward synaptic strengths, which are the statistical indicators of the recent output and input activations, respectively. Hence, the homeostatic gain of the neuron will increase (decrease) if the level of recent output activations falls behind (surpasses) the level of recent input

activations to balance input/output energy levels. The resulting dynamics align with the experimental homeostatic balance observed in biological neurons [57].

Based on the definitions of the synaptic weight matrices in (8), we can write their updates as

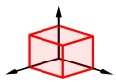
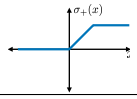
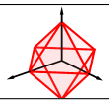
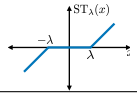
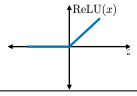
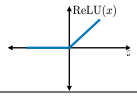
$$\mathbf{M}_H(t+1) = \gamma^2 \mathbf{M}_H(t) + (1-\gamma^2)\mathbf{h}_t\mathbf{h}_t^T, \quad \mathbf{M}_Y(t+1) = \gamma^2 \mathbf{M}_Y(t) + (1-\gamma^2)\mathbf{y}_t\mathbf{y}_t^T, \quad (15)$$
$$\mathbf{W}_{HX}(t+1) = \gamma^2 \mathbf{W}_{HX}(t) + (1-\gamma^2)\mathbf{h}_t\mathbf{x}_t^T, \quad \mathbf{W}_{YH}(t+1) = \gamma^2 \mathbf{W}_{YH}(t) + (1-\gamma^2)\mathbf{y}_t\mathbf{h}_t^T.$$

These updates are local in the sense that they only depend on variables available to the synapse, and hence are biologically plausible.

5.3 Det-max WSM neural network examples for more general source domains

Det-Max Neural Network obtained for the source domain $\mathcal{P} = \mathcal{B}_\infty$ in Section 5.2 can be extended to more general identifiable source domains by only changing the output dynamics. In Appendix D, we provide illustrative examples for different identifiable domain choices. Table 1 summarizes the output dynamics obtained for the identifiable source domain examples in Figure 1.

Table 1: Example source domains from Figure 1 and the corresponding output dynamics.

	Source Domain	Output Dynamics	Output Activation
	$\mathcal{P} = \mathcal{B}_{\infty,+}$ Nonnegative Anti-sparse	$\mathbf{y}_{t,i}(\tau) = \sigma_+\left(\frac{\mathbf{u}_i(\tau)}{\Gamma_{Y\,ii}(t)D_{2,ii}(t)}\right)$	
	$\mathcal{P} = \mathcal{B}_1$ Sparse	$\mathbf{y}_{t,i}(\tau) = \mathrm{ST}_{\lambda_1(\tau)}\left(\frac{u_i(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Y\,ii}(t)D_{2,ii}(t)}\right)$ $\frac{da(\tau)}{d\tau} = -a(\tau) + \sum_{k=0}^{n} \lvert\mathbf{y}_{t,k}(\tau)\rvert - 1 + \lambda_1(\tau),$ $\lambda_1(\tau) = \mathrm{ReLU}(a(\tau))$	
	$\mathcal{P} = \mathcal{B}_{1,+}$ Nonnegative Sparse	$\mathbf{y}_{t,i}(\tau) = \mathrm{ReLU}\left(\frac{u_i(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Y\,ii}(t)D_{2,ii}(t)} - \lambda_1(\tau)\right)$ $\frac{da(\tau)}{d\tau} = -a(\tau) + \sum_{k=0}^{n} \mathbf{y}_{t,k}(\tau) - 1 + \lambda_1(\tau),$ $\lambda_1(\tau) = \mathrm{ReLU}(a(\tau))$	
	$\mathcal{P} = \Delta$ Unit Simplex	$\mathbf{y}_{t,i}(\tau) = \mathrm{ReLU}\left(\frac{u_i(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Y\,ii}(t)D_{2,ii}(t)} - \lambda_1(\tau)\right)$ $\frac{d\lambda_1(\tau)}{d\tau} = -\lambda_1(\tau) + \sum_{k=0}^{n} \mathbf{y}_{t,k}(\tau) - 1 + \lambda_1(\tau)$	

We can make the following observations on Table 1: (1) For sparse and unit simplex settings, there is an additional inhibitory neuron which takes input from all outputs and whose activation is the inhibitory signal $\lambda_1(\tau)$, (2) The source attributes, which are globally defined over all sources, determine the activation functions at the output layer. The proposed framework can be applied to any polytope described by (1) for which the corresponding Det-Max neural network will contain combinations of activation functions in Table 1 as illustrated in Figure 2.(a).

6 Numerical experiments

In this section, we illustrate the applications of the proposed WSM-based BSS framework for both synthetic and natural sources. More details on these experiments and additional examples are provided in Appendix E, including sparse dictionary learning. Our implementation code is publicly available[1].

6.1 Synthetically correlated source separation

In order to illustrate the correlated source separation capability of the proposed WSM neural networks, we consider a numerical experiment with five copula-T distributed (uniform and correlated) sources. For the correlation calibration matrix for these sources, we use Toeplitz matrix whose first row is $[1 \quad \rho \quad \rho \quad \rho \quad \rho]$. The ρ parameter determines the correlation level, and we considered the range $[0, 0.6]$ for this parameter. These sources are mixed with a 10×5 random matrix with independent and identically distributed (i.i.d.) standard normal random variables. The mixtures are corrupted by i.i.d.

[1] https://github.com/BariscanBozkurt/Biologically-Plausible-DetMaxNNs-for-Blind-Source-Separation

normal noise corresponding to 30dB signal-to-noise ratio (SNR) level. In this experiment, we employ the nonnegative-antisparse-WSM neural network (Figure 8 in Appendix D.2) whose activation functions at the output layer are nonnegative-clipping functions, as the sources are nonnegative uniform random variables.

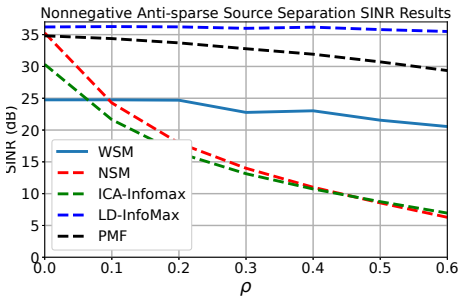


Figure 3: The SINRs of WSM, NSM, ICA, PMF, and LD-InfoMax versus the correlation parameter ρ.

We compared the signal-to-interference-plus-noise-power-ratio (SINR) performance of our algorithm with the NSM algorithm [16], Infomax ICA algorithm [1], as implemented in Python MNE Toolbox [58], LD-InfoMax algorithm [59], and PMF algorithm [26]. Figure 3 shows the SINR performances of these algorithms (averaged over 300 realizations) as a function of the correlation parameter ρ. We observe that our WSM-based network performs well despite correlations. In contrast, performance of NSM and ICA algorithms, which assume uncorrelated sources, degrade noticeably with increasing correlation levels. In addition, we note that the performance of batch Det-Max algorithms, i.e., LD-InfoMax and PMF, are also robust against source correlations. Furthermore, due to their batch nature, these algorithms typically achieved better performance results than our neural network with online-restriction, as expected.

6.2 Image separation

To further illustrate the correlated source separation advantage of our approach, we consider a natural image separation scenario. For this example, we have three RGB images with sizes $324 \times 432 \times 3$ as sources (Figure 4.(a)). The sample Pearson correlation coefficients between the images are $\rho_{12} = 0.263$, $\rho_{13} = 0.066$, $\rho_{23} = 0.333$. We use a random 5×3 mixing matrix whose entries are drawn from i.i.d standard normal distribution. The corresponding mixtures are shown in Figure 4.(b).

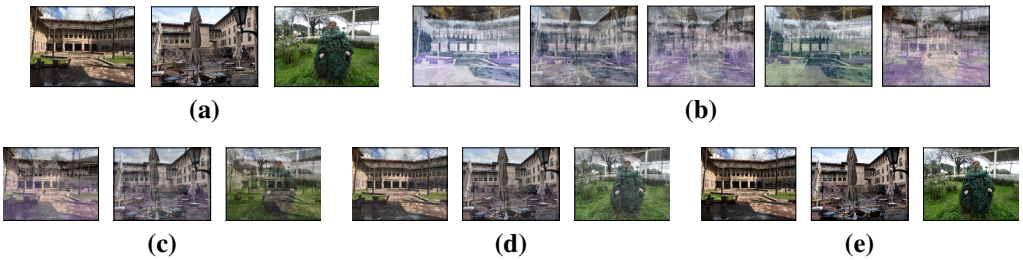


Figure 4: **(a)** Original RGB images, **(b)** Mixture RGB images, **(c)** ICA outputs, **(d)** NSM outputs (using pre-whitened mixtures), **(e)** WSM outputs.

We applied ICA, NSM and WSM algorithms to the mixtures. Figure 4.(c),(d),(e) shows the corresponding outputs. High-resolution versions of all images in this example are available in Appendix E.4 in addition to the comparisons with LD-Infomax and PMF algorithms. The Infomax ICA algorithm's outputs have SINR level of 13.92dB, and this performance is perceivable as residual interference effects in the corresponding output images. The NSM algorithm achieves significantly higher SINR level of 17.45dB and the output images visually reflect this better performance. Our algorithm achieves the best SINR level of 27.49dB, and the corresponding outputs closely resemble the original source images.

7 Discussion and Conclusion

We proposed a general framework for generating biologically plausible neural networks that are capable of separating correlated sources from their linear mixtures, and demonstrated their successful correlated source separation capability through synthetic and natural sources.

Another motivation for our work is to link network structure with function. This is a long standing goal of neuroscience, however examples where this link can be achieved are limited. Our work provides concrete examples where clear links between a network's architecture–i.e. number of interneurons, connections between interneurons and output neurons, nonlinearities (frequency-current curves)– and its function, the type of source separation or feature extraction problem the networks solves, can be established. These links may provide insights and interpretations that might generalize to real biological circuits.

Our networks suffer from the same limitations of other recurrent biologically-plausible BSS networks. First, certain hyperparameters can significantly influence algorithm performance (see Appendix E.9). Especially, the inner product gains (D_{ii}) are sensitive to the combined choices of algorithm parameters, which require careful tuning. Second, the numerical experiments with our neural networks are relatively slow due to the recursive computations in (9)-(10) and (11)-(12) for hidden layer and output vectors, which is common to all biologically plausible recurrent source separation networks (see Appendix F). This could perhaps be addressed by early-stopping the recursive computation [60].

Acknowledgments and Disclosure of Funding

This work/research was supported by KUIS AI Center Research Award. C. Pehlevan acknowledges support from the Intel Corporation.

References

[1] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[2] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[3] Adelbert W Bronkhorst. The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica united with Acustica*, 86(1):117–128, 2000.

[4] Michael S Lewicki. Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–363, 2002.

[5] Hiroki Asari, Barak A Pearlmutter, and Anthony M Zador. Sparse representations for the cocktail party problem. *Journal of Neuroscience*, 26(28):7477–7490, 2006.

[6] Rajiv Narayan, Virginia Best, Erol Ozmeral, Elizabeth McClaine, Micheal Dent, Barbara Shinn-Cunningham, and Kamal Sen. Cortical interference effects in the cocktail party problem. *Nature neuroscience*, 10(12):1601–1607, 2007.

[7] Mark A Bee and Christophe Micheyl. The cocktail party problem: what is it? how can it be solved? and why should animal behaviorists study it? *Journal of comparative psychology*, 122 (3):235, 2008.

[8] Josh H McDermott. The cocktail party problem. *Current Biology*, 19(22):R1024–R1027, 2009.

[9] Nima Mesgarani and Edward F Chang. Selective cortical representation of attended speaker in multi-talker speech perception. *Nature*, 485(7397):233–236, 2012.

[10] Elana M Zion Golumbic, Nai Ding, Stephan Bickel, Peter Lakatos, Catherine A Schevon, Guy M McKhann, Robert R Goodman, Ronald Emerson, Ashesh D Mehta, Jonathan Z Simon, et al. Mechanisms underlying selective neuronal tracking of attended speech at a "cocktail party". *Neuron*, 77(5):980–991, 2013.

[11] Takuya Isomura, Kiyoshi Kotani, and Yasuhiko Jimbo. Cultured cortical neurons can perform blind source separation according to the free-energy principle. *PLoS Comput Biol*, 11(12): e1004643, 2015.

[12] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[13] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[14] Jitendra Sharma, Alessandra Angelucci, and Mriganka Sur. Induction of visual orientation modules in auditory cortex. *Nature*, 404(6780):841–847, 2000.

[15] David M. Eagleman, Olivier J-M. D. Coenen, Vladimir Mitsner, Thomas M. Bartol, Anthony J. Bell, and Terrence J. Sejnowski. Cerebellar glomeruli: Does limited extracellular calcium implement a sparse encoding strategy? In *Proceedings of the 8th Annual Joint Symposium on Neural Computation*, 2001.

[16] Cengiz Pehlevan, Sreyas Mohan, and Dmitri B Chklovskii. Blind nonnegative source separation using biological neural networks. *Neural computation*, 29(11):2925–2954, 2017.

[17] Takuya Isomura and Taro Toyoizumi. Error-gated hebbian rule: A local learning rule for principal and independent component analysis. *Scientific reports*, 8(1):1–11, 2018.

[18] Alper T Erdogan and Cengiz Pehlevan. Blind bounded source separation using neural networks with local learning rules. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3812–3816. IEEE, 2020.

[19] Yanis Bahroun, Dmitri Chklovskii, and Anirvan Sengupta. A normative and biologically plausible algorithm for independent component analysis. *Advances in Neural Information Processing Systems*, 34, 2021.

[20] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[21] Yilun Zhang and Tatyana O Sharpee. A robust feedforward model of the olfactory system. *PLoS computational biology*, 12(4):e1004850, 2016.

[22] Kamesh Krishnamurthy, Ann M Hermundstad, Thierry Mora, Aleksandra M Walczak, and Vijay Balasubramanian. Disorder and the neural representation of complex odors: smelling in the real world. *arXiv preprint arXiv:1707.01962*, 2017.

[23] Vijay Singh, Martin Tchernookov, and Vijay Balasubramanian. What the odor is not: Estimation by elimination. *Physical Review E*, 104(2):024415, 2021.

[24] David Lipshutz, Cengiz Pehlevan, and Dmitri B Chklovskii. Biologically plausible single-layer networks for nonnegative independent component analysis. *Biological Cybernetics*, pages 1–12, 2022.

[25] Chia-Hsiang Lin, Wing-Kin Ma, Wei-Chiang Li, Chong-Yung Chi, and ArulMurugan Ambikapathi. Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case. *IEEE Transactions on Geoscience and Remote Sensing*, 53 (10):5530–5546, 2015.

[26] Gokcan Tatli and Alper T. Erdogan. Polytopic matrix factorization: Determinant maximization based criterion and identifiability. *IEEE Transactions on Signal Processing*, 69:5431–5447, 2021. doi: 10.1109/TSP.2021.3112918.

[27] Reinhard Schachtner, Gerhard Pöppel, and Elmar Wolfgang Lang. Towards unique solutions of non-negative matrix factorization problems by a determinant criterion. *Digital Signal Processing*, 21(4):528–534, 2011.

[28] Alper T Erdogan. A class of bounded component analysis algorithms for the separation of both independent and dependent sources. *IEEE Transactions on Signal Processing*, 61(22): 5730–5743, 2013.

[29] Huseyin A Inan and Alper T Erdogan. A convolutive bounded component analysis framework for potentially nonstationary independent and/or dependent sources. *IEEE Transactions on Signal Processing*, 63(1):18–30, 2014.

[30] Eren Babatas and Alper T Erdogan. An algorithmic framework for sparse bounded component analysis. *IEEE Transactions on Signal Processing*, 66(19):5194–5205, 2018.

[31] Cengiz Pehlevan and Dmitri B Chklovskii. A normative theory of adaptive dimensionality reduction in neural networks supplementary information. *Proceedings of Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

[32] Cengiz Pehlevan and Dmitri B Chklovskii. Neuroscience-inspired online unsupervised learning algorithms: Artificial neural networks. *IEEE Signal Processing Magazine*, 36(6):88–96, 2019.

[33] Ji-Cheng Chen. The nonnegative rank factorizations of nonnegative matrices. *Linear Algebra and Its Applications*, 62:207–217, November 1984.

[34] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[35] Xiao Fu, Kejun Huang, Nicholas D Sidiropoulos, and Wing-Kin Ma. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Process. Mag.*, 36(2):59–80, 2019.

[36] Tsung-Han Chan, Wing-Kin Ma, ArulMurugan Ambikapathi, and Chong-Yung Chi. A simplex volume maximization framework for hyperspectral endmember extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4177–4193, 2011.

[37] Chia-Hsiang Lin, Ruiyuan Wu, Wing-Kin Ma, Chong-Yung Chi, and Yue Wang. Maximum volume inscribed ellipsoid: A new simplex-structured matrix factorization framework via facet enumeration and convex optimization. *SIAM Journal on Imaging Sciences*, 11(2):1651–1679, 2018.

[38] Pando Georgiev, Fabian Theis, and Andrzej Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE transactions on neural networks*, 16(4): 992–996, 2005.

[39] David L Donoho. For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics*, 59(7):907–934, 2006.

[40] Christopher Rozell, Don Johnson, Richard Baraniuk, and Bruno Olshausen. Locally competitive algorithms for sparse approximation. In *2007 IEEE International Conference on Image Processing*, volume 4, pages IV–169. IEEE, 2007.

[41] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[42] Eren Babatas and Alper T Erdogan. Time and frequency based sparse bounded component analysis algorithms for convolutive mixtures. *Signal Processing*, page 107590, 2020.

[43] Sergio Cruces. Bounded component analysis of linear mixtures: A criterion of minimum convex perimeter. *IEEE Transactions on Signal Processing*, 58(4):2141–2154, 2010.

[44] Gokcan Tatli and Alper T. Erdogan. Generalized polytopic matrix factorization. *2021 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, June 2021.

[45] Bariscan Bozkurt and Alper T. Erdogan. On identifiable polytope characterization for polytopic matrix factorization. *2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2022.

[46] Cengiz Pehlevan and Dmitri B Chklovskii. A hebbian/anti-hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 769–775. IEEE, 2014.

[47] Anirvan Sengupta, Cengiz Pehlevan, Mariano Tepper, Alexander Genkin, and Dmitri Chklovskii. Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks. In *Advances in Neural Information Processing Systems*, pages 7080–7090, 2018.

[48] Cengiz Pehlevan. A spiking neural network with local learning rules derived from nonnegative similarity matching. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7958–7962. IEEE, 2019.

[49] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003.

[50] Yuanqing Li, Andrzej Cichocki, and Shun-ichi Amari. Analysis of sparse representation and blind source separation. *Neural computation*, 16(6):1193–1234, 2004.

[51] Clément Elvira, Pierre Chainais, and Nicolas Dobigeon. Bayesian antisparse coding. *IEEE Transactions on Signal Processing*, 65(7):1660–1672, 2016.

[52] Christoph Studer, Tom Goldstein, Wotao Yin, and Richard G Baraniuk. Democratic representations. *arXiv preprint arXiv:1401.3420*, 2014.

[53] David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *Advances in neural information processing systems*, 16: 1141–1148, 2003.

[54] Agnieszka Grabska-Barwińska, Simon Barthelmé, Jeff Beck, Zachary F Mainen, Alexandre Pouget, and Peter E Latham. A probabilistic approach to demixing odors. *Nature neuroscience*, 20(1):98–106, 2017.

[55] Xiao Fu, Kejun Huang, and Nicholas D Sidiropoulos. On identifiability of nonnegative matrix factorization. *IEEE Signal Processing Letters*, 25(3):328–332, 2018.

[56] Christopher J Rozell, Don H Johnson, Richard G Baraniuk, and Bruno A Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10): 2526–2563, 2008.

[57] Gina G Turrigiano. Homeostatic plasticity in neuronal networks: the more things change, the more they stay the same. *Trends in neurosciences*, 22(5):221–227, 1999.

[58] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7 (267):1–13, 2013. doi: 10.3389/fnins.2013.00267.

[59] Alper T Erdogan. An information maximization based blind source separation approach for dependent and independent sources. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4378–4382. IEEE, 2022.

[60] Victor Minden, Cengiz Pehlevan, and Dmitri B Chklovskii. Biologically plausible online principal component analysis without recurrent neural dynamics. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 104–111. IEEE, 2018.

[61] Cengiz Pehlevan, Alexander Genkin, and Dmitri B Chklovskii. A clustering neural network model of insect olfaction. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pages 593–600. IEEE, 2017.

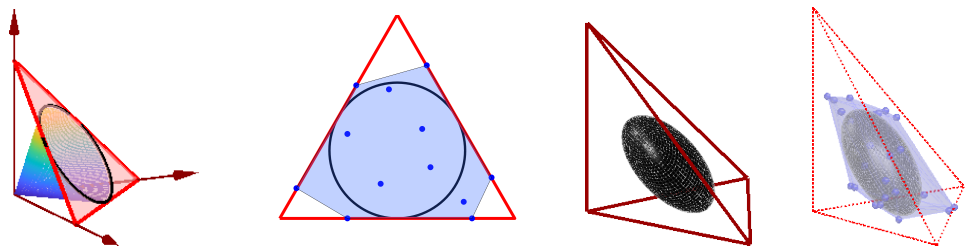
Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We noticeably express the contributions and scope of the paper, i.e., providing a general framework for constructing biologically plausible neural networks for separating both independent and correlated sources while covering more generic source domains.
   (b) Did you describe the limitations of your work? [Yes] See Section 7
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We strongly believe that this work does not have any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the ethics review guidelines and our article conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Section 2 provides the underlying assumptions for our theoretical results.
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide the complete proof of Theorem 1 in the appendix. Moreover, we include the detailed derivations of the proposed network dynamics in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Yes, we include the code in the supplemental material with tutorial Python notebooks and multiple-run simulation scripts for our experiments in addition to README.md file.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the appendix, we provide the hyperparameter selections and training details of the experiments. We also share our code for numerical experiments.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In the appendix, we provide the SINR convergence behaviors for several experiments based on multiple realizations with corresponding percentile envelopes.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We did not discuss the computational power and memory concerns since our method does not require an advanced computing system. An individual source separation experiment can run on a basic computer.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We refer to [20] and link to the data in E.6 for image patches we used for sparse dictionary learning.
   (b) Did you mention the license of the assets? [No] We were unable to find the official license of the image patches we used for sparse dictionary learning. However, we cite the creators of the data [20] and give the link for the image patches in E.6.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Our code is the only new asset which we include in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A On sufficient scattering condition for source vectors

The determinant maximization criterion used in matrix factorization frameworks is based on the assumption that the latent factors are sufficiently scattered in their presumed domain to somewhat reflect its shape. Both simplex structure matrix factorization and polytopic matrix factorization frameworks propose precise sufficient scattering conditions for the latent vectors to guarantee their identifiability under the determinant maximization criterion. In this section, we briefly summarize these conditions.



(a) The unit simplex Δ and the second order cone $\mathcal{C}$.

(b) An illustration of the sufficient scattering condition for Δ samples.

(c) The polytope $\mathcal{B}_{1,+}$ and its MVIE $\mathcal{E}_{\mathcal{B}_1}$.

(d) An illustration of the sufficient scattering condition for $\mathcal{B}_{1,+}$ samples.

Figure 5: The geometry of sufficient scattering conditions for the unit simplex and polytopes illustrated in three-dimensions.

Sufficient scattering condition for unit simplex sources: An earlier latent factor identfiability assumption for SSMF required the inclusion of the vertices of the unit simplex in the generative latent vector samples [53]. This, so-called *separability* or *local dominance*, assumption was later replaced by a weaker *sufficiently scattered condition* (SSC) in [55]. This new condition uses the second order cone

$$\mathcal{C} = \{\mathbf{s} \,|\, \mathbf{1}^T\mathbf{s} \geq \sqrt{n-1}\|\mathbf{s}\|_2, \mathbf{s} \in \mathbb{R}^n\},$$

which is illustrated in Figure 5.(a) together with the unit simplex Δ, as a reference object for defining SSC. The SSC proposed in [55] for SSMF requires that conic hull of the simplex samples contains $\mathcal{C}$, i.e.,

$$\text{cone}(\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t\}) \supseteq \mathcal{C}.$$

Let $\mathcal{A}_\Delta$ represent the affine hull of Δ. Figure 5.(b) illustrates this requirement restricted to $\mathcal{A}_\Delta$: the red triangle is the boundary of Δ, the blue dots are sufficiently scattered samples from Δ, the black circle and the blue polyhedral region are the boundary of $\mathcal{C}$ and the conic hull of sufficiently scattered samples from Δ restricted to $\mathcal{A}_\Delta$, respectively. There is an additional requirement that the boundaries of Δ and $\text{cone}(\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t\}) \cap \mathcal{A}_\Delta$ intersect the boundary of $\mathcal{C} \cap \mathcal{A}_\Delta$ at the identical points.

Sufficient scattering condition for polytopic sources: The reference [26] offers a similar SSC for polytopic sources for which the reference object for SSC is the maximum volume inscribed ellipsoid (MVIE), represented by $\mathcal{E}_\mathcal{P}$ of the polytope $\mathcal{P}$. Figure 5.(c) illustrates MVIE (the black ellipsoid) for the polytope selection $\mathcal{P} = \mathcal{B}_{1,+}$ whose edges are the red lines. The SSC for polytopic sources require that convex hull of the polytopic samples contain the polytope's MVIE,i.e.,

$$\text{conv}(\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t\}) \supseteq \mathcal{E}_\mathcal{P}.$$

This condition is illustrated in Figure 5.(d), where the dots represent sufficiently scattered samples and the blue polyhedral region is their convex hull. The SSC in [26] further require that the boundaries of $\mathcal{P}$ and $\text{conv}(\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t\})$ intersect $\mathcal{E}_\mathcal{P}$ at the identical points.

B Proof of theorem 1

The proof of Theorem 1 relies on the following lemma, which follows from equality constraints (4b) and (4c):

Lemma 1. *Given the mixing model in Section 2.2, for sufficiently large sample sizes enabling full column-rank condition on $\mathbf{X}(t)$, the constraints in (4b) and (4c) define an arbitrary linear mapping between input and output vectors in the form*

$$\mathbf{y}_i = \mathbf{W}(t)\mathbf{x}_i, \qquad i \in \{1, \dots, t\}. \tag{A.1}$$

where $\mathbf{W}(t)$ is full-rank.

Proof of Lemma 1. To see the relation between $\mathbf{h}_i$ and $\mathbf{x}_i$, note that mixing relation (2) and equality constraint (4b) enforces $\mathbf{S}(t)^T\mathbf{A}^T\mathbf{A}\mathbf{S}(t) = \mathbf{H}(t)^T\mathbf{D}_1(t)\mathbf{H}(t)$. Defining $\mathbf{A} = \mathbf{U}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}\mathbf{V}_\mathbf{A}^T$ as the reduced SVD decomposition for $\mathbf{A}$ matrix with $\mathbf{U}_\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{\Sigma}_\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{V}_\mathbf{A} \in \mathbb{R}^{n\times n}$, we can write $\mathbf{A}^T\mathbf{A} = \mathbf{V}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}^2\mathbf{V}_\mathbf{A}^T$. For sufficiently large sample sizes enabling full rank $\mathbf{X}(t)$, these imply $\mathbf{h}_i = \mathbf{D}_1(t)^{-1/2}\mathbf{Q}_1^T(t)\mathbf{\Sigma}_\mathbf{A}\mathbf{V}_\mathbf{A}^T\mathbf{s}_i$, $i \in \{1, \dots, t\}$, for some real-orthogonal matrix $\mathbf{Q}_1(t)$. From this expression, we can also write $\mathbf{h}_i = \mathbf{D}_1(t)^{-1/2}\mathbf{Q}_1^T(t)\mathbf{U}_\mathbf{A}^T\mathbf{x}_i = \mathbf{D}_1(t)^{-1/2}\mathbf{\Theta}_1^T(t)\mathbf{x}_i$, $i \in \{1, \dots, t\}$, where $\mathbf{\Theta}_1(t) = \mathbf{U}_\mathbf{A}\mathbf{Q}_1(t) \in \mathbb{R}^{m\times n}$ is a matrix with orthonormal columns.

The weighted inner product matching condition in (4c) implies that the output $\mathbf{y}_i$'s are related to the the slack vectors $\mathbf{h}_i$'s through the relationship $\mathbf{y}_i = \mathbf{D}_2(t)^{-1/2}(t)\mathbf{\Theta}_2(t)\mathbf{h}_i$, $i \in \{1, \dots, t\}$, where $\mathbf{\Theta}_2(t)$ is another real-orthogonal matrix. Consequently $\mathbf{y}_i = \mathbf{D}_2(t)^{-1/2}\mathbf{\Theta}_2(t)\mathbf{D}_1(t)^{-1/2}\mathbf{\Theta}_1^T(t)\mathbf{x}_i$, $i \in \{1, \dots, t\}$. Here, the multiplier of $\mathbf{x}_i$ is in the form of a Singular Value Decomposition of a full rank matrix, i.e., $\mathbf{\Theta}_2(t)\mathbf{D}_1(t)^{-1/2}\mathbf{\Theta}_1^T(t)$ which is left multiplied by a full rank diagonal matrix, i.e., $\mathbf{D}_2(t)^{-1/2}$. This implies that the equality constraints (4b) and (4c) in conjunction define an arbitrary linear mapping between input and output vectors, through inner product matching. □

Now we can prove Theorem 1.

Proof of Theorem 1. By Lemma 1, $\mathbf{y}_i = \mathbf{W}(t)\mathbf{x}_i$, $i = 1, \dots t$, where $\mathbf{W}(t)$ admits the form $\mathbf{W}(t) = \mathbf{D}_2(t)^{-1/2}\mathbf{\Theta}_2(t)\mathbf{D}_1(t)^{-1/2}\mathbf{\Theta}_1^T(t)$. Therefore, using the mixture-source relationship $\mathbf{x}_i = \mathbf{A}\mathbf{s}_i$, we can write $\mathbf{y}_i = \mathbf{G}(t)\mathbf{s}_i$, where $\mathbf{G}(t) = \mathbf{W}(t)\mathbf{A}$ is the linear mapping relationship between outputs and sources. Plugging this into maximization objective in (3), we obtain $\log(\det(\mathbf{Y}(t)\mathbf{Y}(t)^T)) = 2\log(|\det(\mathbf{G}(t))|) + \log(\det(\mathbf{S}(t)\mathbf{S}(t)^T))$.To proceed, we define $\mathbf{A} = \mathbf{U}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}\mathbf{V}_\mathbf{A}^T$ as the reduced SVD for matrix $\mathbf{A}$ to obtain $\mathbf{G}(t) = \mathbf{D}_2(t)^{-1/2}\mathbf{\Theta}_2(t)\mathbf{D}_1(t)^{-1/2}\mathbf{\Theta}_1(t)^T\mathbf{U}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}\mathbf{V}_\mathbf{A}^T$. Consequently, $2\log(|\det(\mathbf{G}(t))|) = -\log(\det(\mathbf{D}_1)) - \log(\det(\mathbf{D}_2)) + 2\log(\det(\mathbf{\Sigma}_\mathbf{A}))$. As a result, maximizing the objective in (3) is equivalent to minimizing $\log(\det(\mathbf{D}_1)) + \log(\det(\mathbf{D}_2))$ with some additional constant terms. Since $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal, the equivalent function can be written as $\sum_{i=1}^n \log(D_{1,ii}(t)) + \sum_{i=1}^n \log(D_{2,ii}(t))$, which is the objective function in (4a). □

C Derivations

C.1 The simplification of the similarity matching cost functions

In this section, we provide the simplification of the similarity matching cost functions J_1 and J_2 in Section 5.1, by preserving only the quadratic terms that are relevant to online optimization with respect to $\mathbf{h}_t$ and $\mathbf{y}_t$.

Using the matrix partitions $\boldsymbol{\mathcal{X}}(t) = [\ \gamma\boldsymbol{\mathcal{X}}(t-1) \quad \mathbf{x}_t\ ]$ and $\boldsymbol{\mathcal{H}}(t) = [\ \gamma\boldsymbol{\mathcal{H}}(t-1) \quad \mathbf{h}_t\ ]$, we can write $J_1(\mathbf{H}(t), \mathbf{D}_1(t))$ more explicitly as

$$\begin{aligned}
&J_1(\mathbf{H}(t), \mathbf{D}_1(t)) \\
&= \frac{\gamma^2}{2\tau^2}\left\|\begin{bmatrix} \gamma\boldsymbol{\mathcal{X}}(t-1)^T\boldsymbol{\mathcal{X}}(t-1) - \gamma\boldsymbol{\mathcal{H}}(t-1)^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t-1) & \boldsymbol{\mathcal{X}}(t-1)^T\mathbf{x}_t - \boldsymbol{\mathcal{H}}(t-1)^T\mathbf{D}_1(t)\mathbf{h}_t \\ \mathbf{x}_t^T\boldsymbol{\mathcal{X}}(t-1) - \mathbf{h}_t^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t-1) & \frac{\|\mathbf{x}_t\|_2^2 - \|\mathbf{h}_t\|_{\mathbf{D}_1(t)}^2}{\gamma} \end{bmatrix}\right\|_F^2 \\
&= \frac{\gamma^4}{\tau^2}\|\boldsymbol{\mathcal{X}}(t-1)^T\boldsymbol{\mathcal{X}}(t-1)\|_F^2 + \frac{\gamma^4}{\tau^2}\|\boldsymbol{\mathcal{H}}(t-1)^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t-1)\|_F^2 \\
&- \frac{2\gamma^4}{\tau^2}\mathrm{Tr}(\boldsymbol{\mathcal{H}}(t-1)^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t-1)\boldsymbol{\mathcal{X}}(t-1)^T\boldsymbol{\mathcal{X}}(t-1)) \\
&+ \frac{2\gamma^2}{\tau^2}\|\mathbf{x}_t^T\boldsymbol{\mathcal{X}}(t-1)\|_F^2 + \frac{2\gamma^2}{\tau^2}\|\mathbf{h}_t^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t-1)\|_F^2 \\
&- \frac{4\gamma^2}{\tau^2}\mathbf{h}_t^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t-1)\boldsymbol{\mathcal{X}}(t-1)^T\mathbf{x}_t + \frac{1}{\tau^2}(\|\mathbf{x}_t\|^4 + \|\mathbf{h}_t\|_{\mathbf{D}_1(t)}^4 - 2\|\mathbf{x}_t\|^2\|\mathbf{h}_t\|_{\mathbf{D}_1(t)}^2).
\end{aligned}$$

By keeping only the relevant part of this cost function for online optimization with respect to $\mathbf{h}_t$, by scaling with τ/γ^2, and ignoring the small final term, we obtain the effective online cost function corresponding to J_1 as

$$c_1(\mathbf{h}_t) \quad = \quad 2\mathbf{h}_t^T\mathbf{D}_1\mathbf{M}_H(t)\mathbf{D}_1(t)\mathbf{h}_t - 4\mathbf{h}_t^T\mathbf{D}_1(t)\mathbf{W}_{HX}(t)\mathbf{x}_t,$$

where

$$\mathbf{M}_H(t) = \frac{1}{\tau}\boldsymbol{\mathcal{H}}(t-1)\boldsymbol{\mathcal{H}}(t-1)^T = \frac{1}{\tau}\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{h}_k\mathbf{h}_k^T \tag{A.2}$$

$$\mathbf{W}_{HX}(t) = \frac{1}{\tau}\boldsymbol{\mathcal{H}}(t-1)\boldsymbol{\mathcal{X}}(t-1)^T = \frac{1}{\tau}\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{h}_k\mathbf{x}_k^T. \tag{A.3}$$

If we apply the same procedure to $J_2(\mathbf{H}(t), \mathbf{D}_2(t), \mathbf{Y}(t))$:

$$\begin{aligned}
&J_2(\mathbf{H}(t), \mathbf{D}_2(t), \mathbf{Y}(t)) \\
&= \frac{\gamma^2}{2\tau^2}\left\|\begin{bmatrix} \gamma\boldsymbol{\mathcal{H}}(t-1)^T\boldsymbol{\mathcal{H}}(t-1) - \gamma\boldsymbol{\mathcal{Y}}(t-1)^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t-1) & \boldsymbol{\mathcal{H}}(t-1)^T\mathbf{h}_t - \boldsymbol{\mathcal{Y}}(t-1)^T\mathbf{D}_2(t)\mathbf{y}_t \\ \mathbf{h}_t^T\boldsymbol{\mathcal{H}}(t-1) - \mathbf{y}_t^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t-1) & \frac{\|\mathbf{h}_t\|_2^2 - \|\mathbf{y}_t\|_{\mathbf{D}_2(t)}^2}{\gamma} \end{bmatrix}\right\|_F^2 \\
&= \frac{\gamma^4}{\tau^2}\|\boldsymbol{\mathcal{H}}(t-1)^T\boldsymbol{\mathcal{H}}(t-1)\|_F^2 + \frac{\gamma^4}{\tau^2}\|\boldsymbol{\mathcal{Y}}(t-1)^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t-1)\|_F^2 \\
&-\frac{2\gamma^4}{\tau^2}\mathrm{Tr}(\boldsymbol{\mathcal{Y}}(t-1)^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t-1)\boldsymbol{\mathcal{H}}(t-1)^T\boldsymbol{\mathcal{H}}(t-1)) \\
&+\frac{2\gamma^2}{\tau^2}\|\mathbf{h}_t^T\boldsymbol{\mathcal{H}}(t-1)\|_F^2 + \frac{2\gamma^2}{\tau^2}\|\mathbf{y}_t^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t-1)\|_F^2 \\
&-\frac{4\gamma^2}{\tau^2}\mathbf{y}_t^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t-1)\boldsymbol{\mathcal{H}}(t-1)^T\mathbf{h}_t + \frac{1}{\tau^2}(\|\mathbf{h}_t\|^4 + \|\mathbf{y}_t\|_{\mathbf{D}_2(t)}^4 - 2\|\mathbf{h}_t\|^2\|\mathbf{y}_t\|_{\mathbf{D}_2(t)}^2).
\end{aligned}$$

Similar to J_1, we can simplify the part of the J_2 cost function that is dependent on $\mathbf{h}_t$ and $\mathbf{y}_t$ as

$$c_2(\mathbf{h}_t, \mathbf{y}_t) \quad = \quad 2\mathbf{y}_t^T\mathbf{D}_2(t)\mathbf{M}_Y(t)\mathbf{D}_2(t)\mathbf{y}_t - 4\mathbf{y}_t^T\mathbf{D}_2(t)\mathbf{W}_{YH}(t)\mathbf{h}_t + 2\mathbf{h}_t^T\mathbf{M}_H(t)\mathbf{h}_t,$$

where

$$\mathbf{W}_{YH}(t) \quad = \quad \frac{1}{\tau}\boldsymbol{\mathcal{Y}}(t-1)\boldsymbol{\mathcal{H}}(t-1)^T = \frac{1}{\tau}\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{y}_k\mathbf{h}_k^T, \tag{A.4}$$

$$\mathbf{M}_Y(t) \quad = \quad \frac{1}{\tau}\boldsymbol{\mathcal{Y}}(t-1)\boldsymbol{\mathcal{Y}}(t-1)^T = \frac{1}{\tau}\sum_{k=1}^{t-1}(\gamma^2)^{t-1-k}\mathbf{y}_k\mathbf{y}_k^T. \tag{A.5}$$

As a result, we can write the effective online cost function $\mathcal{J}$, corresponding to $\mathbf{h}_t$ and $\mathbf{y}_t$ as

$$C(\mathbf{h}_t, \mathbf{y}_t) = \beta c_1(\mathbf{h}_t) + (1-\beta)c_2(\mathbf{h}_t, \mathbf{y}_t). \tag{A.6}$$

C.2 Derivatives of the WSM cost function

In this section, we provide the expressions for the gradients of the online WSM cost function $\mathcal{J}$ in (5) to be used in the descent algorithm formulation in Section 5.2 and Appendix D. For the gradients with respect to $\mathbf{h}_t$ and $\mathbf{y}_t$, we use $C(\mathbf{h}_t, \mathbf{y}_t)$ in (A.6), which is the simplified version of $\mathcal{J}$, as derived in Section C.1.

- The (scaled) gradient with respect to $\mathbf{h}_t$:

$$\begin{aligned}
\frac{1}{4}\nabla_{\mathbf{h}_t}\mathcal{J}(\mathbf{H}(t), \mathbf{D}_1(t), \mathbf{D}_2(t), \mathbf{Y}(t)) &= \frac{1}{4}\nabla_{\mathbf{h}_t}C(\mathbf{h}_t, \mathbf{y}_t) \\
&= ((1-\beta)\mathbf{M}_H(t) + \beta\mathbf{D}_1(t)\mathbf{M}_H(t)\mathbf{D}_1(t))\mathbf{h}_t \\
&\quad -\beta\mathbf{D}_1(t)\mathbf{W}_{HX}(t)\mathbf{x}_t - (1-\beta)\mathbf{W}_{YH}(t)^T\mathbf{D}_2(t)\mathbf{y}_t.
\end{aligned} \tag{A.7}$$

By applying the decomposition $\mathbf{M}_H(t) = \bar{\mathbf{M}}_H(t) + \boldsymbol{\Gamma}_H(t)$, where

$$\boldsymbol{\Gamma}_H(t) \quad = \quad \mathrm{diag}(\mathbf{M}_{H11}(t), \mathbf{M}_{H22}(t), \ldots, \mathbf{M}_{Hdd}(t)),$$

we can rewrite the gradient expression in (A.7) as

$$\frac{1}{4}\nabla_{\mathbf{h}_t} C(\mathbf{h}_t, \mathbf{y}_t) \quad = \mathbf{v}_t + ((1-\beta)\bar{\mathbf{M}}_H(t) + \beta \mathbf{D}_1(t)\bar{\mathbf{M}}_H(t)\mathbf{D}_1(t))\mathbf{h}_t$$
$$-\beta \mathbf{D}_1(t)\mathbf{W}_{HX}(t)\mathbf{x}_t - (1-\beta)\mathbf{W}_{YH}(t)^T\mathbf{D}_2(t)\mathbf{y}_t. \tag{A.8}$$

In (A.8), we used the substitution,

$$\mathbf{v}_t = ((1-\beta)\mathbf{\Gamma}_H(t) + \beta \mathbf{D}_1(t)\mathbf{\Gamma}_H(t)\mathbf{D}_1(t)))\mathbf{h}_t. \tag{A.9}$$

- The gradient with respect to $\mathbf{y}_t$:

$$\frac{1}{4}\nabla_{\mathbf{y}_t} C(\mathbf{h}_t, \mathbf{y}_t) = (1-\beta)(-\mathbf{D}_2(t)\mathbf{W}_{YH}(t)\mathbf{h}_t + \mathbf{D}_2(t)\mathbf{M}_Y(t)\mathbf{D}_2(t)\mathbf{y}_t).$$

Note that, since $\mathbf{D}_2(t)$ is positive,

$$-\frac{1}{4(1-\beta)}\mathbf{D}_2(t)^{-1}\nabla_{\mathbf{y}_t}\mathcal{J}(\mathbf{h}_t, \mathbf{y}_t) = \mathbf{W}_{YH}(t)\mathbf{h}_t - \mathbf{M}_Y(t)\mathbf{D}_2(t)\mathbf{y}_t \tag{A.10}$$

is a descent direction. Furthermore, by decomposing $\mathbf{M}_Y(t) = \bar{\mathbf{M}}_Y(t) + \mathbf{\Gamma}_Y(t)$, where

$$\mathbf{\Gamma}_Y(t) \quad = \quad \text{diag}(\mathbf{M}_{Y11}(t), \mathbf{M}_{Y22}(t), \ldots, \mathbf{M}_{Ydd}(t)),$$

we can rewrite the the descent direction in (A.10) as

$$-\frac{1}{4(1-\beta)}\mathbf{D}_2(t)^{-1}\nabla_{\mathbf{y}_t} C(\mathbf{h}_t, \mathbf{y}_t) = -\mathbf{u}_t + \mathbf{W}_{YH}(t)\mathbf{h}_t - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}_t, \tag{A.11}$$

where we substituted

$$\mathbf{u}_t = \mathbf{\Gamma}_Y(t)\mathbf{D}_2(t)\mathbf{y}_t. \tag{A.12}$$

- The derivative with respect to $D_{1,ii}(t)$:

$$\begin{aligned}
&\frac{\partial \mathcal{J}(\mathbf{h}_t, \mathbf{y}_t, \mathbf{D}_1(t), \mathbf{D}_2(t))}{\partial D_{1,ii}(t)} \\
&= \lambda_{SM}\beta \text{Tr}((\boldsymbol{\mathcal{H}}(t)^T\mathbf{E}_{ii}\boldsymbol{\mathcal{H}}(t))^T(\boldsymbol{\mathcal{H}}(t)^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t) - \boldsymbol{\mathcal{X}}(t)^T\boldsymbol{\mathcal{X}}(t))) + \frac{1-\lambda_{SM}}{D_{1,ii}(t)} \\
&= \lambda_{SM}\beta \text{Tr}(\boldsymbol{\mathcal{H}}(t)_{i,:}^T\boldsymbol{\mathcal{H}}(t)_{i,:}(\boldsymbol{\mathcal{H}}(t)^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t) - \boldsymbol{\mathcal{X}}(t)^T\boldsymbol{\mathcal{X}}(t))) + \frac{1-\lambda_{SM}}{D_{1,ii}(t)} \\
&= \lambda_{SM}\beta(\boldsymbol{\mathcal{H}}(t)_{i,:}\boldsymbol{\mathcal{H}}(t)^T\mathbf{D}_1(t)\boldsymbol{\mathcal{H}}(t)\boldsymbol{\mathcal{H}}(t)_{i,:}^T - \boldsymbol{\mathcal{H}}(t)_{i,:}\boldsymbol{\mathcal{X}}(t)^T\boldsymbol{\mathcal{X}}(t))\boldsymbol{\mathcal{H}}(t)_{i,:}^T) + \frac{1-\lambda_{SM}}{D_{1,ii}(t)} \\
&= \lambda_{SM}\beta(\|\mathbf{M}_{Hi,:}\|^2_{\mathbf{D}_1(t)} - \|\mathbf{W}_{HXi,:}\|_2^2) + \frac{1-\lambda_{SM}}{D_{1,ii}(t)}.
\end{aligned} \tag{A.13}$$

- The derivative with respect to $D_{2,ii}(t)$:

$$\begin{aligned}
&\frac{\partial \mathcal{J}(\mathbf{h}_t, \mathbf{y}_t, \mathbf{D}_1(t), \mathbf{D}_2(t))}{\partial D_{2,ii}(t)} \\
&= \lambda_{SM}(1-\beta)\text{Tr}((\boldsymbol{\mathcal{Y}}(t)^T\mathbf{E}_{ii}\boldsymbol{\mathcal{Y}}(t))^T(\boldsymbol{\mathcal{Y}}(t)^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t) - \boldsymbol{\mathcal{H}}(t)^T\boldsymbol{\mathcal{H}}(t))) + \frac{1-\lambda_{SM}}{D_{2,ii}(t)} \\
&= \lambda_{SM}(1-\beta)\text{Tr}(\boldsymbol{\mathcal{Y}}(t)_{i,:}^T\boldsymbol{\mathcal{Y}}(t)_{i,:}(\boldsymbol{\mathcal{Y}}(t)^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t) - \boldsymbol{\mathcal{H}}(t)^T\boldsymbol{\mathcal{H}}(t))) + \frac{1-\lambda_{SM}}{D_{2,ii}(t)} \\
&= \lambda_{SM}(1-\beta)(\boldsymbol{\mathcal{Y}}(t)_{i,:}\boldsymbol{\mathcal{Y}}(t)^T\mathbf{D}_2(t)\boldsymbol{\mathcal{Y}}(t)\boldsymbol{\mathcal{Y}}(t)_{i,:}^T - \boldsymbol{\mathcal{Y}}(t)_{i,:}\boldsymbol{\mathcal{H}}(t)^T\boldsymbol{\mathcal{H}}(t))\boldsymbol{\mathcal{Y}}(t)_{i,:}^T) + \frac{1-\lambda_{SM}}{D_{2,ii}(t)} \\
&= \lambda_{SM}(1-\beta)(\|\mathbf{M}_{Yi,:}\|^2_{\mathbf{D}_2(t)} - \|\mathbf{W}_{YHi,:}\|_2^2) + \frac{1-\lambda_{SM}}{D_{2,ii}(t)}.
\end{aligned} \tag{A.14}$$

D Det-max WSM neural networks for example source domains

The proposed Det-Max WSM framework is applicable to infinitely many source domains corresponding to different assumptions on the sources. In this section, we provide derivations and illustrations of WSM-based Det-Max neural networks for some selected source domains.

D.1 Anti-sparse sources

Section 5.2 covers the derivation of the network dynamics and the learning rules for antisparse sources, i.e., the source domain selection of $\mathcal{P} = \mathcal{B}_\infty$. If we summarize the dynamics equations obtained:

Update dynamics for the hidden layer $\mathbf{h}_t$:

$$\begin{aligned}
\frac{d\mathbf{v}(\tau)}{d\tau} &= -\mathbf{v}(\tau) - \lambda_{SM}[((1-\beta)\bar{\mathbf{M}}_H(t) + \beta\mathbf{D}_1(t)\bar{\mathbf{M}}_H(t)\mathbf{D}_1(t))\mathbf{h}(\tau) \\
&\quad +\beta\mathbf{D}_1(t)\mathbf{W}_{HX}(t)\mathbf{x}(\tau) + (1-\beta)\mathbf{W}_{YH}(t)^T\mathbf{D}_2(t)\mathbf{y}(\tau)] \\
\mathbf{h}_{t,i}(\tau) &= \sigma_A\left(\frac{\mathbf{v}_i(\tau)}{\lambda_{SM}\Gamma_{Hii}(t)((1-\beta)+\beta D_{1,ii}(t)^2)}\right) \quad \text{for } i = 1,\dots n.
\end{aligned}$$

where $\sigma_A(\cdot)$ is the clipping nonlinearity with level A.

Update dynamics for the output $\mathbf{y}_t$:

$$\begin{aligned}
\frac{d\mathbf{u}(\tau)}{d\tau} &= -\mathbf{u}(\tau) + \mathbf{W}_{YH}(t)\mathbf{h}(\tau) - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}(\tau) \\
\mathbf{y}_{t,i}(\tau) &= \sigma_1\left(\frac{\mathbf{u}_i(\tau)}{\Gamma_{Yii}(t)D_{2,ii}(t)}\right), \qquad \text{for } i = 1,\dots n,
\end{aligned}$$

Figure 6 shows the corresponding two-layer neural network.

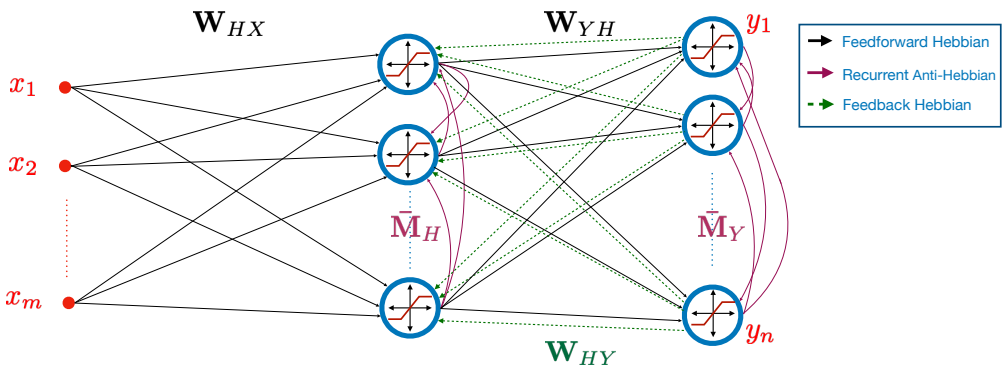


Figure 6: WSM Det-Max neural network for antisparse sources ($\mathcal{P} = \mathcal{B}_\infty$).

D.2 Nonnegative anti-sparse sources

For the case of nonnegative anti-sparse sources, the corresponding network is essentially the same as the antisparse case in Appendix D.1. The only difference is that the clipping activation functions at the output layer are replaced with nonnegative clipping function $\sigma_+(x)$ illustrated in Figure 7.

As a result, we can write the network dynamics corresponding to the nonnegative anti-sparse case as

Update dynamics for the hidden layer $\mathbf{h}_t$:

$$\begin{aligned}
\frac{d\mathbf{v}(\tau)}{d\tau} &= -\mathbf{v}(\tau) - \lambda_{SM}[((1-\beta)\bar{\mathbf{M}}_H(t) + \beta\mathbf{D}_1(t)\bar{\mathbf{M}}_H(t)\mathbf{D}_1(t))\mathbf{h}(\tau) \\
&\quad +\beta\mathbf{D}_1(t)\mathbf{W}_{HX}(t)\mathbf{x}(\tau) + (1-\beta)\mathbf{W}_{YH}(t)^T\mathbf{D}_2(t)\mathbf{y}(\tau)] \\
\mathbf{h}_{t,i}(\tau) &= \sigma_A\left(\frac{\mathbf{v}_i(\tau)}{\lambda_{SM}\Gamma_{Hii}(t)((1-\beta)+\beta D_{1,ii}(t)^2)}\right) \quad \text{for } i = 1,\dots n,
\end{aligned}$$

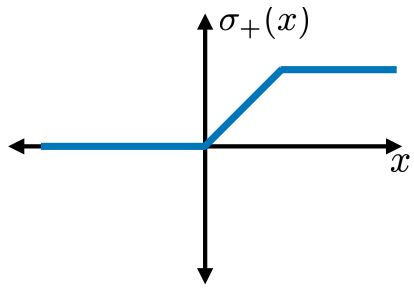


Figure 7: Nonnegative clipping function for elementwise projection to $\mathcal{B}_{\infty,+}$.

Update dynamics for the output $\mathbf{y}_t$:

$$\begin{aligned}\frac{d\mathbf{u}(\tau)}{d\tau} &= -\mathbf{u}(\tau) + \mathbf{W}_{YH}(t)\mathbf{h}(\tau) - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}(\tau) \\ \mathbf{y}_{t,i}(\tau) &= \sigma_1\left(\frac{\mathbf{u}_i(\tau)}{\Gamma_{Y\,ii}(t)D_{2,ii}(t)}\right), \qquad \text{for } i = 1, \ldots n.\end{aligned}$$

The network corresponding to nonnegative anti-sparse sources is shown in Figure 8.

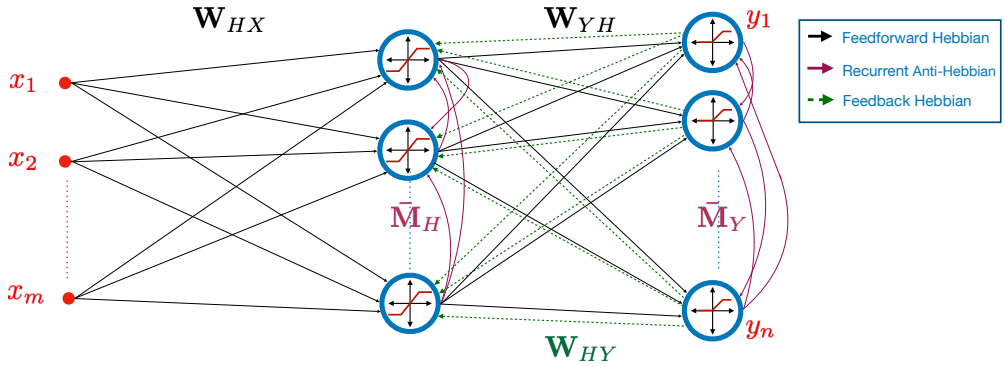


Figure 8: WSM Det-Max neural network for nonnegative anti-sparse sources ($\mathcal{P} = \mathcal{B}_{\infty,+}$).

D.3 Nonnegative sparse sources

For nonnegative sparse sources, i.e., $\mathcal{P} = \mathcal{B}_{1,+}$, we consider the following optimization setting:

$$\begin{aligned}\underset{\mathbf{h}_t,\mathbf{y}_t}{\text{minimize}} \quad & \beta c_1(\mathbf{h}_t) + (1-\beta)c_2(\mathbf{h}_t, \mathbf{y}_t) \\ \text{subject to} \quad & \|\mathbf{y}_t\|_1 \leq 1, \quad \mathbf{y}_t \geq 0\end{aligned} \tag{A.15}$$

for which the Lagrangian based reformulation can be written as

$$\underset{\lambda_1 \geq 0}{\text{maximize}} \ \underset{\mathbf{h}_t,\mathbf{y}_t}{\text{minimize}} \quad \beta c_1(\mathbf{h}_t) + (1-\beta)c_2(\mathbf{h}_t, \mathbf{y}_t) + \lambda_1(\|\mathbf{y}_t\|_1 - 1)$$

The updates for $\mathbf{h}_t$, gain variables $D_{1,ii}, D_{2,ii}$ and the synaptic weights follow the equations provided in Section 5.2.

For the output component $\mathbf{y}_t$, the corresponding cost function is an ℓ_1 regularized quadratic cost function. Following the primal-dual approach in [61], we can obtain the dynamic equations for output update as

$$\begin{aligned}\frac{d\mathbf{u}(\tau)}{d\tau} &= -\mathbf{u}(\tau) + \lambda_{SM}(1-\beta)[\mathbf{W}_{YH}(t)\mathbf{h}(\tau) - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}(\tau)], \\ \mathbf{y}_{t,i}(\tau) &= \text{ReLU}\left(\frac{\mathbf{u}_i(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Y\,ii}(t)D_{2,ii}(t)} - \lambda_1(\tau)\right), \qquad \text{for } i = 1, \ldots n,\end{aligned}$$

where $\text{ReLU}(x, \lambda_1)$ is the rectified-linear unit mapping defined by $\text{ReLU}(x) = \begin{cases} x & x > 0, \\ 0 & \text{otherwise} \end{cases}$.
Based on the dual maximization, the Lagrangian variable $\lambda_1(\tau)$ is updated by

$$\frac{da(\tau)}{d\tau} = -a(\tau) + \sum_{k=0}^{n} \mathbf{y}_{t,k}(\tau) - 1 + \lambda_1(\tau), \qquad \lambda_1(\tau) = \text{ReLU}(a(\tau)). \tag{A.16}$$

According to the expressions obtained above, in addition to the hidden layer and the output layer neurons, there is an additional neuron corresponding to the Lagrangian variable λ_1 of whose dynamics is governed by (A.16). The corresponding neuron generates an inhibition signal for the output neurons, based on the total output activation. The corresponding network structure is shown in Figure 9.

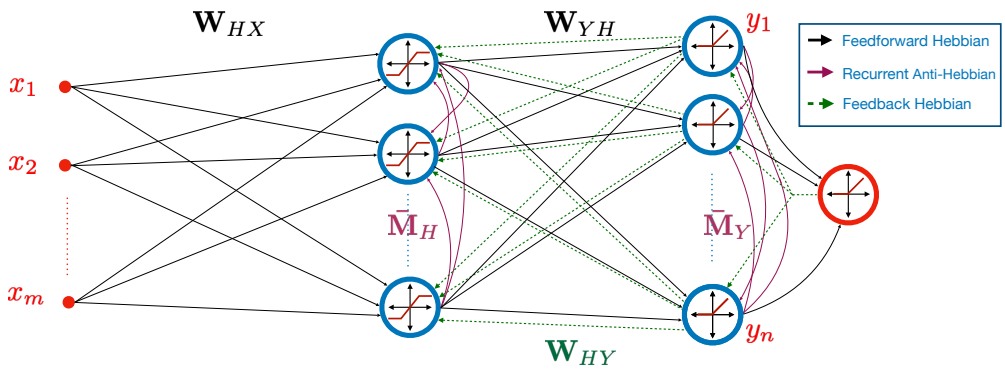


Figure 9: WSM Det-Max neural network for nonnegative sparse sources ($\mathcal{P} = \mathcal{B}_{1,+}$).

D.4 Sparse sources

In the sparse source setting where $\mathcal{P} = \mathcal{B}_1$, the only change relative to the nonnegative sparse case is the replacement of the ReLU output activation function with the soft thresholding function

$$ST_\lambda(x) = \begin{cases} 0 & |x| \leq \lambda \\ x - sign(x)\lambda & \text{otherwise.} \end{cases}$$

Therefore, we can rewrite the output dynamics for $\mathcal{P} = \mathcal{B}_1$ as

$$\begin{aligned} \frac{d\mathbf{u}(\tau)}{d\tau} &= -\mathbf{u}(\tau) + \lambda_{SM}(1-\beta)[\mathbf{W}_{YH}(t)\mathbf{h}(\tau) - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}(\tau)], \\ \mathbf{y}_{t,i}(\tau) &= \text{ST}_{\lambda_1(\tau)}\left(\frac{\mathbf{u}_i(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Yii}(t)D_{2,ii}(t)}\right), \qquad \text{for } i = 1, \ldots n, \\ \frac{da(\tau)}{d\tau} &= -a(\tau) + \sum_{k=0}^{n} |\mathbf{y}_{t,k}(\tau)| - 1 + \lambda_1(\tau), \qquad \lambda_1(\tau) = \text{ReLU}(a(\tau)). \end{aligned}$$

Figure 10 illustrates the WSM based Det-Max neural network for sparse BSS.

D.5 Unit simplex sources

The unit simplex set Δ is a face of the polytope $\mathcal{P} = \mathcal{B}_{1,+}$ which is the domain for nonnegative sparse sources. Therefore, we replace the ℓ_1-norm inequality constraint in (A.15) with the equality constraint to obtain the Det-Max WSM optimization problem for the unit simplex domain:

$$\begin{aligned} \underset{\mathbf{h}_t, \mathbf{y}_t}{\text{minimize}} \quad & \beta c_1(\mathbf{h}_t) + (1-\beta)c_2(\mathbf{h}_t, \mathbf{y}_t) \\ \text{subject to} \quad & \|\mathbf{y}_t\|_1 = 1, \quad \mathbf{y}_t \geq 0 \end{aligned}$$

Therefore, for the Lagrangian based formulation

$$\underset{\lambda_1}{\text{maximize}} \ \underset{\mathbf{h}_t, \mathbf{y}_t}{\text{minimize}} \qquad \beta c_1(\mathbf{h}_t) + (1-\beta)c_2(\mathbf{h}_t, \mathbf{y}_t) + \lambda_1(\|\mathbf{y}_t\|_1 - 1),$$

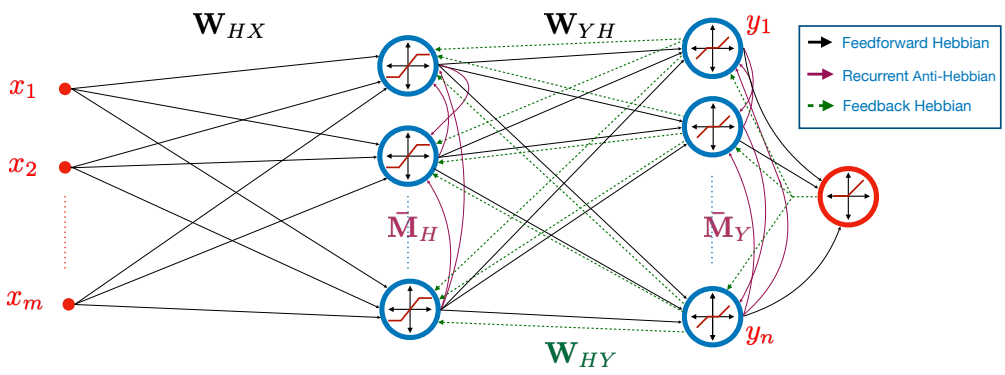


Figure 10: WSM Det-Max neural network for sparse sources ($\mathcal{P} = \mathcal{B}_1$).

we no longer require λ to be nonnegative. Therefore, for $\mathcal{P} = \Delta$ only required change relative to $\mathcal{B}_{1,+}$ is the replacement of the ReLU activation function of the rightmost inhibition neuron in Figure 9 with the linear activation. As a result, the output dynamics for the unit simplex sources can be written as

$$\begin{aligned}
\frac{d\mathbf{u}(\tau)}{d\tau} &= -\mathbf{u}(\tau) + \lambda_{SM}(1-\beta)[\mathbf{W}_{YH}(t)\mathbf{h}(\tau) - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}(\tau)], \\
\mathbf{y}_{t,i}(\tau) &= \text{ReLU}\left(\frac{\mathbf{u}_i(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Yii}(t)D_{2,ii}(t)} - \lambda_1(\tau)\right), \qquad \text{for } i = 1, \ldots n, \\
\frac{d\lambda_1(\tau)}{d\tau} &= -\lambda_1(\tau) + \sum_{k=0}^{n} \mathbf{y}_k(\tau) - 1 + \lambda_1(\tau).
\end{aligned}$$

Figure 11 shows the WSM-based Det-Max neural network for the unit-simplex sources.

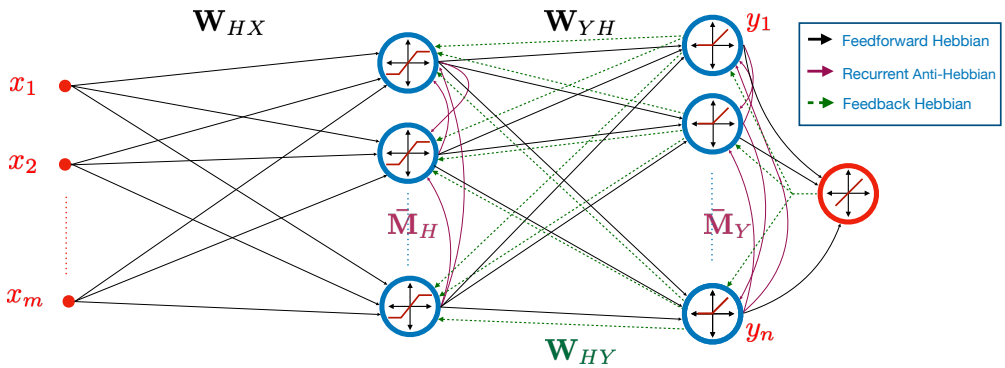


Figure 11: WSM Det-Max neural network for unit-simplex sources ($\mathcal{P} = \Delta$).

D.6 Sources with mixed attributes

We consider the following polytope example provided in [26]

$$\mathcal{P}_{ex} = \left\{ \mathbf{s} \in \mathbb{R}^3 \;\middle|\; \begin{array}{c} s_1, s_2 \in [-1, 1], s_3 \in [0, 1], \\ \left\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \right\|_1 \leq 1, \left\| \begin{bmatrix} s_2 \\ s_3 \end{bmatrix} \right\|_1 \leq 1 \end{array} \right\}, \tag{A.17}$$

which is an example of domains where source attributes such as nonnegativity and sparsity defined only at the subvector level.

The Det-Max WSM optimization setting for this case can be written as

$$\begin{aligned} \underset{\mathbf{h}_t,\mathbf{y}_t}{\text{minimize}} \quad & C(\mathbf{h}_t,\mathbf{y}_t) \\ \text{subject to} \quad & \left\| \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right\|_1 \leq 1, \quad \left\| \begin{bmatrix} y_2 \\ y_3 \end{bmatrix} \right\|_1 \leq 1 \quad y_3 \geq 0 \end{aligned}$$

for which the Lagrangian based reformulation can be written as

$$\underset{\lambda_1,\lambda_2\geq 0}{\text{maximize}} \ \underset{\mathbf{h}_t,y_3\geq 0,y_1,y_2}{\text{minimize}} \quad C(\mathbf{h}_t,\mathbf{y}_t) + \lambda_1 \left(\left\| \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right\|_1 - 1 \right) + \lambda_2 \left(\left\| \begin{bmatrix} y_2 \\ y_3 \end{bmatrix} \right\|_1 - 1 \right).$$

The proximal operator corresponding to the Lagrangian terms can be defined as

$$\text{prox}_{\lambda_1,\lambda_2}(\mathbf{v}) = \underset{q_3\geq 0,q_1,q_2}{\text{argmin}} \left(\frac{1}{2}\|\mathbf{v}-\mathbf{q}\|_2^2 + \lambda_1 \left\| \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \right\|_1 + \lambda_2 \left\| \begin{bmatrix} q_2 \\ q_3 \end{bmatrix} \right\|_1 \right). \tag{A.18}$$

Let $\mathbf{q}^*$ the output of the proximal operator. From the subdifferential set based optimality condition

- if $q_1^* \neq 0$ then $q_1^* - v_1 + \lambda_1 \text{sign}(v_1) = 0$ which implies $q_1^* = v_1 - \lambda_1 \text{sign}(v_1)$,
- if $q_2^* \neq 0$ then $q_2^* = v_2 - (\lambda_1 + \lambda_2)\text{sign}(v_2)$,
- if $q_3^* \neq 0$ then $q_3^* = v_3 - \lambda_2$.

Therefore, we can write $q_1 = \text{ST}_{\lambda_1}(v_1)$, $q_2 = \text{ST}_{\lambda_1+\lambda_2}(v_2)$ and $q_3 = \text{ReLU}(v_3 - \lambda_2)$. As a result, we can write the corresponding output dynamics expressions in the form

$$\begin{aligned} \frac{d\mathbf{u}(\tau)}{d\tau} &= -\mathbf{u}(\tau) + \lambda_{SM}(1-\beta)[\mathbf{W}_{YH}(t)\mathbf{h}(\tau) - \bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)\mathbf{y}(\tau)], \\ \mathbf{y}_{t,1}(\tau) &= \text{ST}_{\lambda_1(\tau)}\left(\frac{\mathbf{u}_1(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Y11}(t)D_{2,11}(t)} \right), \\ \mathbf{y}_{t,2}(\tau) &= \text{ST}_{\lambda_1(\tau)+\lambda_2(\tau)}\left(\frac{\mathbf{u}_2(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Y22}(t)D_{2,22}(t)} \right), \\ \mathbf{y}_{t,3}(\tau) &= \text{ReLU}\left(\frac{\mathbf{u}_3(\tau)}{\lambda_{SM}(1-\beta)\Gamma_{Y33}(t)D_{2,33}(t)} - \lambda_2(\tau) \right), \\ \frac{da_1(\tau)}{d\tau} &= -a_1(\tau) + |\mathbf{y}_{t,1}(\tau)| + |\mathbf{y}_{t,2}(\tau)| - 1 + \lambda_1(\tau), \\ \lambda_1(\tau) &= \text{ReLU}(a_1(\tau)), \\ \frac{da_2(\tau)}{d\tau} &= -a_2(\tau) + |\mathbf{y}_{t,2}(\tau)| + \mathbf{y}_{t,3}(\tau) - 1 + \lambda_2(\tau), \\ \lambda_2(\tau) &= \text{ReLU}(a_2(\tau)). \end{aligned}$$

Figure 12 shows the Det-Max WSM neural network for the source domain in (A.17).

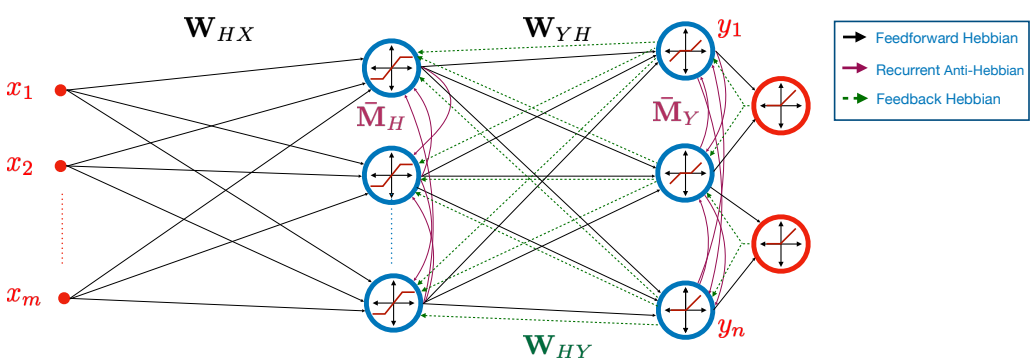


Figure 12: WSM Det-Max neural network for the polytope in (A.17).

E Supplementary on numerical experiments

Update dynamics for the hidden layer $\mathbf{h}_t$ and the output vector $\mathbf{y}_t$ are defined by differential equations depending on the selection of the source domain which lead to recursive neural dynamic iterations. Algorithm 1 summarizes the neural dynamic iterations for anti-sparse sources covered in Section 5.2. Very similar output dynamic calculations for each source assumption can be acquired based on the derivations in Section D. We run the neural dynamic iterations until a convergence check is satisfied or a predetermined maximum number of iterations τ_{max} is reached. In Algorithm 1, ϵ denotes the tolerance in the relative error check for the stopping condition, and $\eta(\tau)$ represents the learning rate at the iteration count τ. In the following subsections, we provide the experimental details and additional source separation examples for different assumptions on the sources.

Algorithm 1 Neural dynamic iterations for anti-sparse sources

Initialize τ_{max}, ϵ, and $\tau = 1$
while $(||\mathbf{v}(\tau) - \mathbf{v}(\tau-1)||/||\mathbf{v}(\tau)|| > \epsilon$ or $||\mathbf{u}(\tau) - \mathbf{u}(\tau-1)||/||\mathbf{u}(\tau)|| > \epsilon)$ and $\tau < \tau_{\text{max}}$ **do**
  $\mathbf{v}(\tau) = \mathbf{v}(\tau-1) + \eta(\tau)\frac{d\mathbf{v}(\tau-1)}{d(\tau-1)}$
  Apply Equation 10 for $\mathbf{h}_{t,i}(\tau)$
  $\mathbf{u}(\tau) = \mathbf{u}(\tau-1) + \eta(\tau)\frac{d\mathbf{u}(\tau-1)}{d(\tau-1)}$
  Apply Equation 12 for $\mathbf{y}_{t,i}(\tau)$
  $\tau = \tau + 1$
end while

E.1 Batch algorithms with correlated source separation capability

In this section, we briefly discuss two batch learning algorithms for blind separation of correlated sources, which reflect the Det-Max problem 3: 1. Polytopic Matrix Factorization [26], 2. Log-Det Mutual Information Maximization [59].

- ***Polytopic Matrix Factorization***: [26] recently introduced the Polytopic Matrix Factorization (PMF) as a structured matrix factorization framework that models the columns of the input matrix, i.e., the mixture signals in our problem, as a linear transformation of source vectors from a polytope. The choice of the underlying polytope in the PMF framework reflects the attributes of the sources possibly in a heterogeneous perspective; e.g., the polytope discussed in Section D.6 provides an example of heterogeneous feature assumptions at the subvector level such as mutual sparsity. Taking into account the mixing model in Section 2.2, PMF uses the following optimization problem,

$$\underset{\mathbf{Y}(t) \in \mathbb{R}^{n\times t}, \mathbf{H} \in \mathbb{R}^{m\times n}}{\text{maximize}} \quad \log(\det(\mathbf{Y}(t)\mathbf{Y}(t)^T)) \tag{A.19a}$$

$$\text{subject to} \quad \mathbf{X}(t) = \mathbf{H}\mathbf{Y}(t), \tag{A.19b}$$

$$\mathbf{y}_i \in \mathcal{P}, i = 1, \ldots, t, \tag{A.19c}$$

  where $\mathbf{H}$ and $\mathbf{Y}(t)$ correspond to the unknown mixing matrix and the source estimates, respectively. The aim of PMF is to obtain the original factors of $\mathbf{A}$ and $\mathbf{S}(t)$ up to some acceptable sign and permutation ambiguities, i.e., $\mathbf{Y}(t) = \mathbf{P}\mathbf{\Lambda}\mathbf{S}(t)$ and $\mathbf{H} = \mathbf{A}\mathbf{P}^T\mathbf{\Lambda}^{-1}$. The reference [26] provides the sufficient condition for the identifiability of the original factors of $\mathbf{A}$ and $\mathbf{S}(t)$ based on the sufficiently scattering condition discussed in Section A, i.e., if the source vectors are sufficiently scattered in a permutation-and/or sign only invariant polytope $\mathcal{P}$, then all global optima of the problem A.19 lead to the ideal separation. For the corresponding algorithm to solve the problem A.19, we refer to the pseudo-code in [26], which is a batch algorithm with a projected gradient search.
- ***Log-Det Mutual Information Maximization***: The reference [59] brings a statistical interpretation to the PMF framework based on a log-determinant (LD) based mutual information measure. According to this approach, the LD-mutual information between the input and output is maximized, under the constraint that the outputs are in the presumed source domain. The corresponding optimization setting is given by

$$\underset{\mathbf{Y}(t) \in \mathbb{R}^{n\times t}}{\text{maximize}} \quad \frac{1}{2}\log\det(\hat{\boldsymbol{R}}_{\boldsymbol{y}} + \epsilon \boldsymbol{I}) - \frac{1}{2}\log\det(\hat{\boldsymbol{R}}_{\boldsymbol{y}} - \hat{\boldsymbol{R}}_{\boldsymbol{yx}}(\epsilon \boldsymbol{I} + \hat{\boldsymbol{R}}_{\boldsymbol{x}})^{-1}\hat{\boldsymbol{R}}_{\boldsymbol{yx}}^T + \epsilon \boldsymbol{I}) \quad \text{(A.20a)}$$

$$\text{subject to} \quad \mathbf{y}_i \in \mathcal{P}, i = 1, \ldots, t, \quad \text{(A.20b)}$$

where the objective A.20a is defined in terms of sample covariance matrices, i.e., $\hat{\boldsymbol{R}}_{\boldsymbol{y}} = \frac{1}{t}\mathbf{Y}(t)\mathbf{Y}(t)^T - \frac{1}{t^2}\mathbf{Y}(t)\mathbf{1}\mathbf{1}^T\mathbf{Y}(t)^T$, and $\hat{\boldsymbol{R}}_{\boldsymbol{yx}} = \frac{1}{t}\mathbf{Y}(t)\mathbf{X}(t)^T - \frac{1}{t^2}\mathbf{Y}(t)\mathbf{1}\mathbf{1}^T\mathbf{X}(t)^T$. Similar to the PMF framework, the LD-InfoMax approach assumes that the source vectors are drawn from a presumed polytope $\mathcal{P}$. The LD-InfoMax approach is capable of separating correlated sources, since it does not assume any statistical independence or uncorrelatedness on the source vectors. Reference [59] proposes a projected gradient ascent-based algorithm to solve the problem A.20 as a batch learning approach.

We compare our algorithm with the PMF and LD-InfoMax frameworks for correlated source separation experiments in Sections 6.1, E.3, E.4, and for sparse source separation experiment in Section E.5.

E.2 Synthetically correlated source separation with nonnegative anti-sparse sources

In this section, we provide the training details and hyperparameter selections for the numerical experiment provided in Section 6.1. For this network, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = \mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.
- $\mu_{\mathbf{D}_1} = 1$, and $\mu_{\mathbf{D}_2} = 10^{-2}$.
- $\beta = 0.5,\ \lambda_{SM} = 1 - 10^{-5}$.
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{\nu/(1 + \log(1 + t)), 0.001\}$, where t is the data sample index, and $\nu = \begin{cases} 0.1 & \rho \leq 0.4, \\ 0.05 & \text{otherwise.} \end{cases}$.
- $\mathbf{M}_H = 2\mathbf{I}, \mathbf{M}_Y = \mathbf{I}$.
- $\mathbf{W}_{HX} = \mathbf{I}, \mathbf{W}_{YH} = \mathbf{I}$.
- Learning rate for the neural dynamic iterations is adjusted using $max\{0.75/(1+\tau \times 0.005), 0.05\}$, where τ is the neural dynamic iteration count.
- The maximum number of neural dynamic iterations is restricted to $\tau_{\text{max}} = 500$ if the stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $0.2 \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $0.2 \prec \text{diag}(\mathbf{D}_2) \prec 5$.

E.3 Synthetically correlated source separation with anti-sparse sources

To illustrate the correlated source separation of WSM neural networks with antisparse sources, we consider a numerical experiment with four copula-T distributed sources in the range $[-1, 1]$ with a Toeplitz correlation calibration matrix whose first row is $\begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \end{bmatrix}$. We consider the range $\rho \in [0, 0.6]$ for the correlation level. The sources are mixed with an 8×4 random matrix with i.i.d. standard normal entries, and corrupted by i.i.d. standard normal noise corresponding to 30dB SNR level. Antisparse-WSM neural network is employed in this experiment, which is illustrated in Figure 6. We compare the SINR performance of WSM algorithm with the BSM algorithm [18], Infomax ICA algorithm [58], PMF algorithm [26], and LD-InfoMax algorithm [59]. Figure 13 illustrates the SINR performances of these algorithms (averaged over 100 realizations) with respect to the correlation parameter ρ. Similar to the results for nonnegative antisparse source separation experiments provided in Section 6.1, the WSM approach maintains its immunity against source correlations, whereas the BSM and ICA algorithms, which assume uncorrelated sources, deteriorate with increasing source correlation. LD-InfoMax and PMF algorithms achieve relatively similar SINR behaviors while their performance remains comparatively steady with respect to increasing source correlation. We note that both PMF and LD-InfoMax typically achieve better performances compared to our proposed online algorithm since these approaches utilize batch algorithms.

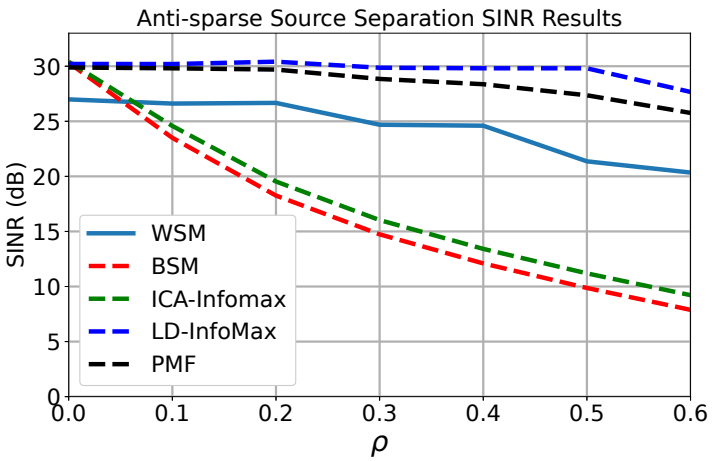


Figure 13: The SINR performances of the WSM, BSM, ICA, LD-InfoMax, and PMF algorithms as a function of the correlation parameter ρ.

For the antisparse source separation setting, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = \mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$,
- $\mu_{\mathbf{D}_1} = 1.125$, and $\mu_{\mathbf{D}_2} = 0.2$,
- $\beta = 0.5,\ \lambda_{SM} = 1 - 5 \times 10^{-5}$,
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{\nu/(1 + \log(1 + t)), 0.001\}$, where t is the data sample index, and $\nu = \begin{cases} 0.6 & \rho \leq 0.4, \\ 0.25 & \text{otherwise.} \end{cases}$.
- $\mathbf{M}_H = 2\mathbf{I}, \mathbf{M}_Y = \mathbf{I}$,
- $\mathbf{W}_{HX} = \mathbf{I}, \mathbf{W}_{YH} = \mathbf{I}$.
- Learning rate for the neural dynamic iterations is adjusted using $max\{0.75/(1+\tau \times 0.005), 0.05\}$, where τ is the neural dynamic iteration count.
- The maximum number of neural dynamic iterations is restricted to $\tau_{\text{max}} = 750$ if the stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $0.2 \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $0.2 \prec \text{diag}(\mathbf{D}_2) \prec 5$.

E.4 Image separation

For the image separation example provided in Section 6.2, the WSM Det-Max Neural Network illustrated in Figure 8 is employed. For this network, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = \mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$.
- $\mu_{\mathbf{D}_1} = 3.725$, and $\mu_{\mathbf{D}_2} = 1.125$.
- $\beta = 0.5,\ \lambda_{SM} = 1 - 10^{-5}$.
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{0.11/(1 + \log(1 + t)), 0.001\}$, where t is the data sample index.
- $\mathbf{M}_H = 2\mathbf{I}, \mathbf{M}_Y = \mathbf{I}$.
- $\mathbf{W}_{HX} = \mathbf{I}, \mathbf{W}_{YH} = \mathbf{I}$.
- Learning rate for the neural dynamic iterations is adjusted using $max\{0.75/(1+\tau \times 0.005), 0.05\}$, where τ is the neural dynamic iteration count.
- Maximum number of neural dynamic iterations is restricted to be $\tau_{\text{max}} = 500$ if stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $10^{-3} \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $10^{-3} \prec \text{diag}(\mathbf{D}_2) \prec 20$.

In this section, we also include the results of batch algorithms PMF and LD-InfoMax as illustrated in Figure 14 in addition to the source images, mixture images, and the outputs of the ICA, NSM, and WSM algorithms with better resolutions compared to Figure 4. Recall that our WSM-based network outputs illustrated in Figure 14e achieves SINR level of 27.49 dB. LD-InfoMax algorihtm's outputs in Figure 14f obtain SINR level of 28.65 dB, and the PMF algorithm's outputs in Figure 14g obtain the SINR level of 31.92 dB. As expected, both PMF and LD-InfoMax algorithms achieve better performances due to their batch nature whereas our proposed approach's output is compatible with these frameworks.

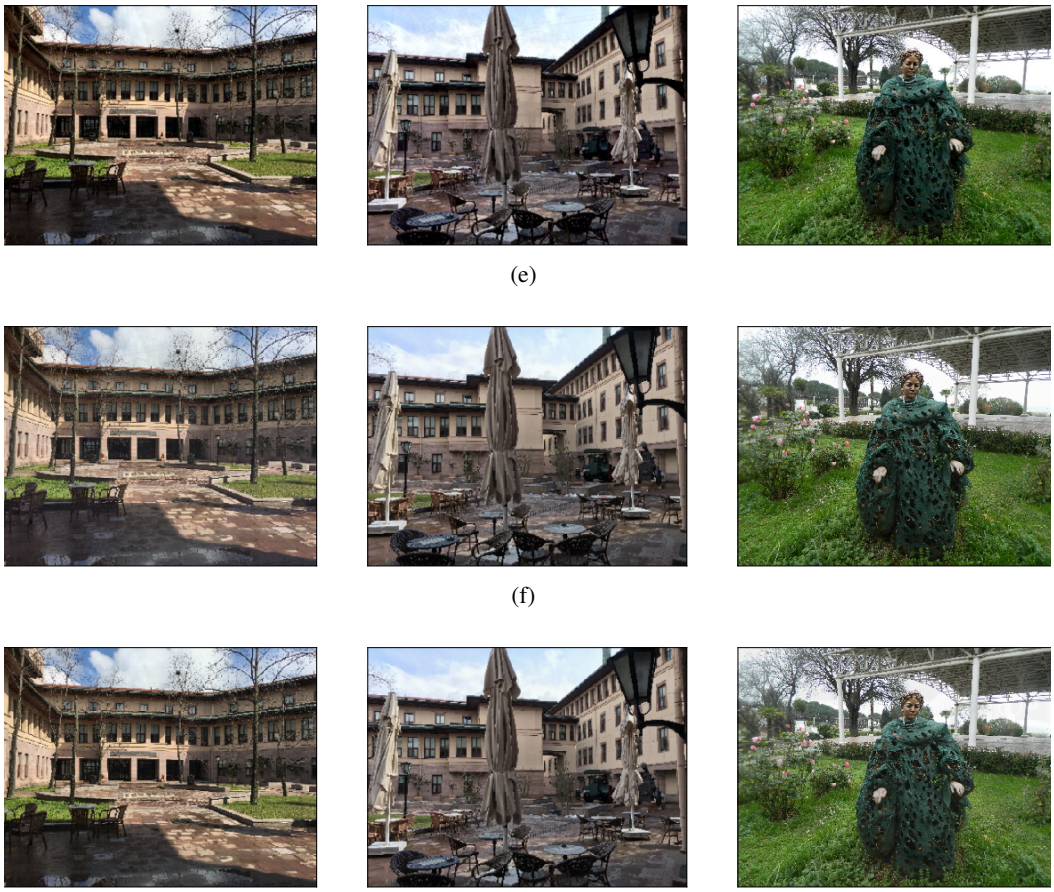
(a)

(b)

(c)

(d)

Figure 14: (a) Original RGB images, (b) mixture RGB images, (c) ICA outputs, (d) NSM outputs.



(e)

(f)

(g)

Figure 14: (e) WSM outputs, (f) LD-InfoMax Outputs, (g) PMF Outputs.

E.5 Sparse source separation

In order to illustrate the use of the proposed framework for a different source domain, we consider sparse sources where $\mathcal{P} = \mathcal{B}_1$. We generate $n = 5$ dimensional source vectors, by projecting i.i.d. uniform vectors in $\mathcal{B}_\infty$ to $\mathcal{B}_1$. The mixing matrix is a 10×5-matrix with i.i.d. standard normal entries. The mixtures are used to train the sparse-WSM Det-Max network in Figure 10 introduced in Appendix D.4. For this network, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = 8\mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$.
- $\mu_{\mathbf{D}_1} = 20$, and $\mu_{\mathbf{D}_2} = 10^{-2}$.
- $\beta = 0.5,\ \lambda_{SM} = 1 - 10^{-5}$.
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{0.25/(1 + \log(1 + t)), 0.001\}$, where t is the data sample index.
- $\mathbf{M}_H = 0.02\mathbf{I}, \mathbf{M}_Y = 0.02\mathbf{I}$.
- $\mathbf{W}$ matrices are initialized first with i.i.d. standard normal random variables. Then, we normalized the Euclidean norm of all rows to 0.0033 by proper scaling.
- Learning rate for the neural dynamic iterations is determined to be 0.5.
- Maximum number of neural dynamic iterations is restricted to be $\tau_{\text{max}} = 750$ if stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $10^{-6} \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $1 \prec \text{diag}(\mathbf{D}_2) \prec 1.001$.

Figure 15 illustrates the SINR convergence behavior for the sparse-WSM network, as a function of update iterations, for the input SNR level of 30dB (averaged over 200 realizations).

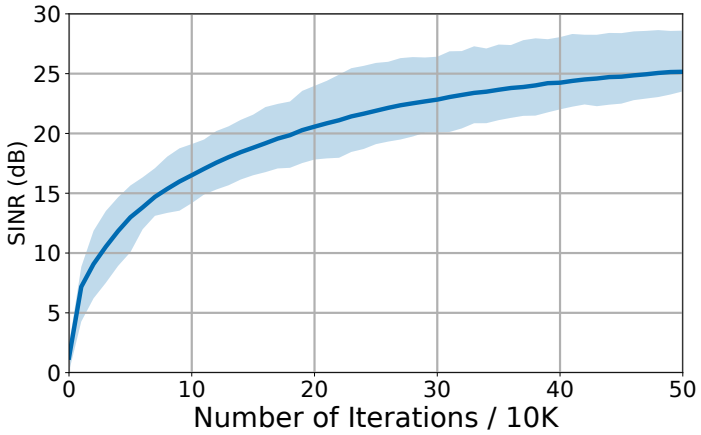


Figure 15: The SINR convergence curve for the sparse-WSM for 30dB input SNR level: mean-solid line with 25/75-percentile envelope.

Figure 16 demonstrates the separation performance of the sparse-WSM network for different noise levels.

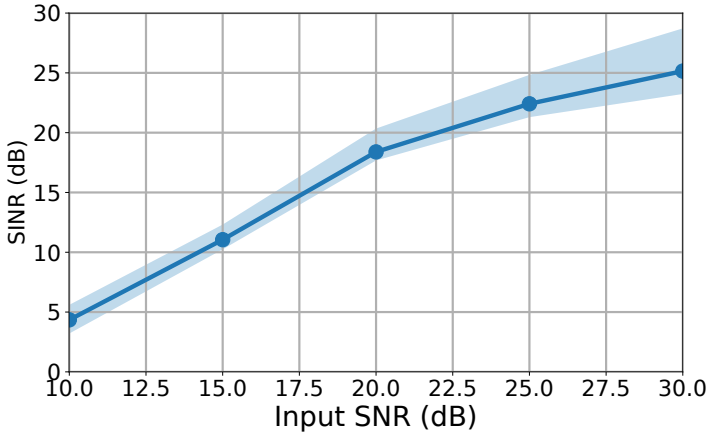


Figure 16: The output SINR with respect to the input SNR level for the sparse-WSM: mean-solid line with 25/75-percentile envelope.

To compare our online approach with the batch algorithms LD-InfoMax and PMF, we also performed experiments with these algorithms for the input SNR level of 30 dB. Table 2 summarizes the averaged SINR results of each algorithm over 200 realizations for 30 dB input SNR level. In these experiments, we observe that both PMF and LD-InfoMax obtain better SINR performances on average compared to our WSM Det-Max network. This condition is due to the batch nature of both PMF and LD-InfoMax as discussed earlier.

Table 2: Sparse source separation averaged SINR results of WSM, PMF, and LD-InfoMax.

Algorithm	**WSM**	**PMF**	**LD-InfoMax**
SINR	25.14	30.17	30.0

E.6 Sparse dictionary learning

Related to the previous example, we consider the well-known example of sparse coding, which is the dictionary learning for natural image patches [20]. For this experiment, we used 12×12 prewhitened image patches obtained from the website, http://www.rctn.org/bruno/sparsenet. We used the vectorized versions of these patches to train the sparse Det-Max WSM neural network in Figure 10. Figure 17 shows the receptive field images obtained from the columns of the inverse of the sparse-WSM separator, which correspond to localized Gabor-like edge features. This example confirms that the sparse WSM neural network with a local update rule successfully captures the behavior observed in primates' primary visual cortical neurons.

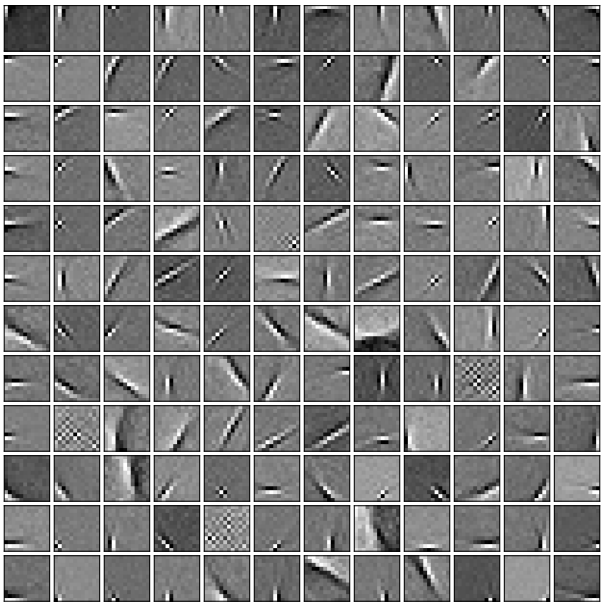


Figure 17: Dictionary obtained from the natural image patches by the sparse-WSM Network.

E.7 Source separation with mixed latent attributes

In this section, we illustrate the source separation setting with different identifiable-enabling polytopes similar to the given example in D.6. These experiments demonstrate the capability of the proposed WSM Neural Network for general identifiable polytopes. The identifiability of the provided sets in this section are verified by the graph automorphism-based identifiability characterization algorithm presented in [45].

E.7.1 Special polytope example in appendix D.6

We provide numerical experiment results for the WSM Det-Max network in Figure 12 corresponding to the polytope in (A.17). To employ this WSM Det-Max Neural Network, we synthetically generated $n = 3$ dimensional uniform vectors in this polytope and mixed them by a random 6×3-matrix with i.i.d. standard normal entries. Also, the mixtures are corrupted by i.i.d. standard normal noise corresponding to 30dB SNR level. Figure 18 illustrates the behavior of the overall SINR and individual source SNRs in addition to the behavior of diagonal weight matrices ($\mathbf{D}_1$ and $\mathbf{D}_2$) with respect to the number of update iterations for a single experiment. To measure the average behavior of this neural network, we run experiments for 100 different source and mixing matrix generation, and Figure 19 illustrates the averaged SINR convergence behavior with the $25/75$-percentile envelope, as a function of update iterations.

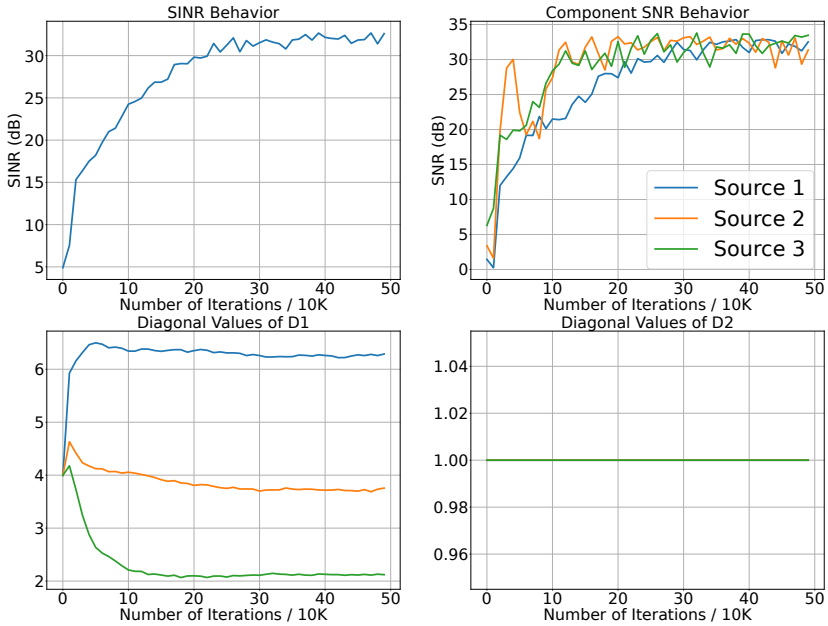


Figure 18: Example behaviors of SINR, component SNR values, and diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ for a single experiment discussed in E.7.1

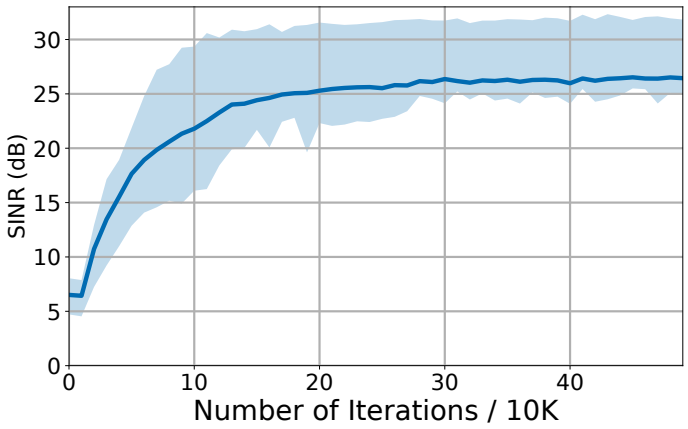


Figure 19: The SINR convergence curve for the experiments discussed in E.7.1: mean-solid line with 25/75-percentile envelope.

For this network, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = 4\mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$.
- $\mu_{\mathbf{D}_1} = 5.725$, and $\mu_{\mathbf{D}_2} = 10^{-2}$ ($\mu_{\mathbf{D}_2} = 0$ for the experiment visualized in Figure 18).
- $\beta = 0.5,\ \lambda_{SM} = 1 - 10^{-4}$.
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{0.25/(1 + \log(1 + t)), 0.001\}$, where t is the data sample index.
- $\mathbf{M}_H = 0.02\mathbf{I}, \mathbf{M}_Y = 0.02\mathbf{I}$.

- $\mathbf{W}$ matrices are initialized first with i.i.d. standard normal random variables. Then, we normalized the Euclidean norm of all rows to 0.0033 by proper scaling.
- Learning rate for the neural dynamic iterations is determined to be 0.5.
- Maximum number of neural dynamic iterations is restricted to be $\tau_{\text{max}} = 750$ if stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $10^{-6} \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $1 \prec \text{diag}(\mathbf{D}_2) \prec 1.001$.

E.7.2 Mixed anti-sparse and nonnegative anti-sparse sources

As another identifiable polytope example, we consider the following set which assigns mixed antisparse attributes to the source components: signed or nonnegative. For this experiment, we randomly selected two components to be nonnegative whereas the remaining three components are antisparse. The mixing matrix is a $10 \times 5-$matrix with i.i.d. standard normal entries. The mixtures are used to train the WSM Det-Max network similar to Figure 6 where the clippings at the output layer corresponding to nonnegative sources are replaced with nonnegative clipping.

$$\mathcal{P} = \left\{ \mathbf{s} \in \mathbb{R}^3 \;\middle|\; s_{j_1}, s_{j_2}, s_{j_3} \in [-1, 1], s_{j_4}, s_{j_5} \in [0, 1], j_i \in \{1, 2, 3, 4, 5\} \right\}, \tag{A.21}$$

To train the WSM Det-Max network in this scenario, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = \mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$.
- $\mu_{\mathbf{D}_1} = 1.125$, and $\mu_{\mathbf{D}_2} = 0.1$.
- $\beta = 0.5,\ \lambda_{SM} = 1 - 5 \times 10^{-5}$.
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{0.4/(1 + \log(1 + t)), 0.001\}$, where t is the data sample index.
- $\mathbf{M}_H = 2\mathbf{I}, \mathbf{M}_Y = \mathbf{I}$.
- $\mathbf{W}_{HX} = \mathbf{I}, \mathbf{W}_{YH} = \mathbf{I}$.
- Learning rate for the neural dynamic iterations is adjusted using $max\{0.75/(1+\tau \times 0.005), 0.05\}$, where τ is the neural dynamic iteration count.
- Maximum number of neural dynamic iterations is restricted to be $\tau_{\text{max}} = 750$ if stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $0.2 \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $0.5 \prec \text{diag}(\mathbf{D}_2) \prec 5$.

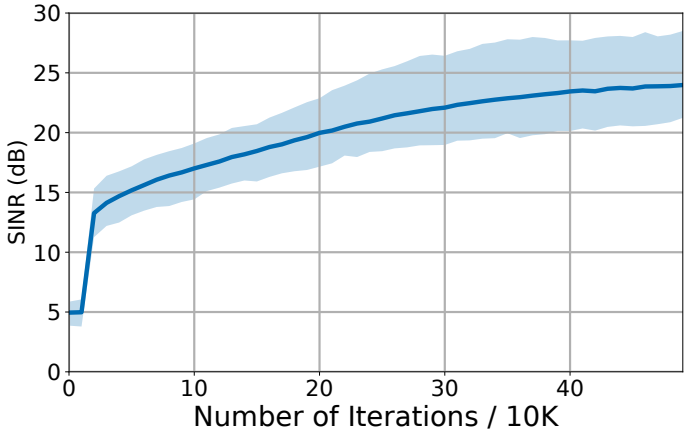


Figure 20: The SINR convergence curve for the experiments discussed in E.7.2: mean-solid line with 25/75-percentile envelope.

E.7.3 Mixed sparse and nonnegative anti-sparse sources

As the last illustration of source separation on identifiable domains, we consider the following polytope,

$$\mathcal{P} = \left\{ \mathbf{s} \in \mathbb{R}^3 \;\middle|\; s_{j_1} \in [0,1], \left\| \begin{bmatrix} s_{j_2} \\ s_{j_3} \\ s_{j_4} \\ s_{j_5} \end{bmatrix} \right\|_1 \leq 1, j_i \in \{1,2,3,4,5\} \right\}, \tag{A.22}$$

where only one component is nonnegative and the subvector containing the remaining components is sparse. To demonstrate the source separation ability of WSM Det-Max Neural Network for this underlying domain, we generated $n = 5$ dimensional uniform vectors in this polytope. The sources are mixed with a 10×5 random matrix with standard normal entries. To train the WSM Det-Max network in this setting, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = 8\mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$.
- $\mu_{\mathbf{D}_1} = 6$, and $\mu_{\mathbf{D}_2} = 0.1$.
- $\beta = 0.5,\ \lambda_{SM} = 1 - 10^{-4}$.
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{0.25/(1 + \log(1 + t)), 0.001\}$, where t is the data sample index.
- $\mathbf{M}_H = 0.02\mathbf{I}, \mathbf{M}_Y = 0.02\mathbf{I}$.
- $\mathbf{W}$ matrices are initialized first with i.i.d. standard normal random variables. Then, we normalized the Euclidean norm of all rows to 0.0033 by proper scaling.
- Learning rate for the neural dynamic iterations is adjusted using $max\{0.5/(1 + \tau \times 0.005), 0.01\}$, where τ is the neural dynamic iteration count.
- Maximum number of neural dynamic iterations is restricted to be $\tau_{\text{max}} = 750$ if stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $10^{-6} \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $1 \prec \text{diag}(\mathbf{D}_2) \prec 5$.

Figure 21 illustrates the SINR convergence behavior (averaged over 100 realizations) of the WSM Det-Max network for this scenario, as a function of update iterations.

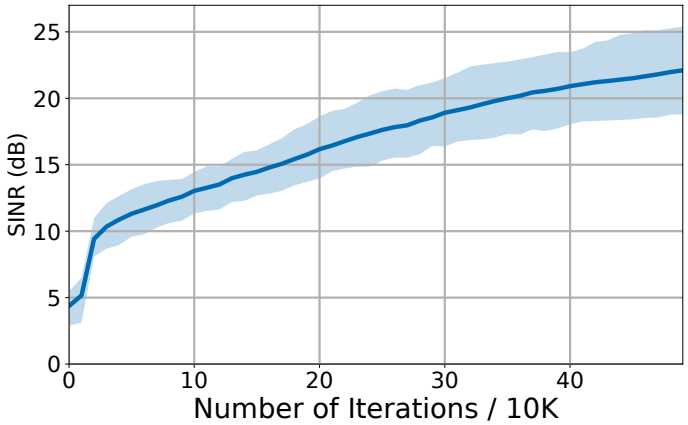


Figure 21: The SINR convergence curve for the experiments discussed in E.7.3: mean-solid line with 25/75-percentile envelope.

E.8 Digital communication example: 4-PAM modulation scheme

We consider the 4 Pulse-amplitude modulation (4-PAM) scheme as a realistic application of blind separation of digital communication signals, with the symbols $\{\pm 3, \pm 1\}$. We consider a uniform symbol distribution, i.e., $P(s = i) = \frac{1}{4}\ \forall i = \pm 3, \pm 1$, where s represents the transmitted symbol.

We assume that 5 sources are transmitted, with 400000 samples each, and mixed through a 10×5 random matrix with standard normal entries. Without loss of generality, we make use of $\mathcal{B}_{\ell_\infty}$ polytope as the source domain assumption so that we feed the mixtures to the WSM Det-Max neural network for the antisparse sources. To train this network, we used the following hyperparameter selections and variable initializations:

- $\mathbf{D}_1 = 0.5\mathbf{I}$, and $\mathbf{D}_2 = 0.5\mathbf{I}$.
- $\mu_{\mathbf{D}_1} = 0.01$, and $\mu_{\mathbf{D}_2} = 0.01$.
- $\beta = 0.5$, and $\lambda_{SM} = 1 - 5 \times 10^{-3}$.
- $1 - \gamma^2$ is dynamically adjusted using $1 - \gamma^2 = max\{0.3/(1 + \log(1 + t)), 0.05\}$, where t is the data sample index.
- $\mathbf{M}_H = 2\mathbf{I}, \mathbf{M}_Y = \mathbf{I}$.
- $\mathbf{W}$ matrices are initialized first with i.i.d. standard normal random variables. Then, we normalized the Euclidean norm of all rows to 0.005 by proper scaling.
- Learning rate for the neural dynamic iterations is adjusted using $max\{0.5/(1 + \tau \times 0.005), 0.01\}$, where τ is the neural dynamic iteration count.
- Maximum number of neural dynamic iterations is restricted to be $\tau_{\text{max}} = 750$ if stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $0.2 \prec \text{diag}(\mathbf{D}_1) \prec 10^6$ and $0.2 \prec \text{diag}(\mathbf{D}_2) \prec 25$.

Figure 21 illustrates the SINR convergence behavior (averaged over 20 realizations) of the WSM Det-Max network for this scenario, as a function of update iterations.We conclude that our proposed approach is able to separate the source symbols from their mixtures.

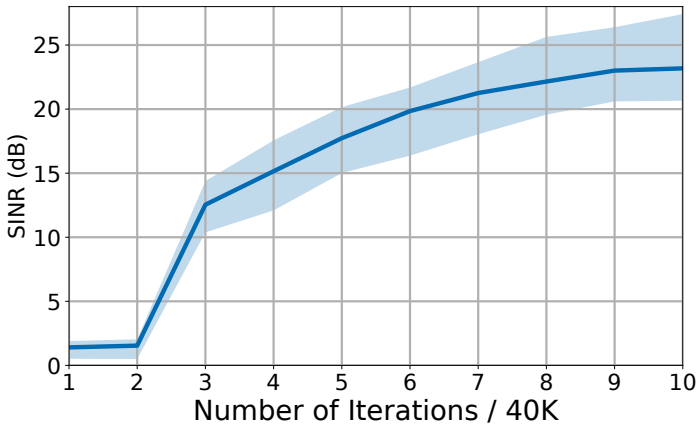


Figure 22: The SINR convergence curve for the 4-PAM digital communication signals: mean solid line with 25/75-percentile envelope.

E.9 Ablation study on hyperparameter selection for nonnegative sparse sources

The proposed Det-Max WSM framework requires many hyperparameter selections. In Section E, we discuss the selection of these hyperparameters for different source domains. Most of the time, we find these hyperparameters by trial error and sensitivity analysis. Several ablation studies similar to grid search are useful to find the optimal values for the hyperparameters. In this section, we provide such ablation studies on effects of the selection of λ_{SM}, $\mathbf{D}_1$, $\mu_{\mathbf{D}_1}$, and γ. We chose to focus on λ_{SM} here because we observed that it is one of the most sensitive parameters. Although the other parameters appear to have less of an effect on the final result than λ_{SM}, the cumulative impacts of the combined hyperparameter choices can substantially influence overall performance.

We consider nonnegative sparse source separation setup, i.e., $\mathcal{P} = \mathcal{B}_{\infty,+}$. We generate $n = 5$ dimensional source vectors uniformly in $\mathcal{B}_{\infty,+}$, and the mixing matrix is a 10×5-matrix with i.i.d. standard normal entries. The mixtures train the nonnegative sparse-WSM Det-Max network

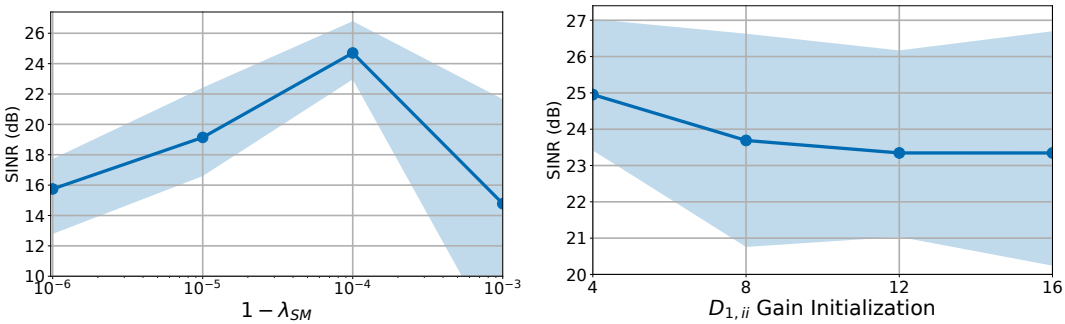
illustrated in Figure 9. In these ablation studies, we specifically consider the effect of hyperparameter selection for $1-\lambda_{\text{SM}}$, initial $\mathbf{D}_1$, $\mu_{\mathbf{D}_1}$, and initial $1-\gamma^2$. For each of the mentioned hyperparameters, we consider the following choices,

- $1-\lambda_{\text{SM}} \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$,
- $\mathbf{D}_1 \in \{4\mathbf{I}, 8\mathbf{I}, 12\mathbf{I}, 16\mathbf{I}\}$,
- $\mu_{\mathbf{D}_1} \in \{5, 10, 15, 20\}$,
- initial $1-\gamma^2 \in \{0.15, 0.20, 0.25, 0.30\}$

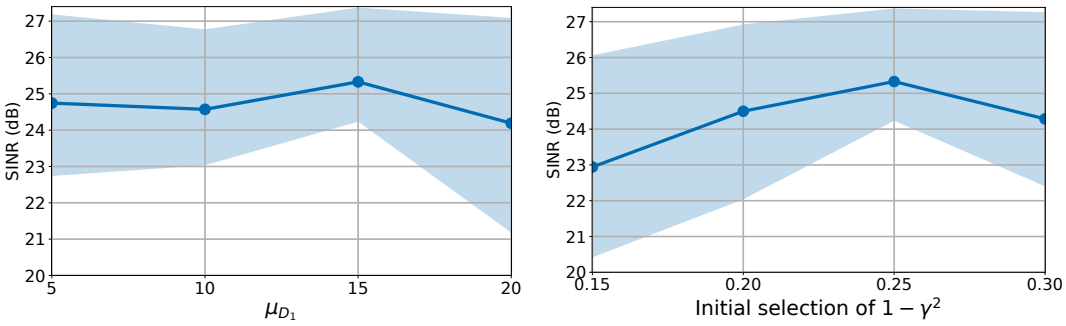
While experimenting with one hyperparameter, we fixed the rest of them as given in the following list,

- $\mathbf{D}_1 = 4\mathbf{I}$, and $\mathbf{D}_2 = \mathbf{I}$.
- $\mu_{\mathbf{D}_1} = 15$, and $\mu_{\mathbf{D}_2} = 0.01$.
- $\beta = 0.5$, and $\lambda_{SM} = 1 - 10^{-4}$.
- $1-\gamma^2$ is dynamically adjusted using $1-\gamma^2 = max\{0.25/(1+\log(1+t)), 10^{-3}\}$, where t is the data sample index.
- $\mathbf{M}_H = 0.02\mathbf{I}, \mathbf{M}_Y = 0.02\mathbf{I}$.
- $\mathbf{W}$ matrices are first initialized with i.i.d. standard normal random variables. Then, we normalize the Euclidean norm of all rows to 0.0033 by proper scaling.
- The learning rate for the neural dynamic iterations is adjusted using $max\{0.5/(1+\tau\times 0.005), 0.2\}$, where τ is the neural dynamic iteration count.
- Maximum number of neural dynamic iterations is restricted to be $\tau_{\text{max}} = 750$ if stopping condition is not satisfied.
- For the stability of the learning process, we keep the diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ in a predetermined range, i.e., $10^{-6} \prec \text{diag}(\mathbf{D}_1) \prec 10^{6}$ and $1 \prec \text{diag}(\mathbf{D}_2) \prec 1.001$.

Figure 23a illustrates the SINR performance of the WSM Det-Max network concerning $1-\lambda_{\text{SM}}$, and it demonstrates that it significantly affects the final SINR behavior of the proposed approach. We argue that the selection $\lambda_{\text{SM}} = 1 - 10^{-4}$ is a near-optimal for nonnegative sparse source separation with the WSM Det-Max network, whereas one can also implement a more detailed search based on possibly other hyperparameter dependencies. We also analyze the effect of initial $\mathbf{D}_1$ on the final SINR, and Figure 23b demonstrates the performance change with $\mathbf{D}_1$ gain initialization. We inspect that the WSM Det-Max network for nonnegative sparse sources relatively maintains its averaged performance against different initial gain parameters, whereas the selection of $\mathbf{D}_1 = 4\mathbf{I}$ leads to best performance with a significantly lower variance compared to other initialization choices. In Figure 22c, we visualize the effect of learning rate choice for $\mathbf{D}_1$. It is noticeable that $\mu_{\mathbf{D}_1}$ is less effective in SINR performance compared to other considered hyperparameters, but $\mu_{\mathbf{D}_1} = 15$ achieves the best average result with a lower variance. As the final ablation study on hyperparameter selection, we consider the initial value of $1-\gamma^2$ which we dynamically adjust using $max\{\nu/(1+\log(1+t)), 10^{-3}\}$, where t is the data sample index and ν is the initial value. Figure 22d illustrates the effect for the initial value ν, and it is remarkable that an improved result is attained for $\nu = 0.25$.



(a) (b)



(c) (d)

Figure 22: SINR performances of WSM Det-Max networks for different hyperparameter selections (averaged over 50 realizations, mean solid lines with 25/75-percentile envelopes): (a) averaged SINR performance with respect to $1-\lambda_{\text{SM}}$, (b) averaged SINR performance with respect to initial $\mathbf{D}_1$, (c) averaged SINR performance with respect to $\mu_{\mathbf{D}_1}$, (d) averaged SINR performance with respect to initial $1-\gamma^2$.

F Discussion on the complexity of the proposed approach

In this section, we discuss the computational complexity of the proposed WSM Det-Max neural network implementations. For simplicity, we consider the antisparse source separation cases discussed in Section 5.2. Remarkably, the overall complexity is due to the output computation complexities which are determined by (9)-(10) and (11)-(12). Note that these differential equations are naturally solved in neuromorphic implementations. However, in digital computer simulations, we need to implement loops to obtain their iterative solutions, as summarized in Algorithm 1. As described in Section 2.2, assume that there are n sources and m mixtures, i.e., $\mathbf{x}_t \in \mathbb{R}^m$, and $\mathbf{h}_t, \mathbf{y}_t \in \mathbb{R}^n$ for all t. Assuming that the factors $(1-\beta)\bar{\mathbf{M}}_H(t)+\beta\mathbf{D}_1(t)\bar{\mathbf{M}}_H(t)\mathbf{D}_1(t)$, $\beta\mathbf{D}_1(t)\mathbf{W}_{HX}(t)$, $\mathbf{W}_{YH}(t)^T\mathbf{D}_2(t)$, and $\bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)$ are computed outside the iterative loop of Algorithm 1, the expressions in (9) and (11) require $2n^2+mn$ and $2n^2$ multiplications, respectively. If we assume that the neural dynamic loop reaches to the pre-determined maximum number of iterations τ_{max}, i.e., the numerical relative error check for the convergence is not satisfied, then the total number of multiplication is dominated by the factor $\tau_{\text{max}}(4n^2+mn)$. If we analyze the computational requirements of the factors $(1-\beta)\bar{\mathbf{M}}_H(t)+\beta\mathbf{D}_1(t)\bar{\mathbf{M}}_H(t)\mathbf{D}_1(t)$, $\beta\mathbf{D}_1(t)\mathbf{W}_{HX}(t)$, $\mathbf{W}_{YH}(t)^T\mathbf{D}_2(t)$, and $\bar{\mathbf{M}}_Y(t)\mathbf{D}_2(t)$, these calculations require multiplications of $(n^2-n)/2+n^2+3n$, $mn+n$, n^2, and n^2, respectively, since $\mathbf{D}_1(t)$ and $\mathbf{D}_2(t)$ are diagonal matrices and $\bar{\mathbf{M}}_H(t), \bar{\mathbf{M}}_Y(t)$ are symmetric matrices. Therefore, the complexity of the neural dynamics of our proposed approach is dominated by the factor of $\tau_{\text{max}}(4n^2+mn)$. The complexity of the update rules of the gain variables expressed in equations (13) and (14) is dominated by the multiplication factor of $3n$ for all $2n$ variables, leading to the dominant multiplication factor of $6n^2$. Moreover, the update rules of the synaptic weight updates expressed in equation (15) are dominated by the multiplication factor of n^2 or mn, leading to $4(n^2+mn)$ number of multiplications. Therefore, the worst-case complexity of our proposed method per sample in terms of the big-O notation is $\mathcal{O}(\tau_{\text{max}}mn)$.

We now compare this with the complexity of the NSM and BSM algorithms. We first consider the prewhitening layer introduced in [16], as both algorithms require input to be prewhitened. Taking into account equations (28), (29), and (30) in [16] for output computation and synaptic weight updates of the prewhitening layer, the complexity can be expressed in terms of big-O notation as $\mathcal{O}(\tau_{\text{max}}^{(\text{NSM})}(m+k)n)$, where $k \geq n$ is an integer introduced as a result of the Lagrangian multiplier in equation (12) in the reference [16], and $\tau_{\text{max}}^{(\text{NSM})}$ is the maximum predetermined number of iterations for the neural dynamic loop of NSM (see equation (28) and (33) in the reference). The output dynamics and the synaptic weight updates of the second layer of the online NSM network is described by the equations (33), (34), and (35) in [16] which lead to the complexity in terms of big-O notation of $\mathcal{O}(n^2)$. As a result, the overall complexity of the NSM algorithm per sample can be stated as $\mathcal{O}(\tau_{\text{max}}^{\text{NSM}}(m+k)n)$. Similar to the WSM network, the neural dynamic loop of BSM has

the complexity of $\mathcal{O}(\tau_{\text{max}}^{(\text{BSM})} mn)$ as a result of recursion defined by Equation (17) in [18], where $\tau_{\text{max}}^{(\text{BSM})}$ is the predetermined maximum number of iterations for the neural dynamic loop of BSM. Furthermore, synaptic weight and gain updates introduce $\mathcal{O}(n^2)$ complexity similar to the WSM algorithm. Therefore, combining with prewhitening, the overall complexity of the BSM algorithm per sample becomes $\mathcal{O}(\tau_{\text{max}}^{(\text{BSM})}(m+k)n)$

In conclusion, for the biologically plausible neural network solutions to the blind source separation problem, the overall complexity is determined by the recursive neural dynamic loops due to the implicit definition of the network output. Although this condition makes the implementation of such algorithms less feasible for digital hardware, they enable low-power implementations in future analog neuromorphic systems with local learning constraints.