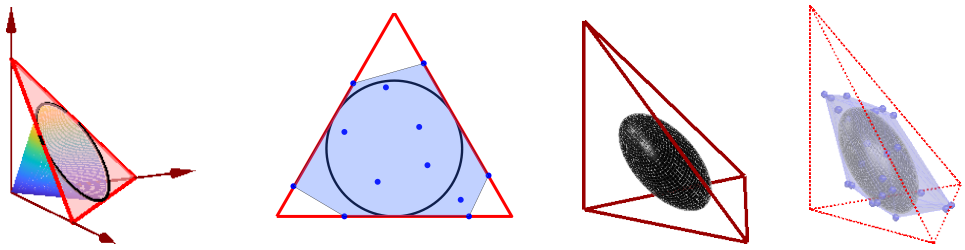
Appendix

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

$$\mathbf{\Gamma}_Y(t) \quad = \quad \text{diag}(\mathbf{M}_{Y\,11}(t), \mathbf{M}_{Y\,22}(t), \ldots, \mathbf{M}_{Y\,dd}(t)),$$

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 $D_{1,ii}(t)$:

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

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

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

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 $\mathbf{y}_t$:

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

Figure 6 shows the corresponding two-layer neural network.

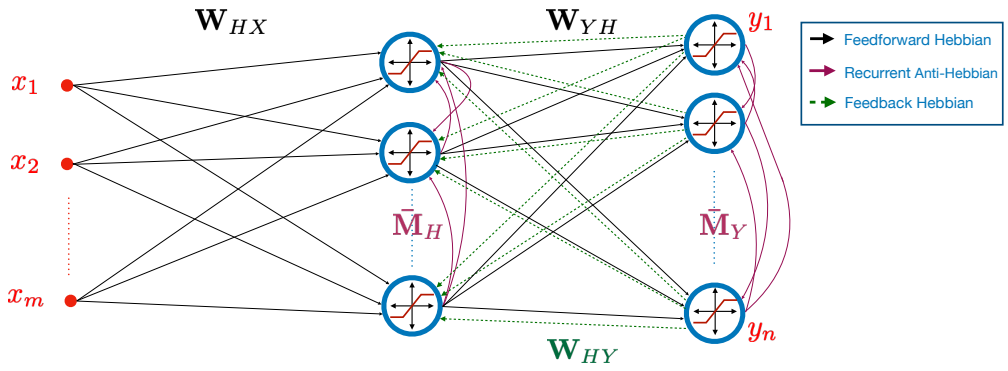


Figure 6: WSM Det-Max neural network for antisparse sources ($\mathcal{P} = \mathcal{B}_\infty$).

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 $\mathbf{h}_t$:

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

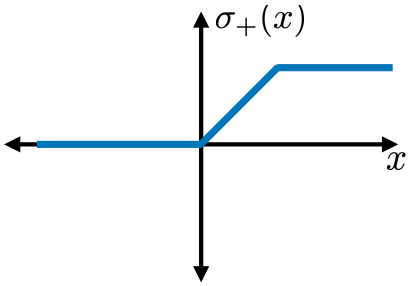


Figure 7: Nonnegative clipping function for elementwise projection to $\mathcal{B}_{\infty,+}$.

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

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

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

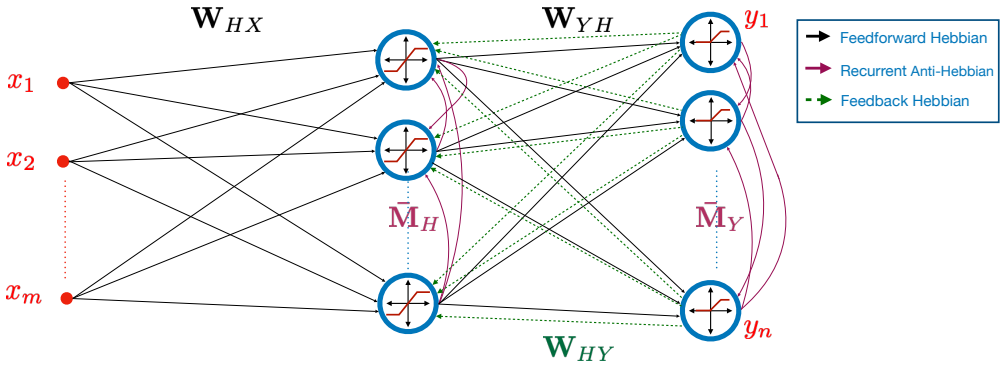


Figure 8: WSM Det-Max neural network for nonnegative anti-sparse sources ($\mathcal{P} = \mathcal{B}_{\infty,+}$).

D.3 Nonnegative sparse sources

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

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

for which the Lagrangian based reformulation can be written as

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

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

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

where $\mathrm{ReLU}(x, \lambda_1)$ is the rectified-linear unit mapping defined by $\mathrm{ReLU}(x) = \begin{cases} x & x > 0, \\ 0 & \text{otherwise} \end{cases}$.

Based on the dual maximization, the Lagrangian variable $\lambda_1(\tau)$ is updated by

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

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.

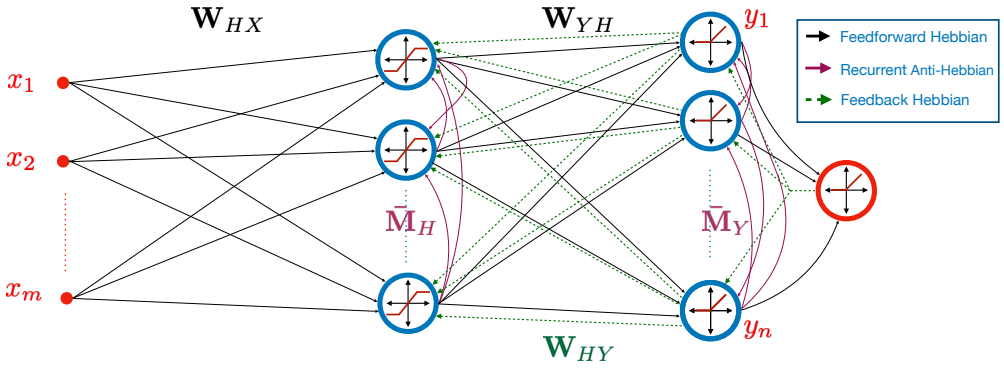


Figure 9: WSM Det-Max neural network for nonnegative sparse sources ($\mathcal{P} = \mathcal{B}_{1,+}$).

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

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

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

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

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:

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

Therefore, for the Lagrangian based formulation

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

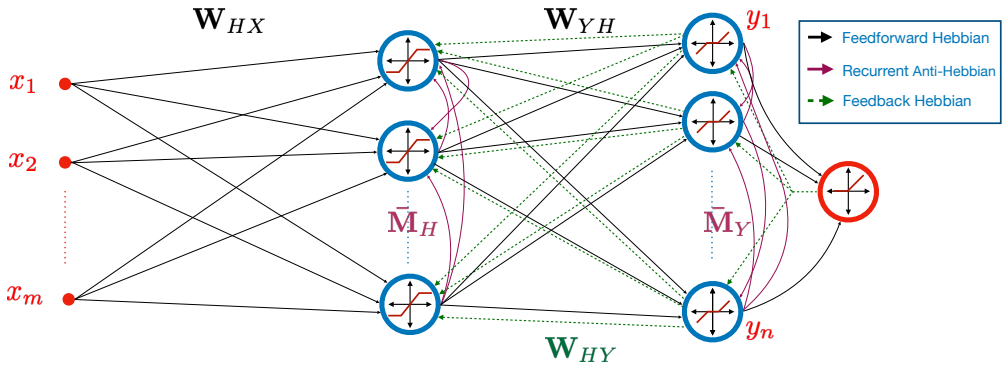


Figure 10: WSM Det-Max neural network for sparse sources ($\mathcal{P} = \mathcal{B}_1$).

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

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

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

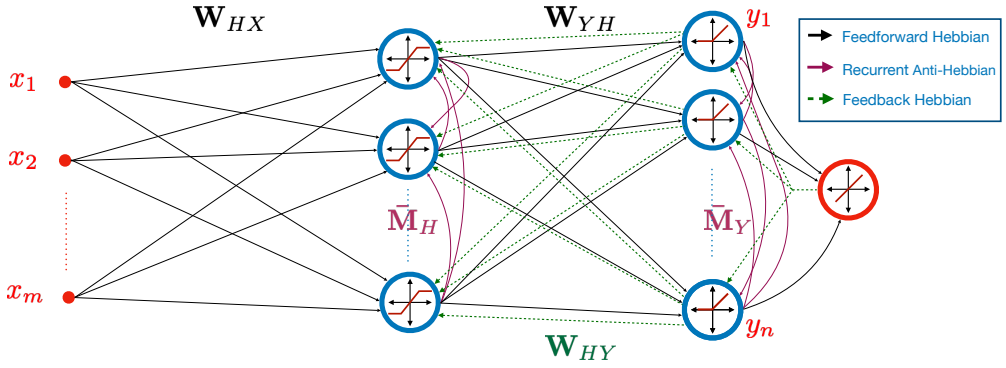


Figure 11: WSM Det-Max neural network for unit-simplex sources ($\mathcal{P} = \Delta$).

D.6 Sources with mixed attributes

We consider the following polytope example provided in [26]

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

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

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

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).

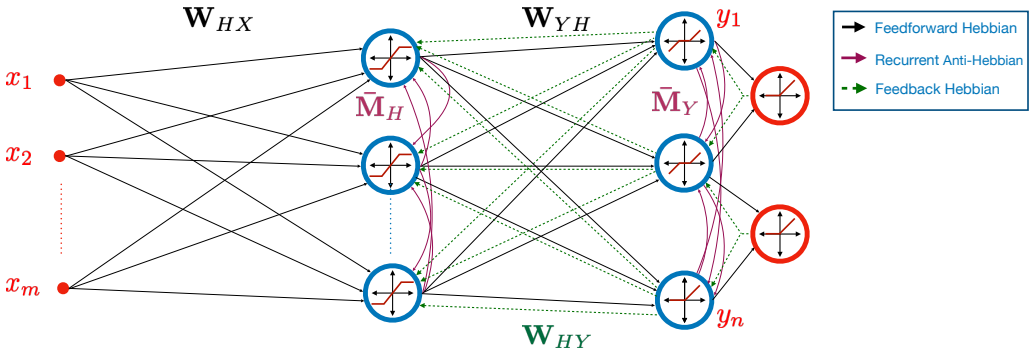


Figure 12: WSM Det-Max neural network for the polytope in (A.17).

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.

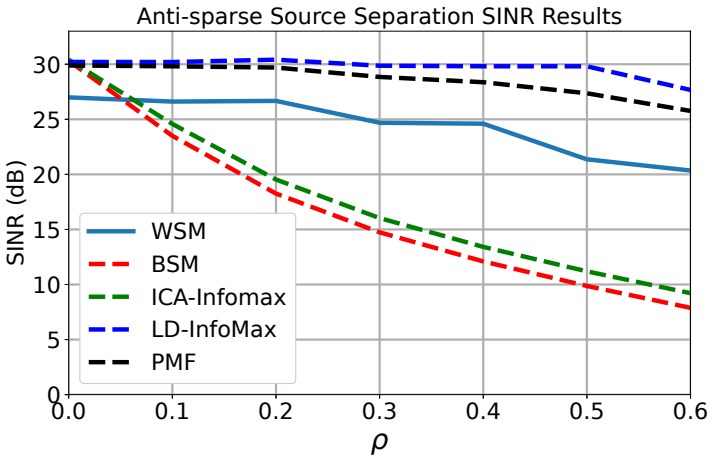


Figure 13: The SINR performances of the WSM, BSM, ICA, LD-InfoMax, and PMF algorithms as a function of the correlation parameter ρ.

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 $\tau_{\max} = 750$ 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).

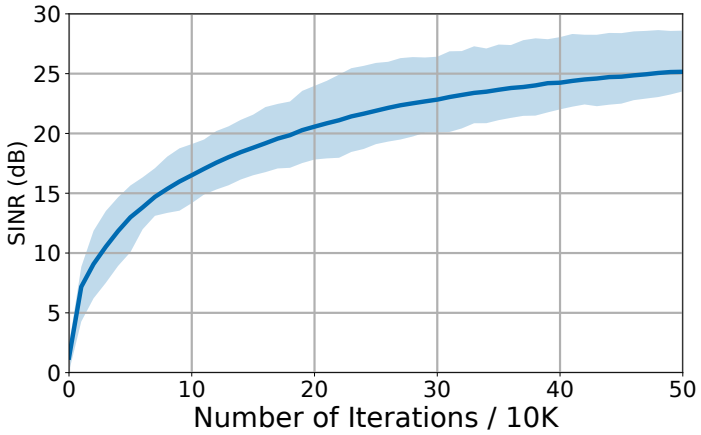


Figure 15: The SINR convergence curve for the sparse-WSM for 30dB input SNR level: mean-solid line with 25/75-percentile envelope.

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

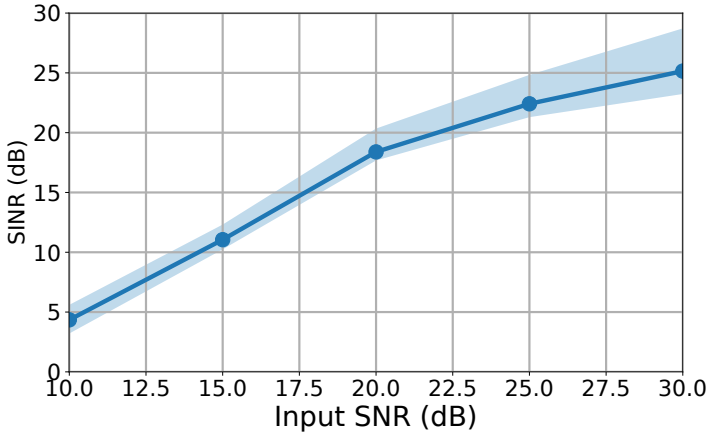


Figure 16: The output SINR with respect to the input SNR level for the sparse-WSM: mean-solid line with 25/75-percentile envelope.

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.

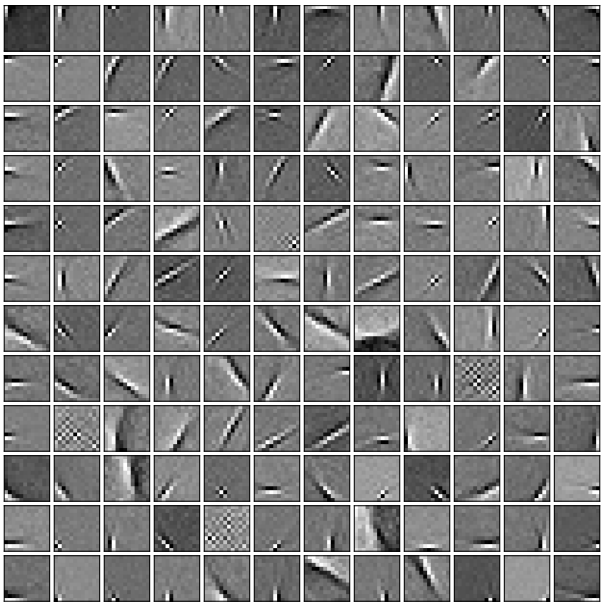


Figure 17: Dictionary obtained from the natural image patches by the sparse-WSM Network.

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.

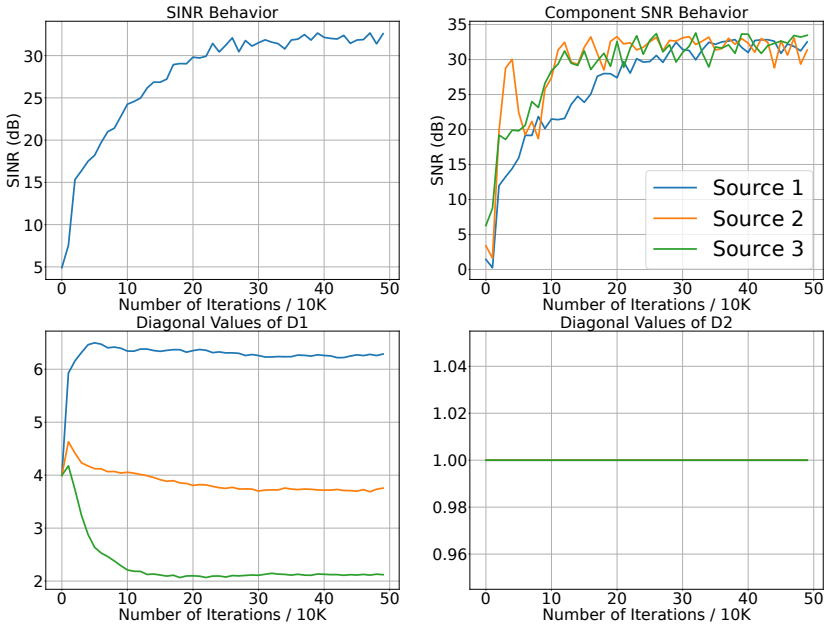


Figure 18: Example behaviors of SINR, component SNR values, and diagonal weights of $\mathbf{D}_1$ and $\mathbf{D}_2$ for a single experiment discussed in E.7.1

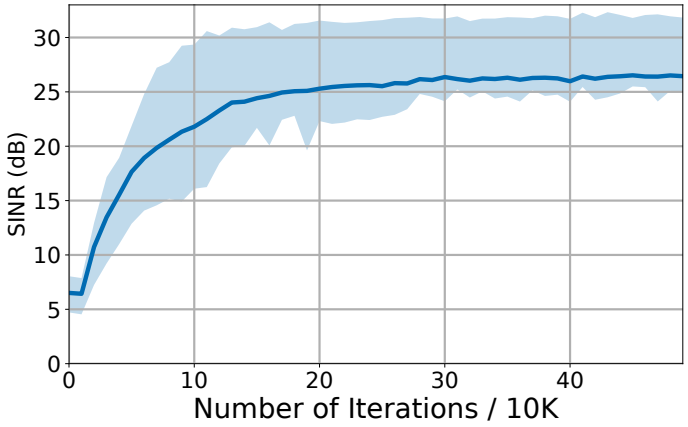


Figure 19: The SINR convergence curve for the experiments discussed in E.7.1: mean-solid line with 25/75-percentile envelope.

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 $\tau_{\max} = 750$ 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 $\tau_{\max} = 750$ 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 $0.5 \prec \text{diag}(\mathbf{D}_2) \prec 5$.

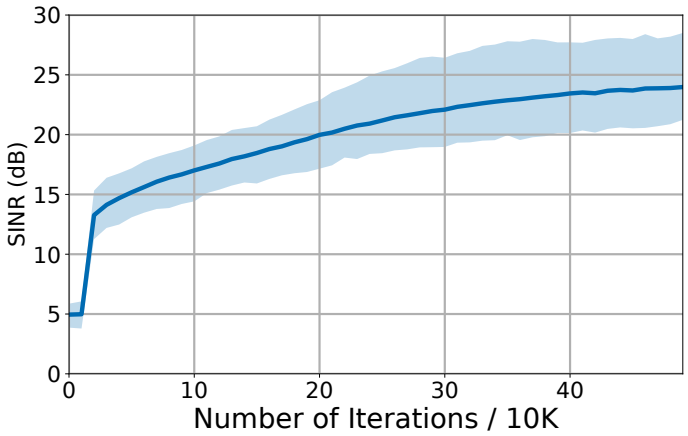


Figure 20: The SINR convergence curve for the experiments discussed in E.7.2: mean-solid line with 25/75-percentile envelope.

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.

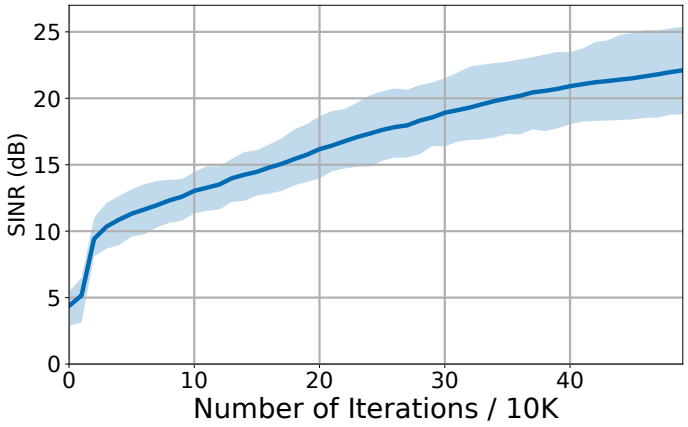


Figure 21: The SINR convergence curve for the experiments discussed in E.7.3; mean-solid line with 25/75-percentile envelope.

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.

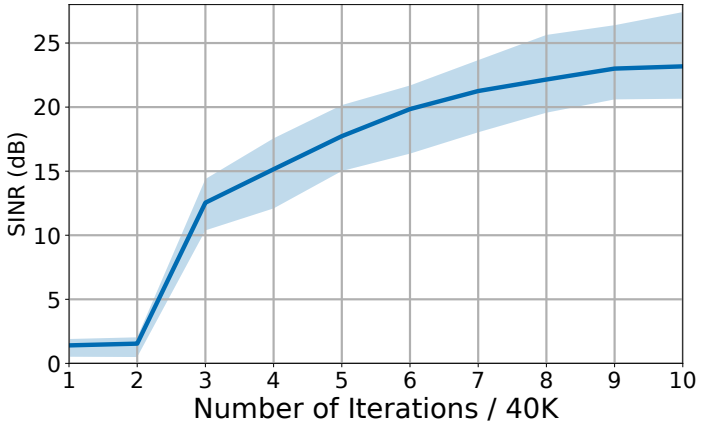


Figure 22: The SINR convergence curve for the 4-PAM digital communication signals: mean solid line with 25/75-percentile envelope.

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

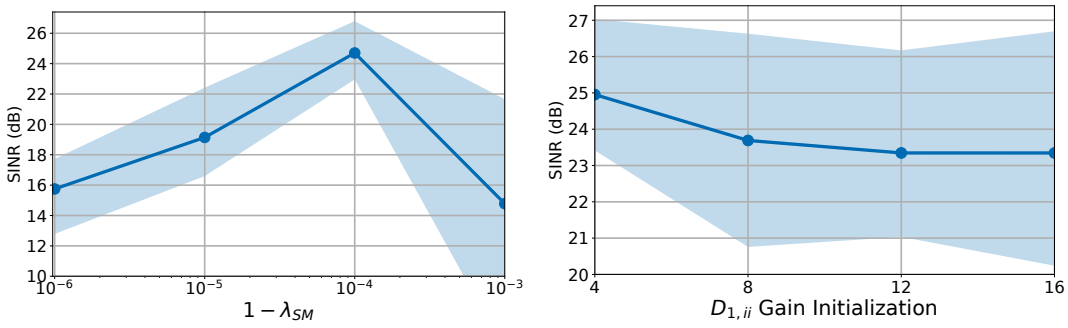
illustrated in Figure 9. In these ablation studies, we specifically consider the effect of hyperparameter selection for $1-\lambda_{\text{SM}}$, initial $\mathbf{D}_1$, $\mu_{\mathbf{D}_1}$, and initial $1-\gamma^2$. For each of the mentioned hyperparameters, we consider the following choices,

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

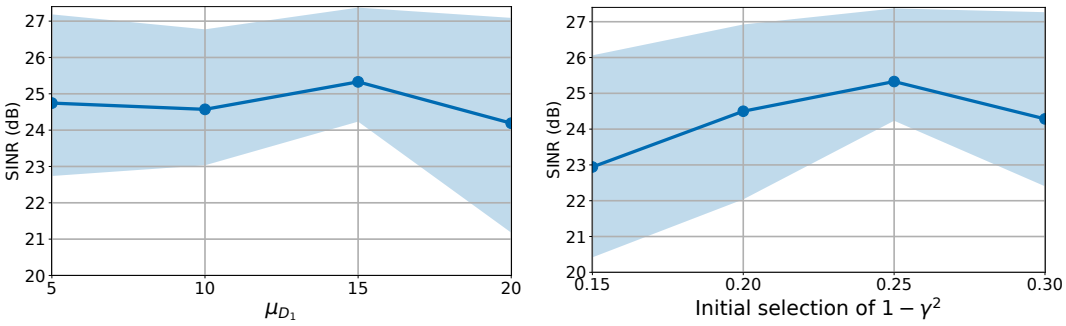
While experimenting with one hyperparameter, we fixed the rest of them as given in the following list,

- $\mathbf{D}_1 = 4\mathbf{I}$, 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.