Normative Networks for Source Separation via Local Plasticity and Dendritic Computation

Bariscan Bozkurt[1,2,*†] **Efe Ali Gorguner**[3,†] **Francesco Innocenti**[2,4] **Rafal Bogacz**[2,4]
[1]Gatsby Computational Neuroscience Unit, University College London, UK
[2]Brain Network Dynamics Unit, University of Oxford, UK
[3]Department of Computer Science, University of Oxford, UK
[4]MRC Centre of Research Excellence in Restorative Neural Dynamics, UK

Abstract

Blind source separation (BSS) is a natural framework for studying how latent causes may be recovered from sensory mixtures, but deriving online and biologically plausible algorithms for structured (i.e., constrained to known domains) and potentially correlated sources remains challenging. Recent work has derived neural networks for BSS from maximization of an entropy measure, yet its online implementations involve complex and nonlocal recurrent dynamics. Motivated by this perspective, we propose *Predictive Entropy Maximization*, which achieves competitive performance in BSS, using only local weight updates. The method employs a close approximation of an entropy measure, yielding an objective function with easily interpretable components. Minimizing this objective leads to a predictive neural architecture in which feedforward synapses follow an error-driven rule (that can be realized through dendritic mechanisms), lateral inhibitory connections are learned with local Hebbian plasticity, and source-domain constraints are enforced through simple output nonlinearities. We derive explicit spectral bounds on the surrogate error, characterizing when the approximation is accurate. Empirically, *Predictive Entropy Maximization* remains robust under increasing source correlation and observation noise, outperforms biologically plausible algorithms that rely on stronger independence or decorrelation assumptions, and remains competitive with exact determinant- and correlative-information-based baselines. These results show how local plasticity and adaptive lateral inhibition can emerge from maximizing a regularized second-order entropy over structured source domains.

1 Introduction and related works

Natural sensory inputs are rarely generated by a single cause. For example, in audition, several sound sources may overlap, while in vision, objects, textures, and illumination are superimposed. The problem of recovering the underlying causes from mixtures alone is known as blind source separation (BSS) [1, 2]. BSS is a classical problem in signal processing and machine learning, and it has also served as a normative model of sensory representation learning in neuroscience. Sparse and independent decompositions of natural stimuli produce features reminiscent of receptive fields in early sensory systems [3, 4, 5], while the cocktail-party problem illustrates the broader challenge of source separation in the presence of strong interference [6, 7, 8].

Because BSS is ill-posed without additional structure, identifiability must come from assumptions on the latent sources. Independent component analysis (ICA), including the information-

*Work completed in part while visiting the University of Oxford.
†Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

maximization (InfoMax) formulation, achieves this through statistical independence [3, 9, 10]. A complementary geometric line of work instead assumes that the sources lie in a structured domain—for example, a simplex (nonnegative entries summing to one) [11, 12], a sparse or bounded set [13, 14], the nonnegative orthant (all coordinates nonnegative) [15, 16], or, more generally, a polytope (a convex bounded set with finitely many vertices) [17, 18, 19]. These geometric approaches seek outputs that spread across the available dimensions of the source domain, through maximization of the determinant of the covariance matrix of output signals. An appealing feature of this approach is that it retains information about inputs as it avoids degenerate low-rank solutions. Another useful property of these methods is that it can recover dependent or correlated sources (under suitable scattering assumptions) [11, 17]. Recent work further shows that determinant-based structured source separation admits an information-theoretic interpretation through a second-order entropy measure, called *correlative entropy* (based on the log-determinant of the regularized output covariance matrix) [20].

A separate challenge, central to neuroscience and neuromorphic computing, is how to realize such objectives with local dynamics and plasticity. Biologically plausible BSS networks have been developed for ICA settings [21, 22, 23], for nonnegative and bounded source separation through similarity matching [24, 25], and more recently for correlated-source separation under determinant-based objectives [26, 27, 28]. These works show that source-domain structure can be implemented through neural nonlinearities, inhibition, and local learning rules. Nonetheless, the online determinant-based formulations for correlated sources still tend to induce complex recurrent dynamics, because optimizing the exact log-determinant objective naturally brings inverse-covariance into the update.

Our goal is therefore to retain the robustness of determinant-based correlated-source separation while replacing these dynamics with a more directly local and biologically plausible recurrent implementation. Specifically, we strive to derive neural networks which satisfy previously proposed criteria of biological plausibility [29]: (i) local computation, where a neuron's activity is only based on its inputs, and (ii) local plasticity, where weight modification is based on the variables encoded in pre- and post-synaptic neurons. To this end, we replace the exact log-determinant with an online second-order surrogate obtained by a Taylor expansion around the diagonal part of the output covariance. The resulting objective decomposes into two interpretable terms: one encourages large variance in each output dimension, while the other penalizes redundant dependencies through normalized cross-covariances. This, in turn, yields a predictive neural architecture with a direct local interpretation: feedforward synapses encode the mapping from inputs to recovered sources and follow a local error-driven rule, recurrent interactions are mediated by local covariance traces and implement adaptive lateral inhibition, and source-domain constraints are enforced through simple output nonlinearities. In this way, we preserve the determinant-based geometric viewpoint while obtaining a more directly local and biologically plausible recurrent implementation.

Beyond determinant-based BSS, our construction is also connected to two adjacent lines of work. First, our approach is related to mechanistic models of learning in neural circuits. These models include predictive coding which emphasizes recurrent inference and local prediction errors as a route to learning [30, 31], and supervised Correlative Information Maximization (CorInfoMax) which shows that predictive pathways and lateral interactions can arise from a log-determinant objective that encourages layer activities to spread across their available dimensions [32]. Likewise, predictive-plasticity models highlight the joint role of predictive and Hebbian mechanisms in sensory representation learning [33]. Second, more abstractly, our surrogate objective is also related to redundancy-reduction methods from self-supervised learning, such as VICReg, which combine variance preservation with penalties on cross-component correlations [34]. These latter methods are not biologically plausible models, but they provide a useful point of comparison for the variance-covariance structure of our objective. Our setting remains distinct from both lines of work, since we aim to tackle BSS in the bio-plausible setting, where observations arrive one at a time as linear mixtures of latent sources, and the recovered sources must satisfy explicit domain constraints.

The main contributions of our *Predictive Entropy Maximization* model are:

- A Taylor approximation of the online determinant-maximization objective, obtained by expanding the log-determinant of the regularized output covariance around its diagonal part. Unlike exact online determinant-based formulations, the resulting updates depend directly on covariance among outputs rather than the inverse of the covariance matrix.
- A family of predictive neural architectures for different source domains, induced by the proposed approximation. Feedforward synapses follow local error-driven plasticity, recurrent interactions

are mediated by local covariance traces, and source-domain constraints are enforced through simple output nonlinearities, as illustrated in Figure 1.
- A theoretical characterization of the surrogate, including explicit error bounds (derived through an exact spectral representation of the Taylor remainder).
- Empirical evidence that our method remains robust with increasing source correlation and observation noise, competes with bio-plausible baselines built on stronger decorrelation assumptions, and extends naturally to auditory source separation and sparse receptive-field learning.

The remainder of the paper is organized as follows. After defining the BSS setting and the determinant-maximization objective (Section 2), we derive the online Taylor-approximated objective and the resulting predictive neural dynamics (Section 3). We then present numerical results evaluating our model (Section 4), concluding with a summary, limitations and future directions (Section 5).

2 Problem setup for blind source separation

We now review the formulation of the linear BSS setting considered in this work [1, 2]. Each observation vector $\boldsymbol{x}(t) \in \mathbb{R}^m$ is generated by linearly mixing an unknown source vector $\boldsymbol{s}(t) \in \mathcal{P} \subset \mathbb{R}^n$ through an unknown matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$:

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t), \qquad t \in \{1, 2, \ldots, T\}.$$

Here, the set $\mathcal{P}$ denotes a prescribed source domain that encodes structural assumptions on the source vectors. We focus on the determined or overdetermined regime ($m \geq n$) and assume that $\boldsymbol{A}$ has full column rank. Stacking the samples in the snapshot matrices $\boldsymbol{S} = [\boldsymbol{s}(1), \ldots, \boldsymbol{s}(T)]$ and $\boldsymbol{X} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(T)]$, the model can be written compactly as $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$. The goal of BSS is to recover the source vectors from the mixtures alone by learning a separator $\boldsymbol{W} \in \mathbb{R}^{n \times m}$ such that the outputs $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ match the original sources. Since linear BSS is non-unique without additional structural assumptions on $\mathcal{P}$, successful recovery is defined only up to permutation and sign ambiguities, as formalized next.

Definition 2.1 (Ideal separation)**.** *A separator $\boldsymbol{W}$ achieves ideal separation if the recovered outputs satisfy $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X} = \boldsymbol{\Pi}\boldsymbol{\Lambda}\boldsymbol{S}$, where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{\Lambda}$ is a diagonal sign matrix.*

The BSS model is *identifiable* over a source domain $\mathcal{P}$ if every exact factorization $\boldsymbol{X} = \boldsymbol{A}'\boldsymbol{S}'$ with columns of $\boldsymbol{S}'$ in $\mathcal{P}$ agrees with the true factorization up to the ideal-separation ambiguity of Definition 2.1. For the source domains considered here, this identifiability property is established in the determinant-maximization literature for polytopic and simplex-structured source models [11, 18].

Determinant-maximization. A natural way to exploit the geometry of $\mathcal{P}$ is through the determinant-maximization criterion, which selects, among all feasible outputs constrained to lie in $\mathcal{P}$, those with maximal second-order spread as measured by the log-determinant of their sample autocorrelation or autocovariance matrix [12, 17, 20]. The corresponding recovery guarantees additionally require the source samples to be sufficiently scattered in $\mathcal{P}$. Geometrically, this means that the convex hull of the source samples must capture the shape of $\mathcal{P}$ more faithfully than its best ellipsoidal approximation [12, 17].

Accordingly, we consider the batch determinant-maximization problem

$$\underset{\boldsymbol{W},\boldsymbol{Y}}{\text{minimize}} \quad -\log\det(\hat{\boldsymbol{C}}) \quad \text{subject to} \quad \boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t), \ \text{and} \ \boldsymbol{y}(t) \in \mathcal{P}, \quad t = 1, \ldots, T. \tag{1}$$

In the above objective, $\hat{\boldsymbol{C}}$ is the output autocovariance matrix for the sample (batch) defined as:

$$\hat{\boldsymbol{C}} = \frac{1}{T}\sum_{t=1}^{T}\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}\big)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}\big)^{\top}, \tag{2}$$

where $\hat{\boldsymbol{\mu}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}(t)$ is the output mean in the sample.

Entropy interpretation. The above optimization problem has an information-theoretic interpretation. It has been established [20] that the regularized log-determinant of a sample covariance matrix is approximately proportional to a deterministic second-order entropy measure, which we refer to as

the *correlative entropy* (CE):

$$\hat{\mathcal{H}}^{(\varepsilon)}_{\mathrm{CE}}(\boldsymbol{Y}) = \frac{1}{2}\log\det(\hat{\boldsymbol{C}} + \varepsilon\boldsymbol{I}) + \frac{n}{2}\log(2\pi e), \tag{3}$$

where $\varepsilon > 0$ is a small regularization constant. Under this interpretation, the optimization problem in Eq. 1 can be viewed as maximizing the output correlative entropy subject to the source-domain and mixing constraints. Further details on this second-order entropy and the related correlative-information definitions are given in Appendix A.

Source domains. In this paper, we consider the following source domains, each encoding a different structural assumption on the sources, for which the BSS model is identifiable under the scattering assumptions discussed above:

- *Antisparse sources:* $\mathcal{B}_{\max} := \{\boldsymbol{s} \in \mathbb{R}^n : \max_i |s_i| \leq 1\}$. This domain models sources with bounded signed amplitudes.
- *Nonnegative antisparse sources:* $\mathcal{B}_{\max,+} := \mathcal{B}_{\max} \cap \mathbb{R}^n_+$, where $\mathbb{R}^n_+$ is the nonnegative orthant. This domain models bounded nonnegative features, as in our image feature-learning experiments (Appendix E.2).
- *Sparse sources:* $\mathcal{B}_1 := \{\boldsymbol{s} \in \mathbb{R}^n : \|\boldsymbol{s}\|_1 \leq 1\}$, where $\|\cdot\|_1$ denotes the ℓ_1 norm. The ℓ_1 constraint limits total absolute activity and supports sparse representations, as illustrated by our natural-image dictionary-learning experiment (Figure 3).
- *Nonnegative sparse sources:* $\mathcal{B}_{1,+} := \mathcal{B}_1 \cap \mathbb{R}^n_+$. This domain combines nonnegative activity with a bound on its total amplitude.
- *Simplex sources:* $\Delta := \{\boldsymbol{s} \in \mathbb{R}^n : s_i \geq 0\ \forall i,\ \sum_i s_i = 1\}$. This domain represents probability vectors or nonnegative source proportions that sum to one.

Figure 5 in Appendix C illustrates these source domains in three-dimensional Euclidean space. It also shows the corresponding output nonlinearities, which we derive later for the proposed networks.

3 Online surrogate objective for blind source separation

In this section, we derive an online approximation to the regularized correlative-entropy objective in Eq. 3 and show how it leads to biologically plausible learning dynamics. We also summarize the corresponding approximation error.

From batch correlative entropy to an online objective. To compute the correlative-entropy objective online, we replace the batch output covariance by an exponentially weighted estimate, following the standard streaming-statistics construction also used in online CorInfoMax [28]. At time t, the algorithm receives $\boldsymbol{x}(t)$, infers $\boldsymbol{y}(t)$ using pre-update weights and statistics, and then updates these state variables. Specifically, we replace $\hat{\boldsymbol{C}}$ defined in Eq. 3 with:

$$\hat{\boldsymbol{C}}^{\lambda}(t) = \frac{1}{\eta(t)}\sum_{t'=1}^{t}\lambda^{\,t-t'}\big(\boldsymbol{y}(t') - \hat{\boldsymbol{\mu}}^{\lambda}(t')\big)\big(\boldsymbol{y}(t') - \hat{\boldsymbol{\mu}}^{\lambda}(t')\big)^{\top}, \tag{4}$$

where $\lambda \in (0,1)$ is a forgetting factor, $\eta(t) = \sum_{t'=1}^{t}\lambda^{\,t-t'}$, and $\hat{\boldsymbol{\mu}}^{\lambda}(t) = \eta(t)^{-1}\sum_{t'=1}^{t}\lambda^{\,t-t'}\boldsymbol{y}(t')$. The corresponding exact finite-time recursions are given in Appendix B. Since $\eta(t)^{-1} \to 1-\lambda$ as $t \to \infty$, the estimators can be updated online based on a new sample $\boldsymbol{y}(t)$:

$$\hat{\boldsymbol{\mu}}^{\lambda}(t) = \lambda\hat{\boldsymbol{\mu}}^{\lambda}(t-1) + (1-\lambda)\boldsymbol{y}(t), \tag{5}$$

$$\hat{\boldsymbol{C}}^{\lambda}(t) = \lambda\hat{\boldsymbol{C}}^{\lambda}(t-1) + (1-\lambda)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)^{\top}. \tag{6}$$

Thus, the exact online counterpart of the batch problem is to minimize the negative output-entropy term, $-\log\det(\hat{\boldsymbol{C}}^{\lambda}(t) + \varepsilon\boldsymbol{I})$, under the source-domain constraint $\boldsymbol{y}(t) \in \mathcal{P}$ and the input-output relation $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$. Existing online formulations based on correlative information maximize a related mutual-information objective built from this output-entropy term together with an additional error-entropy term [28]. Direct gradients of these log-determinant objectives involve inverse covariance or correlation matrices, leading to inverse second-order states updated through the matrix inversion lemma. These inverse-state updates are less directly synapse-local than covariance-trace updates, making the local-plasticity interpretation less immediate; see Appendix A.4. In contrast, our approach introduced below avoids this step altogether. By replacing the exact regularized

output-entropy term with a second-order Taylor surrogate, we obtain an objective whose gradients depend directly on covariance statistics rather than on their inverse. This preserves the entropy-maximization viewpoint while leading to recurrent interactions and plasticity rules with a more direct local interpretation.

Approximate entropy objective. We present here our first result - an approximate objective that can be optimized for BSS. Since below we apply a Taylor expansion around the diagonal of the covariance estimate $\hat{\boldsymbol{C}}^{\lambda}(t)$, we decompose it as $\hat{\boldsymbol{C}}^{\lambda}(t) = \hat{\boldsymbol{D}}^{\lambda}(t) + \hat{\boldsymbol{O}}^{\lambda}(t)$, where $\hat{\boldsymbol{D}}^{\lambda}(t)$ is diagonal and $\hat{\boldsymbol{O}}^{\lambda}(t)$ collects the off-diagonal cross-covariances. We apply a second-order Taylor expansion to the $\log\det$ objective of the regularized covariance, as formalized in Theorem D.1 of Appendix D,

$$-\log\det\big(\hat{\boldsymbol{C}}^{\lambda}(t) + \varepsilon\boldsymbol{I}\big) = \underbrace{-\sum_{i=1}^{n}\log\big(\hat{v}_i(t)+\varepsilon\big)}_{\text{variance expansion}} + \underbrace{\frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{\hat{c}_{ij}(t)^2}{\big(\hat{v}_i(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)}}_{\text{normalized covariance penalty}} - R_2(t), \quad (7)$$

where $R_2(t)$ denotes the Taylor remainder, $\hat{v}_i(t) := [\hat{\boldsymbol{D}}^{\lambda}(t)]_{ii} = [\hat{\boldsymbol{C}}^{\lambda}(t)]_{ii}$ denotes the i-th output variance, and $\hat{c}_{ij}(t) := [\hat{\boldsymbol{O}}^{\lambda}(t)]_{ij} = [\hat{\boldsymbol{C}}^{\lambda}(t)]_{ij}$ for $i \neq j$ denotes the corresponding cross-covariance. Up to the additive constant in the entropy definition, maximizing the output correlative entropy is therefore equivalent to minimizing the right-hand side of Eq. 7. The resulting surrogate decomposes this objective into two interpretable terms: (i) a variance-expansion term and (ii) a variance-normalized cross-covariance penalty. A first-order approximation would retain only variance expansion, since the linear correction in the off-diagonal covariance vanishes. Second order introduces a pairwise covariance penalty, providing the structure from which we derive a network with local computation and synaptic plasticity, as required by our criteria for biological plausibility. Higher-order corrections involve products of covariance traces across multiple neuron pairs and would require additional computation beyond this direct pairwise construction; see Remark D.2. Because the covariance penalty acts jointly with the variance-expansion term, the surrogate does not enforce exact decorrelation: uncorrelated outputs do not necessarily minimize it. Instead, it favors outputs with large component-wise variance and sufficiently small *normalized* cross-covariances, subject to the geometric constraint $\mathcal{P}$. This is precisely the regime relevant to structured source recovery: successful separation does not require zero cross-covariances, but rather nondegenerate spread within the source domain together with controlled inter-component dependence. This also explains why the method can recover correlated sources beyond the scope of ICA-based approaches built on mutual independence, as demonstrated by our numerical results in Section 4.

Bound for the surrogate approximation error. The error $|R_2(t)|$ resulting from the above approximation can be quantified directly. Let $\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t) := \hat{\boldsymbol{D}}^{\lambda}(t) + \varepsilon\boldsymbol{I}$, and define $\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t) := (\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t))^{-1/2}\hat{\boldsymbol{O}}^{\lambda}(t)(\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t))^{-1/2}$. Corollary D.5 in Appendix D shows that the Taylor remainder $R_2(t)$ in Eq. 7 satisfies

$$|R_2(t)| \leq \frac{\|\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t)\|_F^2\|\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t)\|_2}{3\big(1+\lambda_{\min}(\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t))\big)}. \quad (8)$$

Here, $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue, $\|\cdot\|_2$ denotes the spectral norm, and $\|\cdot\|_F$ denotes the Frobenius norm. The numerator $\|\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t)\|_F^2\|\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t)\|_2$ measures the overall size of the regularized normalized off-diagonal covariance: the Frobenius norm captures its aggregate magnitude, while the spectral norm captures its largest absolute eigenvalue. The denominator $1+\lambda_{\min}(\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t))$ shows that the bound worsens as the smallest eigenvalue approaches -1; hence the surrogate is accurate when the normalized off-diagonal covariance is small and the spectrum stays away from -1. In Section 4, Figure 4 reports a diagnostic of the resulting error bound as source correlation varies.

Two-timescale optimization. Since biological neural networks can only learn online, we wish to generate the output $\boldsymbol{y}$ and then update the parameters $\boldsymbol{W}$ based on the currently provided input $\boldsymbol{x}$. Therefore at each step or sample t, first the output $\boldsymbol{y}$ is optimized, and then the weights $\boldsymbol{W}$ are updated. To enable this bi-level optimization, we replace the hard relation $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ with a quadratic penalty, adding to the online objective of Eq. 7:

$$\mathcal{J}_t = -\sum_{i=1}^{n}\log\big(\hat{v}_i(t)+\varepsilon\big) + \frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{\hat{c}_{ij}(t)^2}{\big(\hat{v}_i(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)} + \gamma\|\boldsymbol{y}(t)-\boldsymbol{W}\boldsymbol{x}(t)\|_2^2, \quad (9)$$

and $\gamma > 0$ controls the strength of the prediction-error term.

To optimize Eq. 9, we adopt a two-timescale procedure (which naturally admits a neural interpretation as we will show later). For each incoming sample, the current output $\boldsymbol{y}(t)$ is first updated on a fast timescale. During this fast relaxation, the input $\boldsymbol{x}(t)$, the weights $\boldsymbol{W}$, and the running statistics are held fixed, and the output $\boldsymbol{y}(t)$ is updated until the value minimizing $\mathcal{J}_t$ over $\boldsymbol{y}(t) \in \mathcal{P}$ is found. Once this fast inference stage has settled, the resulting output state is used to update the feedforward weights and the running second-order statistics on a slower timescale.

Inference of output activity. We first introduce notation that will help us to describe the inference process. For each streaming sample t, we introduce a faster neural-relaxation time index $\tau = 0, 1, \ldots, \tau_{\max}$. The index t labels the slow arrival of new input samples and the corresponding parameter updates, whereas τ labels the fast within-sample activity dynamics used to infer $\boldsymbol{y}(t)$. For notational convenience, let $\hat{\mu}_k(t) := [\hat{\boldsymbol{\mu}}^{\lambda}(t)]_k$ denote the k-th component of the running mean and we define the centered activity as $\bar{y}_k(t,\tau) = y_k(t,\tau) - \hat{\mu}_k(t)$.

During inference we change the output $\boldsymbol{y}$ to reduce the objective of Eq. 9. Using the gradient of Eq. 9, we define the fast activity-update direction as the negative truncated gradient $\boldsymbol{d}(t,\tau) \approx -\nabla_{\boldsymbol{y}}\mathcal{J}_t$, where the truncation discards the higher-order term quadratic in the off-diagonal covariance entries for biological plausibility; see Appendix B.3. Componentwise,

$$d_k(t,\tau) = \underbrace{\frac{\bar{y}_k(t,\tau)}{\hat{v}_k(t)+\varepsilon}}_{\text{variance drive}} - \underbrace{\sum_{\substack{j=1\\ j\neq k}}^{n} \frac{\hat{c}_{kj}(t)\,\bar{y}_j(t,\tau)}{\big(\hat{v}_k(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)}}_{\text{covariance reduction}} - \underbrace{\gamma\left(y_k(t,\tau) - \sum_{\ell=1}^{m} W_{k\ell}(t-1)x_\ell(t)\right)}_{\text{predictive correction}}. \tag{10}$$

The corresponding projected activity update is

$$y_k(t,\tau+1) = \sigma_{\mathcal{P}}\big(y_k(t,\tau) + \eta_y(t,\tau)\, d_k(t,\tau)\big)\,, \tag{11}$$

where $\sigma_{\mathcal{P}}$ projects its argument on $\mathcal{P}$ if it lies outside, and $\eta_y(t,\tau) > 0$ is the inference step size. Terms in Eq. 10 have intuitive interpretation. The first term in $d_k(t,\tau)$ promotes regularized variance expansion, the second reduces covariance, and the third pulls the output activity toward its feedforward prediction.

The form of $\sigma_{\mathcal{P}}$ in Eq. 11 is determined by the source domain. For box-type domains such as $\mathcal{B}_{\max}$ and $\mathcal{B}_{\max,+}$, it reduces to elementwise saturation, namely clipping to $[-1,1]$ or $[0,1]$. For $\mathcal{B}_1$, $\mathcal{B}_{1,+}$, and Δ, we instead project the output on the source domain as in standard proximal algorithms [35], and the corresponding domain-specific derivations are given in Appendix C.

Parameter updates. Once the fast inference dynamics have settled for sample t, the resulting output state is used to update the weights $\boldsymbol{W}$ and the running output statistics on a slower timescale. With the settled output $\boldsymbol{y}(t)$ held fixed, the only $\boldsymbol{W}$-dependent part of $\mathcal{J}_t$ is the quadratic prediction loss. A gradient descent step on $\frac{1}{2}\|\boldsymbol{y}(t) - \boldsymbol{W}\boldsymbol{x}(t)\|_2^2$ with learning rate $\alpha_W(t)$ gives

$$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) + \alpha_W(t)\boldsymbol{e}(t)\boldsymbol{x}(t)^\top, \qquad \boldsymbol{e}(t) := \boldsymbol{y}(t) - \boldsymbol{W}(t-1)\boldsymbol{x}(t), \tag{12}$$

as derived in Appendix B (Eq. B.8). The running mean is updated according to Eq. 5, and the centered activity $\bar{\boldsymbol{y}}(t) = \boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)$ is then used to update the cross-covariance and variance traces:

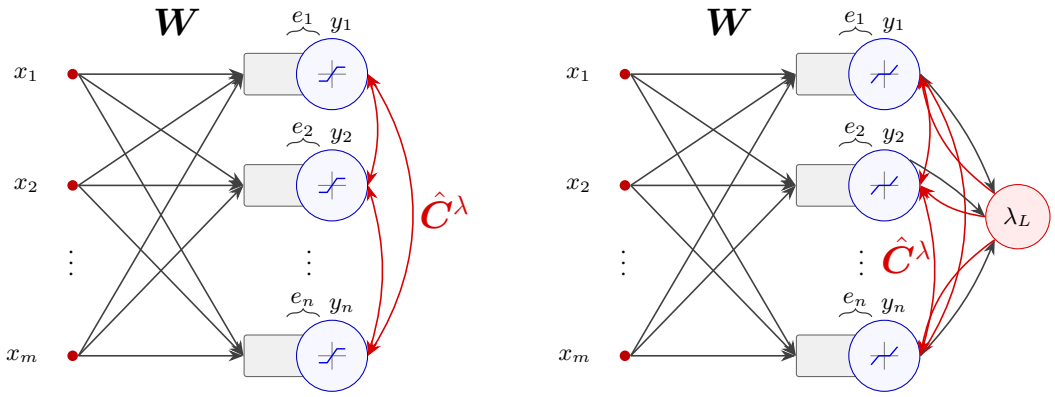
$$\hat{c}_{ij}(t) = \lambda \hat{c}_{ij}(t-1) + (1-\lambda)\bar{y}_i(t)\bar{y}_j(t), \;\; i \neq j, \tag{13}$$

$$\hat{v}_i(t) = \lambda \hat{v}_i(t-1) + (1-\lambda)\bar{y}_i(t)^2, \;\; i = 1, \ldots, n, \tag{14}$$

as derived in Appendix B (see Eq.s B.6 and B.7). We refer to the resulting procedure as *Predictive Entropy Maximization* (*PEM*) and summarize it in Algorithm 1 in Appendix C.

Neural implementation. PEM can be naturally implemented in a neural network with local weight update rules. In this implementation, we assume that output signals y_i are encoded in neural activity with dynamics on a fast time scale, while parameters W_{ij} and $\hat{c}_{ij}$ are encoded in the weights of connections between neurons which evolve on a slower time scale.

PEM naturally maps onto a two-layer network, where the input layer encodes the mixture inputs $\boldsymbol{x}$, while the output layer encodes the outputs $\boldsymbol{y}$. To make the network mapping explicit, we rewrite the



(a) Antisparse network

(b) Sparse network

Figure 1: **Representative Predictive Entropy Maximization architectures for two source domains. (a)** Antisparse architecture. Mixture inputs are mapped through feedforward weights $\boldsymbol{W}$ to local prediction compartments, each paired with an output unit y_k. Prediction errors e_k are computed as the differences between somatic and dendritic activity. The output layer is coupled through adaptive recurrent inhibitory interactions driven by the running output-covariance statistics $\hat{\boldsymbol{C}}^\lambda$, while the antisparse constraint is enforced locally by clipping. Black and red arrows denote excitatory and inhibitory connections, respectively. **(b)** Sparse architecture. The same predictive feedforward pathway and adaptive recurrent inhibitory interactions are retained, but an additional shared inhibitory unit λ_L sends a common suppressive signal to all outputs. This shared inhibition enforces the sparsity constraint and yields soft-thresholding dynamics.

activity-update direction from Eq. 10 as

$$d_k(t,\tau) = \underbrace{\gamma \sum_{\ell=1}^{m} W_{k\ell}(t-1)x_\ell(t)}_{\text{feedforward input}} - \underbrace{\sum_{\substack{j=1 \\ j\neq k}}^{n} \frac{\hat{c}_{kj}(t)\bar{y}_j(t,\tau)}{(\hat{v}_k(t)+\varepsilon)(\hat{v}_j(t)+\varepsilon)}}_{\text{recurrent input}} \underbrace{-\gamma y_k(t,\tau) + \frac{\bar{y}_k(t,\tau)}{\hat{v}_k(t)+\varepsilon}}_{\text{leak}}. \quad (15)$$

Eq. 15 describes changes in the output signals, which correspond to changes in the output-layer activity in the network in Figure 1a. The first term is the feedforward input to output unit k through weights $W_{k\ell}$. The second term maps to lateral recurrent inhibition via connections with weights $\hat{c}_{kj}$. The remaining terms depend only on neuron k's own activity (but not on any other neurons in the network), so they can be interpreted as a leak or decay of activity (dependent on parameters $\hat{\mu}_k$ and $\hat{v}_k$ that are intrinsic to the neuron).

Assuming domains with limited total activity of output neurons ($\mathcal{B}_1$, $\mathcal{B}_{1,+}$, and Δ) requires normalization of activity that could be driven by an additional neural population λ_L, which acts as a shared inhibitory signal enforcing the corresponding population-level constraint (as shown in Figure 1b).

The parameter updates in Eqs. 12-14 are *local*. The feedforward rule of Eq. 12 depends only on the presynaptic activity and the postsynaptic prediction error, and corresponds to a previously proposed plasticity rule [36]. It has been postulated that biological neurons can implement such rule: If input $\boldsymbol{W}(t-1)\boldsymbol{x}(t)$ comes to dendritic compartments (denoted by grey rectangles in Figure 1a), then the neurons can compute errors $\boldsymbol{e}(t)$ as the differences between their somatic and dendritic activity, and trigger plasticity proportional to these errors [36].

The change in recurrent connections in Eq. 13 is proportional to the product of centered activity of pre and post-synaptic neurons; hence, it corresponds to Hebbian plasticity. This yields a more directly synapse-local interpretation of the recurrent updates compared to CorInfoMax [28] (where it relies on transformed inverse-covariance signals).

The variance trace $\hat{v}_i(t)$ determines the leak of neuron i and its change in Eq. 14 depends only on the centered activity of neuron i. The updates of the variance trace could correspond to homeostatic regulation [37], that seeks to maintain average neural activity in a desired range.

Unnormalized Predictive Entropy Maximization. A potential challenge for the neural implementation of Eq. 15 is the dependence of recurrent inhibition between neurons k and j on $\hat{v}_k$ and $\hat{v}_j$. We assumed above that these parameters determine neurons' leak, hence it is unclear how they could affect synaptic transmission specifically via recurrent connections. Therefore, in our subsequent experiments, we also consider a simplified and arguably even more biologically plausible variant of *PEM* called *unnormalized PEM* (*u-PEM*). We replace the variance normalization $((\hat{v}_i(t)+\varepsilon)(\hat{v}_j(t)+\varepsilon))^{-1}$ from the 'covariance reduction' term (Eq. 9) in the cost function with a parameter $\gamma_{lateral} > 0$ that controls the strength of the lateral coupling. Such approximation is justified, as the variance of outputs $\hat{v}_i(t)$ tends to approach a fixed value (Figure 14) corresponding to outputs spreading the available range in $\mathcal{P}$. Monitoring the variance in the *PEM* model provides a starting point for the range we should sweep to select the hyperparameter $\gamma_{lateral}$, further detailed in Appendix E.9. The resulting update direction is:

$$d_k(t,\tau) = \frac{\bar{y}_k(t,\tau)}{\hat{v}_k(t)+\varepsilon} - \gamma_{lateral} \sum_{\substack{j=1 \\ j\neq k}}^{n} \hat{c}_{kj}(t)\, \bar{y}_j(t,\tau) - \gamma \left(y_k(t,\tau) - \sum_{\ell=1}^{m} W_{k\ell}(t-1) x_\ell(t) \right).$$

u-PEM maps onto the same neural architecture as before, but now the lateral inhibition is directly implemented via the product of the weights and the neural activity, without needing further normalization. Further details are given in Appendix E.9.

4 Numerical results

We compare *PEM* and its unnormalized variant (*u-PEM*) with both batch and online baselines. The batch baselines are *CorInfoMax (Batch)* [20] and *ICA-InfoMax* [3]. Among them, *CorInfoMax (Batch)* provides a nonlocal, gradient-based batch reference: it optimizes the exact regularized correlative-information objective (Eq. A.2) through repeated projected updates of the batch output matrix (see Appendix A.2). The bio-plausible online baselines are *CorInfoMax (Online)* [28] and *Nonnegative Similarity Matching (Online)* [24]. Specifically, *CorInfoMax (Online)* is the most directly comparable method, as it targets the exact log-determinant objective that our Taylor surrogate approximates. Our aim is therefore to match the robustness of exact determinant-based correlated-source separation while obtaining a more directly local and biologically plausible implementation. Performance is measured by the mean component signal-to-noise ratio (mSNR) defined in Eq. E.1. Unless stated otherwise, all results are averaged over 30 independent realizations, and shaded bands denote 95% confidence intervals computed using Eq. E.2. Additional implementation details, extended results, and supplementary experiments, including comparisons with additional baselines, are reported in Appendix E.

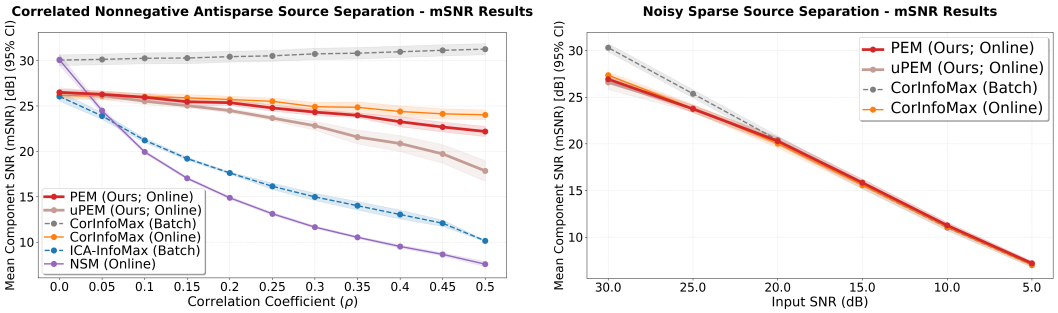
Synthetic source separation. We first consider synthetic linear mixtures with $n = 5$ sources, $m = 10$ mixtures, and $T = 10^5$ samples. For each realization, the mixing matrix $\boldsymbol{A} \in \mathbb{R}^{10\times 5}$ has i.i.d. standard normal entries, and the observations are generated as $\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{\epsilon}(t)$, $\boldsymbol{\epsilon}(t) \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_{10})$, $t = 1, \ldots, T$, with σ^2 chosen to match the prescribed input SNR level $\mathrm{SNR}_{\mathrm{in}}$.

We evaluate the method in two complementary regimes. In the first, we vary the source correlation level while keeping the observation noise fixed. In the second, we vary the observation noise while keeping the source-domain geometry fixed.

For the correlation experiment, we consider the nonnegative antisparse domain $\mathcal{B}_{\max,+}$. Sources are generated using a copula-t model, a standard way to generate correlated variables with uniform marginals; here ρ controls the dependence level and is varied over $\{0, 0.05, \ldots, 0.5\}$, while the mixture SNR is fixed at 30 dB. The results are shown in Figure 2a. Both *PEM* and its unnormalized variant (*u-PEM*) remain robust as ρ increases and clearly outperform *ICA-InfoMax* and *NSM*, whose performance degrades rapidly as source correlation grows. *PEM* appears more resistant to increased source correlation than *u-PEM*, suggesting that the variance normalization inherited from the entropy objective contributes to robustness under correlated sources.

Relative to *CorInfoMax*, the *PEM* models preserve robust correlated-source separation while replacing inverse-covariance dynamics by the simpler surrogate-based updates introduced in Section 3. As expected, the batch *CorInfoMax (Batch)* baseline achieves the strongest overall performance.

For the noise experiment, we consider the sparse domain $\mathcal{B}_1$. Sources are generated uniformly from $\mathcal{B}_1$, and the mixture SNR is varied over $\mathrm{SNR}_{\mathrm{in}} \in \{30, 25, \ldots, 5\}$ dB. Figure 2b shows that



(a) Correlated nonnegative antisparse ($\mathcal{B}_{\max,+}$)

(b) Noisy sparse ($\mathcal{B}_1$)

Figure 2: **Performance comparisons across source domains. (a)** Mean component SNR versus correlation ρ for nonnegative antisparse sources. *Predictive Entropy Maximization* (*PEM*) and its unnormalized variant (*u-PEM*) remain robust and outperform online baselines relying on stronger independence or decorrelation assumptions. **(b)** Mean component SNR versus input SNR for sparse sources. The *PEM* models stay close to the batch *CorInfoMax (Batch)* baseline across noise levels while remaining online and biologically plausible.

both *PEM* and *u-PEM* track the batch *CorInfoMax (Batch)* baseline closely across the full noise range. This indicates that replacing inverse-covariance dynamics by the Taylor surrogate does not materially degrade source separation performance, while still yielding an online and bio-plausible neural implementation.

Additional results for the antisparse, nonnegative sparse, and simplex domains follow the same pattern and are reported in Appendix Figure 10.

Learning sparse receptive fields. We evaluate the sparse architecture illustrated in Figure 1b on pre-whitened 12×12 natural image patches [4].[1] The network is trained on vectorized patches in $\mathbb{R}^{144}$. The learned filters, shown in Figure 3, are localized and oriented, with the familiar Gabor-like structure associated with sparse coding models of the early visual cortex [4]. This experiment shows that the same sparse neural mechanism used for source separation also captures statistically meaningful features from natural images. Appendix E.3 provides a quantitative comparison with Online CorInfoMax and Gaussian random filters using two-dimensional Gabor fits [38].

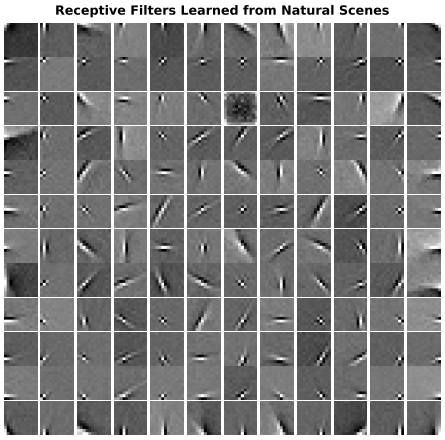
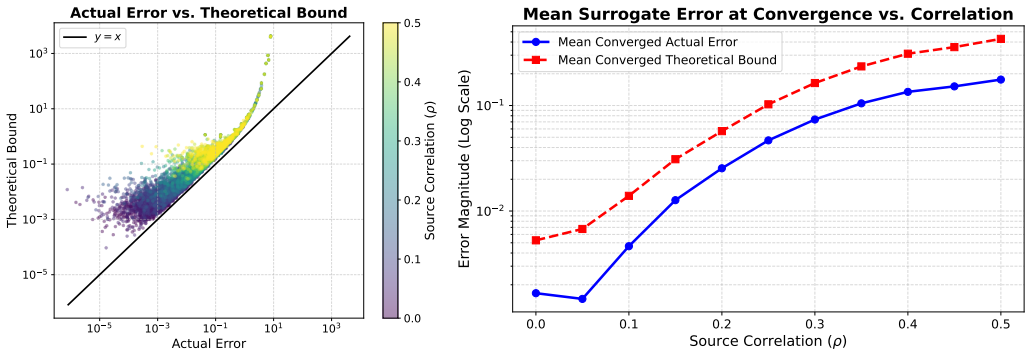


Figure 3: *Receptive fields* learned by the sparse *Predictive Entropy Maximization* model from natural image patches exhibit localized and oriented structure characteristic of sparse sensory representations.

Auditory source separation. We also evaluate the sparse architecture on an audio separation task inspired by the cocktail-party setting [6, 8]. We use three 5-second audio sources from the *librosa* library [39], namely `fishin`, `pistachio`, and `vibeace`, and transform the mixtures with a db4 discrete wavelet transform before running *PEM*. This choice is motivated by the approximate sparsity of natural sounds in multiscale representations [40, 5, 41]; the transform-domain formulation and representative temporal alignments are reported in Appendix E. Across 30 random mixing scenarios, the three recovered sources achieve source-wise SNR values of 24.12 ± 1.98 dB, 25.07 ± 2.14 dB, and 21.54 ± 2.17 dB (95% CI), indicating that the sparse-domain dynamics remain effective on realistic audio mixtures. Additionally, Figure 9 provides waveform and spectrogram comparisons between the reference sources and their PEM estimates for a representative trial.

Nonnegative image representations. We also evaluate the nonnegative antisparse network for unsupervised feature learning on Olivetti faces [42] and Fashion-MNIST [43]. PEM learns filters capturing facial structure on Olivetti faces (Figure 6) and representations useful for downstream

[1] The dataset and original Sparsenet implementation are available at `https://www.rctn.org/bruno/sparsenet/`.



(a) Actual error vs. theoretical bound

(b) Converged error vs. source correlation

Figure 4: **Taylor-surrogate diagnostics.** **(a)** Exact Taylor remainder versus the theoretical bound from Eq. D.14, over recorded runs and iterations, color-coded by source correlation ρ. The solid line denotes $y = x$. **(b)** Mean Taylor remainder and corresponding bound (Eq. D.14) as functions of ρ.

classification on Fashion-MNIST (Figure 7). Appendix E.2 reports these experiments, including reconstruction errors, visualizations, and comparisons with principal component analysis (PCA) and nonnegative matrix factorization (NMF).

Diagnostic validation of the Taylor surrogate. To complement the analytical results of Section 3, we empirically assess the accuracy of the second-order Taylor surrogate used to derive the neural dynamics. We consider the antisparse synthetic setting with controlled source correlation $\rho \in \{0, \dots, 0.5\}$, while fixing the input SNR at 30 dB.

To make the approximation nontrivial from the outset, we initialize the covariance statistic with substantial off-diagonal structure by setting $\hat{\boldsymbol{C}}^{\lambda}(0) = \boldsymbol{G}\boldsymbol{G}^{\top}$, $\boldsymbol{G} = \sqrt{0.2}\,\boldsymbol{I} + \boldsymbol{Z}$, $Z_{ij} \sim \mathcal{N}(0, 1/5)$. At each recorded iteration, we compute both the exact Taylor remainder and the sharper spectral upper bound given by the inequality in Eq. D.14, using the current covariance estimate $\hat{\boldsymbol{C}}^{\lambda}(t)$. Figure 4a reports all recorded error–bound pairs across runs and correlation levels. The empirical points lie on or above the identity line, confirming that the analytical bound consistently upper-bounds the observed approximation error. Figure 4b shows that, as expected, the approximation error increases with source correlation but remains small throughout the range considered and continues to be well controlled by the theoretical bound. Additional transient diagnostics, including time-evolution plots of the Taylor approximation error are reported in Appendix E.

5 Conclusion

Summary. In this work, we introduced *Predictive Entropy Maximization*, an online determinant-based entropy maximization framework for blind source separation over identifiable source domains. By replacing the exact log-determinant objective with a second-order Taylor surrogate, we obtained a predictive neural architecture in which feedforward plasticity is local and error-driven, while recurrent interactions are mediated by covariance traces and implement adaptive lateral inhibition. We also provided explicit spectral bounds for the surrogate error, clarifying when this approximation is accurate. Empirically, the resulting method remains robust under increasing source correlation and observation noise, compares favorably with existing biologically plausible baselines, and extends naturally to sparse receptive-field learning and auditory source separation. Taken together, these results provide a normative account of how structured source separation can give rise to simple local plasticity rules and interpretable inhibitory dynamics in recurrent networks.

Limitations and future directions. The performance of the proposed model can be sensitive to hyperparameter choices and to the number of observed mixtures, as analyzed empirically in Appendix E.7-E.8. In addition, similar to other biologically plausible recurrent BSS networks, the computational cost of our method scales with the number of fast neural-dynamics iterations (see Appendix E.11), motivating future work on faster inference schemes. General underdetermined source separation is outside the scope of our framework, but we provide exploratory ablations in Appendix E.8.3.

Acknowledgements

B.B. is supported by the Gatsby Charitable Foundation. This work was supported by the Wellcome Trust grant 313955/Z/24/Z and the Medical Research Council grant UKRI/MR/B000936/1. Part of this work was completed while B.B. was a visiting PhD student at the University of Oxford. B.B. thanks Vasco Portilheiro and Houssam Zenati for helpful discussions on an earlier version of this work.

Code availability

Our implementation code is available at https://github.com/BariscanBozkurt/Predictive-Entropy-Maximization.

References

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

[2] 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.

[3] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159, November 1995. ISSN 0899-7667. doi: 10.1162/neco.1995.7.6.1129. URL https://doi.org/10.1162/neco.1995.7.6.1129.

[4] 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. doi: 10.1038/381607a0.

[5] Michael S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363, 2002. doi: 10.1038/nn831. URL https://doi.org/10.1038/nn831.

[6] E Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the Acoustical Society of America*, 25(5):975–979, 1953.

[7] 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.

[8] Simon Haykin and Zhe Chen. The cocktail party problem. *Neural Computation*, 17(9):1875–1902, 2005. doi: 10.1162/0899766054322964.

[9] Aapo Hyvärinen and Erkki Oja. Independent component analysis: A tutorial, 1999. URL https://api.semanticscholar.org/CorpusID:118629.

[10] Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000. doi: 10.1016/S0893-6080(00)00026-5.

[11] 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, 06 2014. doi: 10.1109/TGRS.2015.2424719.

[12] 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. doi: 10.1137/17M114145X. URL https://doi.org/10.1137/17M114145X.

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

[14] 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, August 2013.

[15] 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. doi: https://doi.org/10.1002/env.3170050203. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/env.3170050203.

[16] 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 Processing Magazine*, 36(2):59–80, 2019. doi: 10.1109/MSP.2018.2877582.

[17] 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.

[18] Gokcan Tatli and Alper T. Erdogan. Generalized polytopic matrix factorization. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3235–3239, 2021. doi: 10.1109/ICASSP39728.2021.9413709.

[19] Bariscan Bozkurt and Alper T. Erdogan. On identifiable polytope characterization for polytopic matrix factorization. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3343–3347, 2022. doi: 10.1109/ICASSP43922.2022.9746370.

[20] 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, 2022. doi: 10.1109/ICASSP43922.2022.9746099.

[21] Takuya Isomura and Taro Toyoizumi. A local learning rule for independent component analysis. *Scientific Reports*, 6(1):28073, Jun 2016. ISSN 2045-2322. doi: 10.1038/srep28073. URL https://doi.org/10.1038/srep28073.

[22] Takuya Isomura and Taro Toyoizumi. Error-gated hebbian rule: A local learning rule for principal and independent component analysis. *Scientific Reports*, 8(1):1835, Jan 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-20082-0. URL https://doi.org/10.1038/s41598-018-20082-0.

[23] Yanis Bahroun, Dmitri Chklovskii, and Anirvan M. Sengupta. A normative and biologically plausible algorithm for independent component analysis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=fpvUKdqcPV.

[24] Cengiz Pehlevan, Sreyas Mohan, and Dmitri B. Chklovskii. Blind nonnegative source separation using biological neural networks. *Neural Computation*, 29(11):2925–2954, 11 2017. ISSN 0899-7667. doi: 10.1162/neco_a_01007. URL https://doi.org/10.1162/neco_a_01007.

[25] Alper Tunga Erdogan and Cengiz Pehlevan. Blind bounded source separation using neural networks with local learning rules. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3812–3816, 2020. URL https://api.semanticscholar.org/CorpusID:215745493.

[26] Berfin Simsek and Alper T. Erdogan. Online bounded component analysis: A simple recurrent neural network with local update rule for unsupervised separation of dependent and independent sources. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1639–1643, 2019. doi: 10.1109/IEEECONF44664.2019.9048916.

[27] Bariscan Bozkurt, Cengiz Pehlevan, and Alper Tunga Erdogan. Biologically-plausible determinant maximization neural networks for blind separation of correlated sources. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=espX_4CLr46.

[28] Bariscan Bozkurt, Ateş İsfendiyaroğlu, Cengiz Pehlevan, and Alper Tunga Erdogan. Correlative information maximization based biologically plausible neural networks for correlated source separation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8JsaP7j1cL0.

[29] Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017.

[30] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999. doi: 10.1038/4580.

[31] James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250, 2019.

[32] Bariscan Bozkurt, Cengiz Pehlevan, and Alper Tunga Erdogan. Correlative information maximization: A biologically plausible approach to supervised deep neural networks without weight symmetry. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=TUGoUNkccV.

[33] Manu Srinath Halvagal and Friedemann Zenke. The combination of hebbian and predictive plasticity learns invariant object representations in deep sensory networks. *Nature Neuroscience*, 26(11):1906–1915, Nov 2023. ISSN 1546-1726. doi: 10.1038/s41593-023-01460-y. URL https://doi.org/10.1038/s41593-023-01460-y.

[34] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

[35] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

[36] Robert Urbanczik and Walter Senn. Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528, 2014.

[37] Gina Turrigiano. Homeostatic synaptic plasticity: Local and global mechanisms for stabilizing neuronal function. *Cold Spring Harbor Perspectives in Biology*, 4(1):a005736, 2012. doi: 10.1101/cshperspect.a005736.

[38] Judson P. Jones and Larry A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58 6:1233–58, 1987. URL https://api.semanticscholar.org/CorpusID:16809045.

[39] Brian McFee, Matt McVicar, Daniel Faronbi, Iran Roman, Matan Gover, Stefan Balke, Scott Seyfarth, Ayoub Malek, Colin Raffel, Vincent Lostanlen, Benjamin van Niekirk, Dana Lee, Frank Cwitkowitz, Frank Zalkow, Oriol Nieto, Dan Ellis, Jack Mason, Kyungyun Lee, Bea Steers, Emily Halvachs, Carl Thomé, Fabian Robert-Stöter, Rachel Bittner, Ziyao Wei, Adam Weiss, Eric Battenberg, Keunwoo Choi, Ryuichi Yamamoto, CJ Carr, Alex Metsai, Stefan Sullivan, Pius Friesch, Asmitha Krishnakumar, Shunsuke Hidaka, Steve Kowalik, Fabian Keller, Dan Mazur, Alexandre Chabot-Leclerc, Curtis Hawthorne, Chandrashekhar Ramaprasad, Myungchul Keum, Juanita Gomez, Will Monroe, Viktor Andreevitch Morozov, Kian Eliasi, nullmightybofo, Paul Biberstein, N. Dorukhan Sergin, Romain Hennequin, Rimvydas Naktinis, beantowel, Taewoon Kim, Jon Petter Åsen, Joon Lim, Alex Malins, Darío Hereñú, Stef van der Struijk, Lorenz Nickel, Jackie Wu, Zhen Wang, Tim Gates, Matt Vollrath, Andy Sarroff, Xiao-Ming, Alastair Porter, Seth Kranzler, Voodoohop, Mattia Di Gangi, Helmi Jinoz, Connor Guerrero, Abduttayyeb Mazhar, toddrme2178, Zvi Baratz, Anton Kostin, Xinlu Zhuang, Cash TingHin Lo, Pavel Campr, Eric Semeniuc, Monsij Biswal, Shayenne Moura, Paul Brossier, Hojin Lee, Waldir Pimenta, Jon Petter Åsen, Shin Hyun, Iliya S, Eugene Rabinovich, Geo Lei, Jize Guo, Phillip S.M. Skelton, Matt Pitkin, Anmol Mishra, Slava Chaunin, BenedictSt, Scott VanRavenswaay, and David Südholt. librosa/librosa: 0.11.0, March 2025. URL https://doi.org/10.5281/zenodo.15006942.

[40] Evan C Smith and Michael S Lewicki. Efficient auditory coding. *Nature*, 439(7079):978–982, 2006.

[41] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992. ISBN 978-0-89871-274-2.

[42] F.S. Samaria and A.C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994. doi: 10.1109/ACV.1994.341300.

[43] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[44] Serdar Ozsoy, Shadi Hamdan, Sercan O Arik, Deniz Yuret, and Alper T Erdogan. Self-supervised learning with an information maximization criterion. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=5MgZAu2NR7X.

[45] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[46] Pau Bofill and Michael Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81(11):2353–2362, 2001. doi: 10.1016/S0165-1684(01)00120-7.

[47] Yuanqing Li, Shun-ichi Amari, Andrzej Cichocki, Daniel W. C. Ho, and Shengli Xie. Underdetermined blind source separation based on sparse representation. *IEEE Transactions on Signal Processing*, 54(2):423–437, 2006. doi: 10.1109/TSP.2005.861743.

[48] 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. doi: 10.1109/TNN.2005.849840.

[49] Özgür Yılmaz and Scott Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, 2004. doi: 10.1109/TSP.2004.828896.

[50] Thomas Melia and Scott Rickard. Underdetermined blind source separation in echoic environments using DESPRIT. *EURASIP Journal on Advances in Signal Processing*, 2007:86484, 2007. doi: 10.1155/2007/86484.

[51] Nobutaka Ito, Rintaro Ikeshita, Hiroshi Sawada, and Tomohiro Nakatani. A joint diagonalization based efficient approach to underdetermined blind audio source separation using the multichannel Wiener filter. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1950–1965, 2021. doi: 10.1109/TASLP.2021.3079815.

[52] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[53] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 12310–12320. PMLR, 2021. URL http://proceedings.mlr.press/v139/zbontar21a.html.

[54] Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

[55] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. URL https://arxiv.org/abs/2002.05709.

[56] Horace B. Barlow. Possible principles underlying the transformations of sensory messages. In Walter A. Rosenblith, editor, *Sensory Communication*, pages 217–234. MIT Press, Cambridge, MA, 1961.

[57] Katrina Drozdov, Ravid Shwartz-Ziv, and Yann LeCun. Video representation learning with joint-embedding predictive architectures, 2024. URL https://arxiv.org/abs/2412.10925.

Appendix

A Review of Correlative Information Maximization **17**

A.1 Correlative entropy and mutual information 17

A.2 Batch CorInfoMax objective 18

A.3 Online CorInfoMax objective and neural dynamics 18

A.4 Comparison with Predictive Entropy Maximization. 19

B Derivation of the update rules for the online optimization objective **20**

B.1 Recursive updates for the running mean and covariance 20

B.2 Derivation of the feedforward weight update 21

B.3 Gradient of the surrogate objective for the output dynamics 22

C Domain-specific network realizations **25**

C.1 Antisparse sources 25

C.2 Nonnegative antisparse sources 26

C.3 Sparse sources 26

C.4 Nonnegative sparse sources 27

C.5 Simplex sources 27

D Error analysis and analytical bounds for the second-order Taylor approximation **27**

D.1 Pointwise remainder representation and bounds 28

D.2 Batch surrogate optimality relative to the exact determinant objective 33

E Supplementary on numerical experiments **35**

E.1 Performance evaluation metric 35

E.2 Nonnegative feature learning with predictive entropy maximization 35

E.3 Quantitative analysis of sparse dictionary learning 37

E.4 Transform invariance in auditory source separation 38

E.5 Antisparse, nonnegative sparse, and simplex source separation examples 39

E.6 Transient diagnostics for the Taylor approximation 39

E.7 Simulation hyperparameters 40

E.8 Ablation Studies 42

E.9 Unnormalized Predictive Entropy Maximization 44

E.10 Variance-covariance regularisation in self-supervised learning 45

E.11 Computational complexity of the proposed method 45

E.12 Existing assets and licences 47

Overview of the appendices. Appendix A reviews CorInfoMax and explains the central distinction of PEM: replacing inverse-covariance states with variance and cross-covariance traces permits local recurrent plasticity. Appendix B derives the statistical, synaptic, and activity updates, and gives a sufficient condition under which the truncated activity direction decreases the full second-order surrogate. Appendix C shows how source-domain geometry determines the output nonlinearity—clipping, soft thresholding, or rectification—and any auxiliary inhibitory dynamics, while the slow learning rules remain shared across domains. Appendix D bounds the approximation error in terms of normalized cross-covariances and explains the choice of a second-order surrogate. Under uniform error control, global minimizers of the batch surrogate are nearly optimal for the exact regularized determinant objective. Appendix E provides supplementary information on numerical experiments as well as ablation studies.

A Review of Correlative Information Maximization

This section briefly reviews the correlative-information-maximization (CorInfoMax) framework underlying the batch method of Erdogan [20] and its online neural implementation in Bozkurt et al. [28]. Our purpose is to make explicit the information-theoretic quantities that motivate CorInfoMax, the exact batch and online objectives it optimizes, and how its recurrent learning rules differ from those derived in the present paper.

A.1 Correlative entropy and mutual information

Following Erdogan [20], we use the correlative entropy as a second-order entropy measure. For a finite set of vectors $\boldsymbol{X} = [\boldsymbol{x}(1), \dots, \boldsymbol{x}(T)] \in \mathbb{R}^{m\times T}$ with sample autocorrelation matrix

$$\hat{\boldsymbol{R}}_{\boldsymbol{x}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}(t)\boldsymbol{x}(t)^\top,$$

the corresponding deterministic correlative entropy is (see Erdogan [20][Section 4])

$$\hat{\mathcal{H}}_{\mathrm{CE}}^{(\varepsilon)}(\boldsymbol{X}) := \frac{1}{2}\log\det\big(\hat{\boldsymbol{R}}_{\boldsymbol{x}} + \varepsilon \boldsymbol{I}\big) + \frac{m}{2}\log(2\pi e),$$

where $\varepsilon > 0$ is a small regularization constant. This quantity was introduced in Erdogan [20] under the name *log-determinant entropy*. In later work, the terminology *correlative entropy* has been adopted for the same second-order entropy measure [28, 32, 44], and we follow that convention here. No zero-mean assumption is made on $\boldsymbol{x}$ or $\boldsymbol{y}$; the autocorrelation matrices used here are uncentered second moments. Such uncentered statistics are also used in the determinant-maximization formulation of polytopic matrix factorization [17]. Here, we review the online CorInfoMax formulation [28], which avoids explicit mean estimation and centering. CorInfoMax also admits a formulation using centered covariances [20]. We use centered covariances in PEM so that the off-diagonal terms entering our Taylor approximation exclude products of component means, which can be nonzero even for uncorrelated sources.

For a pair of finite sets $\boldsymbol{X} = [\boldsymbol{x}(1), \dots, \boldsymbol{x}(T)] \in \mathbb{R}^{m\times T}$ and $\boldsymbol{Y} = [\boldsymbol{y}(1), \dots, \boldsymbol{y}(T)] \in \mathbb{R}^{n\times T}$, define

$$\hat{\boldsymbol{R}}_{\boldsymbol{xy}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}(t)\boldsymbol{y}(t)^\top, \qquad \hat{\boldsymbol{R}}_{\boldsymbol{yx}} = \hat{\boldsymbol{R}}_{\boldsymbol{xy}}^\top.$$

The deterministic joint correlative entropy is (see Erdogan [20][Section 4])

$$\hat{\mathcal{H}}_{\mathrm{CE}}^{(\varepsilon)}(\boldsymbol{X},\boldsymbol{Y}) := \frac{1}{2}\log\det\left(\begin{bmatrix} \hat{\boldsymbol{R}}_{\boldsymbol{x}} + \varepsilon \boldsymbol{I} & \hat{\boldsymbol{R}}_{\boldsymbol{xy}} \\ \hat{\boldsymbol{R}}_{\boldsymbol{yx}} & \hat{\boldsymbol{R}}_{\boldsymbol{y}} + \varepsilon \boldsymbol{I} \end{bmatrix}\right) + \frac{m+n}{2}\log(2\pi e).$$

Using the block determinant identity, this can be written as

$$\hat{\mathcal{H}}_{\mathrm{CE}}^{(\varepsilon)}(\boldsymbol{X},\boldsymbol{Y}) = \hat{\mathcal{H}}_{\mathrm{CE}}^{(\varepsilon)}(\boldsymbol{X}) + \hat{\mathcal{H}}_{\mathrm{CE}}^{(\varepsilon)}(\boldsymbol{Y} \mid_L \boldsymbol{X}),$$

where

$$\hat{\mathcal{H}}_{\mathrm{CE}}^{(\varepsilon)}(\boldsymbol{Y} \mid_L \boldsymbol{X}) := \frac{1}{2}\log\det\big(\hat{\boldsymbol{R}}_{\boldsymbol{e}} + \varepsilon \boldsymbol{I}\big) + \frac{n}{2}\log(2\pi e),$$

and

$$\hat{\boldsymbol{R}}_{\boldsymbol{e}} := \hat{\boldsymbol{R}}_{\boldsymbol{y}} - \hat{\boldsymbol{R}}_{\boldsymbol{yx}}\big(\hat{\boldsymbol{R}}_{\boldsymbol{x}} + \varepsilon \boldsymbol{I}\big)^{-1}\hat{\boldsymbol{R}}_{\boldsymbol{xy}}. \tag{A.1}$$

The notation $\boldsymbol{Y} \mid_L \boldsymbol{X}$ emphasizes that this is the correlative entropy of the residual under the *best linear* predictor of $\boldsymbol{Y}$ from $\boldsymbol{X}$, not Shannon conditional entropy.

The corresponding deterministic *correlative mutual information* is defined by

$$\hat{\mathcal{I}}^{(\varepsilon)}_{\mathrm{CMI}}(\boldsymbol{X}, \boldsymbol{Y}) := \hat{\mathcal{H}}^{(\varepsilon)}_{\mathrm{CE}}(\boldsymbol{Y}) - \hat{\mathcal{H}}^{(\varepsilon)}_{\mathrm{CE}}(\boldsymbol{Y} \mid_L \boldsymbol{X}).$$

For sufficiently small ε, $\hat{\mathcal{H}}^{(\varepsilon)}_{\mathrm{CE}}(\boldsymbol{Y} \mid_L \boldsymbol{X})$ corresponds to the correlative entropy of the error of the best linear minimum-mean-square predictor of $\boldsymbol{y}$ from $\boldsymbol{x}$. Thus, the associated mutual information measures linear/correlative dependence rather than general statistical dependence.

A.2 Batch CorInfoMax objective

Let $\boldsymbol{X} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(T)] \in \mathbb{R}^{m\times T}$ denote the matrix of mixed inputs and $\boldsymbol{Y} = [\boldsymbol{y}(1), \ldots, \boldsymbol{y}(T)] \in \mathbb{R}^{n\times T}$ the separated outputs, constrained by $\boldsymbol{y}(t) \in \mathcal{P}$ for all t. The batch CorInfoMax objective is the maximization of the deterministic correlative information:

$$\begin{aligned} \underset{\boldsymbol{Y}\in\mathbb{R}^{n\times T}}{\text{maximize}} \quad & \hat{\mathcal{I}}^{(\varepsilon)}_{\mathrm{CMI}}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2}\log\det\big(\hat{\boldsymbol{R}}_{\boldsymbol{y}} + \varepsilon \boldsymbol{I}\big) - \frac{1}{2}\log\det\big(\hat{\boldsymbol{R}}_{\boldsymbol{e}} + \varepsilon \boldsymbol{I}\big), \\ \text{subject to} \quad & \boldsymbol{y}(t) \in \mathcal{P}, \qquad t = 1, \ldots, T. \end{aligned} \tag{A.2}$$

Here $\hat{\boldsymbol{R}}_{\boldsymbol{e}}$ is defined by Equation A.1. For sufficiently small ε, it approximates the sample correlation matrix of the error of the best linear minimum-mean-square predictor of $\boldsymbol{y}$ from $\boldsymbol{x}$. Alternatively, a centered formulation replaces the sample auto- and cross-correlations with their covariance counterparts. Under the same identifiable-domain and sufficient-scattering assumptions discussed in Section 2, global optima of Equation A.2 recover the sources up to ideal separation.

For the purposes of the present paper, the key point is that batch CorInfoMax optimizes the *exact* regularized log-determinant objective. Following Erdogan [20], the objective in Eq. A.2 can be optimized by projected gradient ascent:

$$\boldsymbol{Y}^{(k+1)} = \sigma_{\mathcal{P}}\left(\boldsymbol{Y}^{(k)} + \mu^{(k)}\nabla_{\boldsymbol{Y}}\hat{\mathcal{I}}^{(\varepsilon)}_{\mathrm{CMI}}\big(\boldsymbol{X}, \boldsymbol{Y}^{(k)}\big)\right). \tag{A.3}$$

Here, k indexes optimization iterations, $\mu^{(k)} > 0$ is the step size, and $\sigma_{\mathcal{P}}$ projects each column onto the source domain $\mathcal{P}$. In our implementation, projected output updates alternate with batch updates of the linear predictor $\boldsymbol{W}$. The ascent direction uses output and prediction-error second-order statistics computed across the batch and involves linear solves with the corresponding regularized covariance or correlation matrices (see Section 6 in [20]). These computations are not restricted to information available at individual synapses. Batch CorInfoMax therefore provides a nonlocal, gradient-based comparison for the local learning rules proposed here. The inverse second-order matrices also become central dynamical variables in the online formulation reviewed next.

A.3 Online CorInfoMax objective and neural dynamics

The online CorInfoMax construction replaces the batch correlation matrices by exponentially weighted versions [28]. Using notation parallel to the main text, define

$$\hat{\boldsymbol{R}}^{\lambda_y}_{\boldsymbol{y}}(t) := \frac{1}{\eta_y(t)}\sum_{t'=1}^{t}\lambda_y^{t-t'}\boldsymbol{y}(t')\boldsymbol{y}(t')^{\top}, \tag{A.4}$$

$$\hat{\boldsymbol{R}}^{\lambda_e}_{\boldsymbol{e}}(t) := \frac{1}{\eta_e(t)}\sum_{t'=1}^{t}\lambda_e^{t-t'}\boldsymbol{e}(t')\boldsymbol{e}(t')^{\top}, \tag{A.5}$$

with forgetting factors $\lambda_y, \lambda_e \in (0,1)$ and normalization factors

$$\eta_y(t) = \sum_{t'=1}^{t}\lambda_y^{t-t'}, \qquad \eta_e(t) = \sum_{t'=1}^{t}\lambda_e^{t-t'}.$$

The error vector is

$$\boldsymbol{e}(t) = \boldsymbol{y}(t) - \boldsymbol{W}(t)\boldsymbol{x}(t).$$

The corresponding online CorInfoMax problem at time t is

$$\begin{aligned} \underset{\boldsymbol{y}(t)\in\mathbb{R}^n}{\text{maximize}} \quad & \mathcal{J}_t^{\mathrm{CI}}(\boldsymbol{y}(t)) := \frac{1}{2}\log\det\big(\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{\lambda_y}(t) + \varepsilon\boldsymbol{I}\big) - \frac{1}{2}\log\det\big(\hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\lambda_e}(t) + \varepsilon\boldsymbol{I}\big), \\ \text{subject to} \quad & \boldsymbol{y}(t) \in \mathcal{P}. \end{aligned} \tag{A.6}$$

In the online formulation, the best linear predictor $\boldsymbol{W}(t)$ is treated as an adaptive parameter and updated through an online regularized least-squares criterion, which yields the usual LMS-type rule. Define the inverse-correlation states

$$\boldsymbol{B}_{\boldsymbol{y}}(t) := \big(\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{\lambda_y}(t) + \varepsilon\boldsymbol{I}\big)^{-1}, \qquad \boldsymbol{B}_{\boldsymbol{e}}(t) := \big(\hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\lambda_e}(t) + \varepsilon\boldsymbol{I}\big)^{-1}.$$

Gradient ascent on Equation A.6 yields

$$\nabla_{\boldsymbol{y}(t)}\mathcal{J}_t^{\mathrm{CI}} = \gamma_y(t)\,\boldsymbol{B}_{\boldsymbol{y}}(t-1)\boldsymbol{y}(t) - \gamma_e(t)\,\boldsymbol{B}_{\boldsymbol{e}}(t-1)\boldsymbol{e}(t),$$

where

$$\gamma_y(t) := \Big(\lambda_y\eta_y(t-1) + \boldsymbol{y}(t)^\top\boldsymbol{B}_{\boldsymbol{y}}(t-1)\boldsymbol{y}(t)\Big)^{-1}, \qquad \gamma_e(t) := \Big(\lambda_e\eta_e(t-1) + \boldsymbol{e}(t)^\top\boldsymbol{B}_{\boldsymbol{e}}(t-1)\boldsymbol{e}(t)\Big)^{-1}.$$

Depending on the domain $\mathcal{P}$, CorInfoMax then combines this gradient step with either a projected ascent update (for box-type domains) or a primal-dual/proximal step (for sparse and simplex domains).

The feedforward predictor is updated by the local LMS rule

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \mu_W(t)\boldsymbol{e}(t)\boldsymbol{x}(t)^\top.$$

By contrast, the lateral learning rules are based on inverse-correlation states. Applying the Matrix Inversion Lemma to Equations A.4 and A.5 yields the exact recursions

$$\boldsymbol{B}_{\boldsymbol{y}}(t+1) = \frac{\eta_y(t)}{\lambda_y\eta_y(t-1)}\left(\boldsymbol{B}_{\boldsymbol{y}}(t) - \gamma_y(t)\,\boldsymbol{B}_{\boldsymbol{y}}(t)\boldsymbol{y}(t)\boldsymbol{y}(t)^\top\boldsymbol{B}_{\boldsymbol{y}}(t)\right), \tag{A.7}$$

$$\boldsymbol{B}_{\boldsymbol{e}}(t+1) = \frac{\eta_e(t)}{\lambda_e\eta_e(t-1)}\left(\boldsymbol{B}_{\boldsymbol{e}}(t) - \gamma_e(t)\,\boldsymbol{B}_{\boldsymbol{e}}(t)\boldsymbol{e}(t)\boldsymbol{e}(t)^\top\boldsymbol{B}_{\boldsymbol{e}}(t)\right). \tag{A.8}$$

In Bozkurt et al. [28], these exact inverse-state updates are then simplified to obtain a biologically plausible implementation. The inverse error-correlation update in Equation A.8 is not retained as a learned state; instead, the approximation

$$\hat{\boldsymbol{R}}_{\boldsymbol{e}}^{\lambda_e}(t) + \varepsilon\boldsymbol{I} \approx \varepsilon\boldsymbol{I}, \qquad \text{hence} \qquad \boldsymbol{B}_{\boldsymbol{e}}(t) \approx \varepsilon^{-1}\boldsymbol{I}, \tag{A.9}$$

is invoked, motivated by the expectation that the prediction error becomes small in the noiseless linear setting. In addition, for λ_y close to 1 and t sufficiently large, the scalar factor $\gamma_y(t)$ is approximated by

$$\gamma_y(t) \approx \frac{1-\lambda_y}{\lambda_y}. \tag{A.10}$$

Substituting Equation A.10 into Equation A.7 yields the simplified lateral update used in the CorInfoMax neural networks:

$$\boldsymbol{B}_{\boldsymbol{y}}(t+1) = \lambda_y^{-1}\left(\boldsymbol{B}_{\boldsymbol{y}}(t) - \frac{1-\lambda_y}{\lambda_y}\,\boldsymbol{B}_{\boldsymbol{y}}(t)\boldsymbol{y}(t)\boldsymbol{y}(t)^\top\boldsymbol{B}_{\boldsymbol{y}}(t)\right). \tag{A.11}$$

A.4 Comparison with Predictive Entropy Maximization.

The distinction from the present paper can now be stated precisely. In CorInfoMax, the exact regularized log-determinant objective produces gradients involving inverse second-order states, so recurrent plasticity is expressed through $\boldsymbol{B}_{\boldsymbol{y}}(t)$ rather than through covariance traces. Even after the approximations in Equations A.9 and A.10, the lateral update in Equation A.11 remains a rank-one update built from transformed population activities.

Indeed, if we define $\boldsymbol{z}(t) := \boldsymbol{B}_{\boldsymbol{y}}(t)\boldsymbol{y}(t)$, then Equation A.11 becomes

$$\boldsymbol{B}_{\boldsymbol{y}}(t+1) = \lambda_y^{-1}\left(\boldsymbol{B}_{\boldsymbol{y}}(t) - \frac{1-\lambda_y}{\lambda_y}\,\boldsymbol{z}(t)\boldsymbol{z}(t)^\top\right).$$

Hence the update of an individual lateral coefficient depends on the transformed signal $\boldsymbol{z}(t)$, not directly on the co-activity of the neuron pair it couples. In the 3×3 case, if

$$\boldsymbol{B}_{\boldsymbol{y}}(t) = \begin{bmatrix} b_{11}(t) & b_{12}(t) & b_{13}(t) \\ b_{21}(t) & b_{22}(t) & b_{23}(t) \\ b_{31}(t) & b_{32}(t) & b_{33}(t) \end{bmatrix}, \qquad \boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix},$$

then

$$z_1(t) = b_{11}(t)y_1(t)+b_{12}(t)y_2(t)+b_{13}(t)y_3(t), \qquad z_2(t) = b_{21}(t)y_1(t)+b_{22}(t)y_2(t)+b_{23}(t)y_3(t),$$

and Equation A.11 gives

$$b_{21}(t+1) = \lambda_y^{-1}\left(b_{21}(t) - \frac{1-\lambda_y}{\lambda_y}\,z_2(t)z_1(t)\right).$$

Thus, the update of $b_{21}(t)$ depends not only on the pair $(y_2(t), y_1(t))$, but also on $y_3(t)$. In general, a single recurrent update depends on the activity of the full output population. This violates the local plasticity condition of biological plausibility, as defined in the introduction.

By contrast, Predictive Entropy Maximization introduces a second-order Taylor surrogate of the regularized log-determinant and works directly with covariance traces rather than inverse-covariance states. The corresponding recurrent statistics are the exponentially weighted variance and cross-covariance traces of the centered outputs, defined by $\hat{v}_i(t) := [\hat{\boldsymbol{C}}^\lambda(t)]_{ii}$ and $\hat{c}_{ij}(t) := [\hat{\boldsymbol{C}}^\lambda(t)]_{ij}$ for $i \neq j$. Their recursions are

$$\hat{v}_i(t) = \lambda\hat{v}_i(t-1) + (1-\lambda)\bar{y}_i(t)^2, \qquad \hat{c}_{ij}(t) = \lambda\hat{c}_{ij}(t-1) + (1-\lambda)\bar{y}_i(t)\bar{y}_j(t),$$

so the cross-covariance update is written directly in terms of centered pairwise co-activity, while the variance update depends only on the squared centered activity of the corresponding neuron. This is the key distinction emphasized throughout the main text: the surrogate formulation removes the need to maintain inverse-covariance states and yields a more directly local interpretation of recurrent plasticity.

B Derivation of the update rules for the online optimization objective

In this section, we derive the recursive updates for the online second-order statistics and the feedforward weights, as well as the output gradient used in the fast neural dynamics.

B.1 Recursive updates for the running mean and covariance

Recall from Equation 4 that the exponentially weighted output covariance is defined by

$$\hat{\boldsymbol{C}}^\lambda(t) = \frac{1}{\eta(t)}\sum_{t'=1}^{t}\lambda^{\,t-t'}\big(\boldsymbol{y}(t') - \hat{\boldsymbol{\mu}}^\lambda(t')\big)\big(\boldsymbol{y}(t') - \hat{\boldsymbol{\mu}}^\lambda(t')\big)^\top,$$

where

$$\eta(t) = \sum_{t'=1}^{t}\lambda^{\,t-t'} = \frac{1-\lambda^t}{1-\lambda}.$$

Since $\eta(t) = \lambda\eta(t-1) + 1$, we define $\alpha(t) := \frac{1}{\eta(t)}$.

Using this identity, we isolate the most recent sample in the covariance sum and obtain

$$\hat{\boldsymbol{C}}^\lambda(t) = \frac{1}{\eta(t)}\left[\lambda\sum_{t'=1}^{t-1}\lambda^{\,t-1-t'}\big(\boldsymbol{y}(t') - \hat{\boldsymbol{\mu}}^\lambda(t')\big)\big(\boldsymbol{y}(t') - \hat{\boldsymbol{\mu}}^\lambda(t')\big)^\top\right]$$

$$
\begin{aligned}
&+ \frac{1}{\eta(t)}\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)^{\top} \\
&= \frac{\lambda\eta(t-1)}{\eta(t)}\hat{\boldsymbol{C}}^{\lambda}(t-1) + \frac{1}{\eta(t)}\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)^{\top} \\
&= \big(1-\alpha(t)\big)\hat{\boldsymbol{C}}^{\lambda}(t-1) + \alpha(t)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)^{\top}. \qquad \text{(B.1)}
\end{aligned}
$$

Exactly the same argument applied to the exponentially weighted mean

$$
\hat{\boldsymbol{\mu}}^{\lambda}(t) = \frac{1}{\eta(t)}\sum_{t'=1}^{t}\lambda^{\,t-t'}\boldsymbol{y}(t')
$$

yields the recursion

$$
\hat{\boldsymbol{\mu}}^{\lambda}(t) = \big(1-\alpha(t)\big)\hat{\boldsymbol{\mu}}^{\lambda}(t-1) + \alpha(t)\boldsymbol{y}(t). \qquad \text{(B.2)}
$$

For large t, the normalization converges to $\eta(t) \approx (1-\lambda)^{-1}$, hence $\alpha(t) \approx 1-\lambda$. Under this steady-state approximation, Equations B.1 and B.2 reduce to

$$
\hat{\boldsymbol{\mu}}^{\lambda}(t) = \lambda\hat{\boldsymbol{\mu}}^{\lambda}(t-1) + (1-\lambda)\boldsymbol{y}(t), \qquad \text{(B.3)}
$$

$$
\hat{\boldsymbol{C}}^{\lambda}(t) = \lambda\hat{\boldsymbol{C}}^{\lambda}(t-1) + (1-\lambda)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)\big)^{\top}. \qquad \text{(B.4)}
$$

For convenience, define the centered activity

$$
\bar{\boldsymbol{y}}(t) := \boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t).
$$

Then Equation B.4 can be written as

$$
\hat{\boldsymbol{C}}^{\lambda}(t) = \lambda\hat{\boldsymbol{C}}^{\lambda}(t-1) + (1-\lambda)\bar{\boldsymbol{y}}(t)\bar{\boldsymbol{y}}(t)^{\top}.
$$

We now decompose the covariance matrix as

$$
\hat{\boldsymbol{C}}^{\lambda}(t) = \hat{\boldsymbol{D}}^{\lambda}(t) + \hat{\boldsymbol{O}}^{\lambda}(t), \qquad \text{(B.5)}
$$

where $\hat{\boldsymbol{D}}^{\lambda}(t)$ is diagonal and $\hat{\boldsymbol{O}}^{\lambda}(t)$ has zero diagonal. Reading off the diagonal and off-diagonal parts of Equation B.4 yields

$$
\hat{v}_i(t) = \lambda\hat{v}_i(t-1) + (1-\lambda)\bar{y}_i(t)^2, \qquad i = 1,\dots,n, \qquad \text{(B.6)}
$$

$$
\hat{c}_{ij}(t) = \lambda\hat{c}_{ij}(t-1) + (1-\lambda)\bar{y}_i(t)\bar{y}_j(t), \qquad i \neq j. \qquad \text{(B.7)}
$$

B.2 Derivation of the feedforward weight update

Let $\boldsymbol{e}(t) := \boldsymbol{y}(t) - \boldsymbol{W}\boldsymbol{x}(t)$ denote the instantaneous prediction error. We define the instantaneous quadratic cost

$$
\mathcal{J}_t(\boldsymbol{W}) \propto \frac{1}{2}\|\boldsymbol{e}(t)\|_2^2 = \frac{1}{2}\|\boldsymbol{y}(t) - \boldsymbol{W}\boldsymbol{x}(t)\|_2^2.
$$

Differentiating with respect to $\boldsymbol{W}$ gives

$$
\begin{aligned}
\nabla_{\boldsymbol{W}}\mathcal{J}_t(\boldsymbol{W}) &= \frac{\partial}{\partial \boldsymbol{W}}\left[\frac{1}{2}(\boldsymbol{y}(t) - \boldsymbol{W}\boldsymbol{x}(t))^{\top}(\boldsymbol{y}(t) - \boldsymbol{W}\boldsymbol{x}(t))\right] \\
&= -(\boldsymbol{y}(t) - \boldsymbol{W}\boldsymbol{x}(t))\boldsymbol{x}(t)^{\top} \\
&= -\boldsymbol{e}(t)\boldsymbol{x}(t)^{\top}.
\end{aligned}
$$

A gradient step with learning rate $\alpha_W(t)$ therefore yields

$$
\begin{aligned}
\boldsymbol{W}(t) &= \boldsymbol{W}(t-1) - \alpha_W(t)\nabla_{\boldsymbol{W}}\mathcal{J}_t(\boldsymbol{W}(t-1)) \\
&= \boldsymbol{W}(t-1) + \alpha_W(t)\boldsymbol{e}(t)\boldsymbol{x}(t)^{\top}. \qquad \text{(B.8)}
\end{aligned}
$$

This formulation yields a biologically plausible, local error-modulated Hebbian learning rule where the synaptic update is proportional to the product of the presynaptic activity $\boldsymbol{x}(t)$ and the local postsynaptic error signal $\boldsymbol{e}(t)$.

B.3 Gradient of the surrogate objective for the output dynamics

We now derive the gradient of the fast-time-scale objective with respect to the current output $\boldsymbol{y}(t)$. Recall from Equation 9 that

$$\mathcal{J}_t(\boldsymbol{y}) = -\sum_{i=1}^{n} \log\big(\hat{v}_i(t) + \varepsilon\big) + \frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\hat{c}_{ij}(t)^2}{\big(\hat{v}_i(t) + \varepsilon\big)\big(\hat{v}_j(t) + \varepsilon\big)} + \gamma\|\boldsymbol{y}(t) - \boldsymbol{W}(t-1)\boldsymbol{x}(t)\|_2^2.$$

At time t, the current statistics $\hat{\boldsymbol{\mu}}^{\lambda}(t)$, $\hat{\boldsymbol{D}}^{\lambda}(t)$, and $\hat{\boldsymbol{O}}^{\lambda}(t)$ depend on $\boldsymbol{y}(t)$ through the steady-state updates in Equations B.3, B.6, and B.7. We first compute the necessary derivatives.

Derivative of the centered activity. For each component,

$$\hat{\mu}_k(t) = \lambda\hat{\mu}_k(t-1) + (1-\lambda)y_k(t).$$

Hence

$$\bar{y}_k(t) = y_k(t) - \hat{\mu}_k(t) = \lambda\big(y_k(t) - \hat{\mu}_k(t-1)\big),$$

and therefore

$$\frac{\partial \bar{y}_i(t)}{\partial y_k(t)} = \lambda\,\delta_{ik}, \tag{B.9}$$

where δ_{ik} is the Kronecker delta.

Derivatives of the diagonal and off-diagonal statistics. Using Equation B.6 together with Equation B.9, we obtain

$$\frac{\partial \hat{v}_i(t)}{\partial y_k(t)} = 2\lambda(1-\lambda)\bar{y}_i(t)\,\delta_{ik}. \tag{B.10}$$

Similarly, from Equation B.7,

$$\frac{\partial \hat{c}_{ij}(t)}{\partial y_k(t)} = \lambda(1-\lambda)\big(\delta_{ik}\bar{y}_j(t) + \delta_{jk}\bar{y}_i(t)\big), \qquad i \neq j. \tag{B.11}$$

Derivative of the variance term. Using Equation B.10,

$$\begin{aligned}\frac{\partial}{\partial y_k(t)}\left(-\sum_{i=1}^{n}\log(\hat{v}_i(t)+\varepsilon)\right) &= -\sum_{i=1}^{n}\frac{1}{\hat{v}_i(t)+\varepsilon}\frac{\partial \hat{v}_i(t)}{\partial y_k(t)} \\ &= -\frac{2\lambda(1-\lambda)\bar{y}_k(t)}{\hat{v}_k(t)+\varepsilon}.\end{aligned} \tag{B.12}$$

Derivative of the cross-covariance penalty. Define

$$\mathcal{R}_t := \frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1 \\ j\neq i}}^{n}\frac{\hat{c}_{ij}(t)^2}{\big(\hat{v}_i(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)}.$$

Only terms with $i = k$ or $j = k$ contribute to the derivative with respect to $y_k(t)$. Using the symmetry of $\hat{\boldsymbol{O}}^{\lambda}(t)$, together with Equations B.10 and B.11, a direct calculation gives

$$\begin{aligned}\frac{\partial \mathcal{R}_t}{\partial y_k(t)} &= 2\lambda(1-\lambda)\sum_{\substack{j=1 \\ j\neq k}}^{n}\frac{\hat{c}_{kj}(t)\bar{y}_j(t)}{\big(\hat{v}_k(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)} \\ &\quad - 2\lambda(1-\lambda)\sum_{\substack{j=1 \\ j\neq k}}^{n}\frac{\hat{c}_{kj}(t)^2\bar{y}_k(t)}{\big(\hat{v}_k(t)+\varepsilon\big)^2\big(\hat{v}_j(t)+\varepsilon\big)}.\end{aligned} \tag{B.13}$$

Derivative of the prediction term. Let $e_k(t) := [\boldsymbol{y}(t) - \boldsymbol{W}(t-1)\boldsymbol{x}(t)]_k$. Then

$$\frac{\partial}{\partial y_k(t)}\gamma\|\boldsymbol{y}(t) - \boldsymbol{W}(t-1)\boldsymbol{x}(t)\|_2^2 = 2\gamma e_k(t). \tag{B.14}$$

Full gradient. Combining Equations B.12, B.13, and B.14, we obtain

$$\frac{\partial \mathcal{J}_t(\boldsymbol{y})}{\partial y_k(t)} = 2\lambda(1-\lambda)\left[-\frac{\bar{y}_k(t)}{\hat{v}_k(t)+\varepsilon} + \sum_{\substack{j=1\\ j\neq k}}^{n} \frac{\hat{c}_{kj}(t)\bar{y}_j(t)}{\big(\hat{v}_k(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)}\right] - 2\lambda(1-\lambda)\sum_{\substack{j=1\\ j\neq k}}^{n} \frac{\hat{c}_{kj}(t)^2\bar{y}_k(t)}{\big(\hat{v}_k(t)+\varepsilon\big)^2\big(\hat{v}_j(t)+\varepsilon\big)} + 2\gamma e_k(t). \tag{B.15}$$

The term proportional to $\hat{c}_{kj}(t)^2$ in Equation B.15 is quadratic in the off-diagonal covariance entries. Since the surrogate objective is used precisely in the regime where the normalized cross-covariances are small, this term is of higher order than the leading recurrent interaction retained in the main text. Discarding it yields the approximation

$$\frac{\partial \mathcal{J}_t(\boldsymbol{y})}{\partial y_k(t)} \approx 2\lambda(1-\lambda)\left[-\frac{\bar{y}_k(t)}{\hat{v}_k(t)+\varepsilon} + \sum_{\substack{j=1\\ j\neq k}}^{n} \frac{\hat{c}_{kj}(t)\bar{y}_j(t)}{\big(\hat{v}_k(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)}\right] + 2\gamma e_k(t).$$

Absorbing the constant prefactor $\lambda(1-\lambda)$ into γ, we recover the simplified gradient form used in the main text:

$$\frac{\partial \mathcal{J}_t(\boldsymbol{y})}{\partial y_k(t)} \propto \left[-\frac{\bar{y}_k(t)}{\hat{v}_k(t)+\varepsilon} + \sum_{\substack{j=1\\ j\neq k}}^{n} \frac{\hat{c}_{kj}(t)\bar{y}_j(t)}{\big(\hat{v}_k(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)}\right] + \gamma e_k(t).$$

The negative of this expression is the descent direction used for the fast neural dynamics in Section 3.

Descent property of the truncated direction. The fast neural dynamics omit the term proportional to $\hat{c}_{kj}(t)^2\bar{y}_k(t)$ in Equation B.15, which arises from differentiating the variance-dependent denominators of the cross-covariance penalty. Because this additional approximation discards part of the gradient, a decrease in the full second-order surrogate objective $\mathcal{J}_t$ is no longer automatic. We therefore derive a sufficient condition under which the activity-update direction $\boldsymbol{d}$ from Equation 10 remains a descent direction.

Fix a sample t and a fast neural-dynamics iteration τ. All quantities below are evaluated at the current output $\boldsymbol{y}(t,\tau)$, and we suppress τ in $\boldsymbol{y}$, $\bar{\boldsymbol{y}}$, $\boldsymbol{e}$, and $\boldsymbol{d}$. Thus $\boldsymbol{d}(t) := \boldsymbol{d}(t,\tau)$ is the same direction as in Equation 10, with components

$$d_k(t) := \frac{\bar{y}_k(t)}{\hat{v}_k(t)+\varepsilon} - \sum_{\substack{j=1\\ j\neq k}}^{n} \frac{\hat{c}_{kj}(t)\bar{y}_j(t)}{\big(\hat{v}_k(t)+\varepsilon\big)\big(\hat{v}_j(t)+\varepsilon\big)} - \gamma e_k(t), \qquad k=1,\dots,n, \tag{B.16}$$

where $\bar{y}_k(t) = y_k(t) - \hat{\mu}_k(t)$ and $e_k(t) = y_k(t) - \sum_{\ell=1}^{m} W_{k\ell}(t-1)x_\ell(t)$.

Define the regularized diagonal matrix

$$\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t) := \hat{\boldsymbol{D}}^{\lambda}(t) + \varepsilon \boldsymbol{I}, \qquad \boldsymbol{B}^{\lambda,\varepsilon}(t) := \big(\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t)\big)^{-1/2}\hat{\boldsymbol{O}}^{\lambda}(t)\big(\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t)\big)^{-1/2}.$$

Let $\boldsymbol{r}(t) \in \mathbb{R}^n$ denote the omitted correction, with entries

$$r_k(t) := \frac{[\boldsymbol{B}^{\lambda,\varepsilon}(t)^2]_{kk}}{\hat{v}_k(t)+\varepsilon}\,\bar{y}_k(t), \qquad k=1,\dots,n.$$

Proposition B.1 (Sufficient condition for descent of the truncated direction)**.** *With the notation above and $\lambda \in (0,1)$, the exact gradient of the full second-order surrogate is*

$$\nabla_{\boldsymbol{y}} \mathcal{J}_t(\boldsymbol{y}) = -2\lambda(1-\lambda)\big(\boldsymbol{d}(t) + \boldsymbol{r}(t)\big). \tag{B.17}$$

Consequently, $\boldsymbol{d}(t)$ is a strict descent direction for $\mathcal{J}_t$ whenever

$$\|\boldsymbol{r}(t)\|_2 < \|\boldsymbol{d}(t)\|_2. \tag{B.18}$$

In particular, for sufficiently small $\eta > 0$, the unprojected step $\boldsymbol{y} + \eta \boldsymbol{d}(t)$ strictly decreases $\mathcal{J}_t$. Moreover,

$$\|\boldsymbol{r}(t)\|_2 \le \frac{\|\boldsymbol{B}^{\lambda,\varepsilon}(t)\|_2^2}{\hat{v}_{\min}^{(\varepsilon)}(t)} \, \|\bar{\boldsymbol{y}}(t)\|_2, \qquad \hat{v}_{\min}^{(\varepsilon)}(t) := \min_{1 \le k \le n} \big(\hat{v}_k(t) + \varepsilon\big), \tag{B.19}$$

so the simpler sufficient condition

$$\frac{\|\boldsymbol{B}^{\lambda,\varepsilon}(t)\|_2^2}{\hat{v}_{\min}^{(\varepsilon)}(t)} \, \|\bar{\boldsymbol{y}}(t)\|_2 < \|\boldsymbol{d}(t)\|_2 \tag{B.20}$$

also guarantees that $\boldsymbol{d}(t)$ is a strict descent direction.

Proof. Since $\hat{\boldsymbol{O}}^{\lambda}(t)$ is symmetric with zero diagonal,

$$[\boldsymbol{B}^{\lambda,\varepsilon}(t)^2]_{kk} = \sum_{j=1}^{n} B_{kj}^{\lambda,\varepsilon}(t)^2 = \sum_{\substack{j=1 \\ j \ne k}}^{n} \frac{\hat{c}_{kj}(t)^2}{\big(\hat{v}_k(t) + \varepsilon\big)\big(\hat{v}_j(t) + \varepsilon\big)}.$$

Therefore,

$$r_k(t) = \sum_{\substack{j=1 \\ j \ne k}}^{n} \frac{\hat{c}_{kj}(t)^2 \bar{y}_k(t)}{\big(\hat{v}_k(t) + \varepsilon\big)^2 \big(\hat{v}_j(t) + \varepsilon\big)}.$$

Combining this identity with the definition of $d_k(t)$ and Equation B.15, using $\gamma_0 = \lambda(1-\lambda)\gamma$, gives

$$\frac{\partial \mathcal{J}_t(\boldsymbol{y})}{\partial y_k(t)} = -2\lambda(1-\lambda)\big(d_k(t) + r_k(t)\big),$$

which proves Equation B.17.

For a small step along $\boldsymbol{d}(t)$, Taylor expansion gives

$$\mathcal{J}_t\big(\boldsymbol{y} + \eta \boldsymbol{d}(t)\big) = \mathcal{J}_t(\boldsymbol{y}) + \eta \, \langle \nabla_{\boldsymbol{y}} \mathcal{J}_t(\boldsymbol{y}), \boldsymbol{d}(t) \rangle + O(\eta^2).$$

Using Equation B.17 and the Cauchy–Schwarz inequality,

$$\begin{aligned} \langle \nabla_{\boldsymbol{y}} \mathcal{J}_t(\boldsymbol{y}), \boldsymbol{d}(t) \rangle &= -2\lambda(1-\lambda)\Big(\|\boldsymbol{d}(t)\|_2^2 + \langle \boldsymbol{r}(t), \boldsymbol{d}(t) \rangle\Big) \\ &\le -2\lambda(1-\lambda) \, \|\boldsymbol{d}(t)\|_2 \big(\|\boldsymbol{d}(t)\|_2 - \|\boldsymbol{r}(t)\|_2\big). \end{aligned}$$

Condition equation B.18 makes the last expression strictly negative, establishing descent for sufficiently small $\eta > 0$.

Finally, symmetry of $\boldsymbol{B}^{\lambda,\varepsilon}(t)$ implies

$$0 \le [\boldsymbol{B}^{\lambda,\varepsilon}(t)^2]_{kk} \le \|\boldsymbol{B}^{\lambda,\varepsilon}(t)\|_2^2.$$

It follows componentwise that

$$|r_k(t)| \le \frac{\|\boldsymbol{B}^{\lambda,\varepsilon}(t)\|_2^2}{\hat{v}_{\min}^{(\varepsilon)}(t)} \, |\bar{y}_k(t)|.$$

Squaring and summing over k proves Equation B.19. Equation B.20 then implies Equation B.18. □

C Domain-specific network realizations

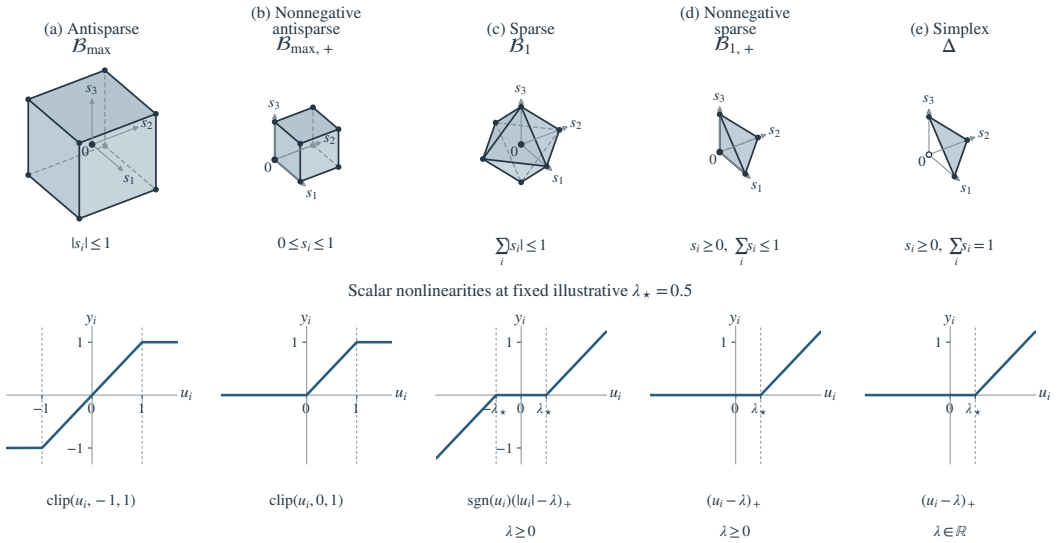


Figure 5: **Source domains and output nonlinearities.** Source domains in three dimensional Euclidean space (top) and corresponding scalar nonlinearities (bottom). The sparse, nonnegative sparse and simplex curves use an illustrative fixed $\lambda_\star = 0.5$; the network adapts a shared threshold λ to the domain constraint.

This section presents domain-specific realizations of the proposed network. Figure 5 illustrates the geometry of the source domains and their corresponding scalar output nonlinearities, derived in the following subsections. The domain assumption is application-specific; for example, our sparse dictionary-learning experiment (Figure 3) uses the sparse domain for natural-image representations.

The slow synaptic and statistical updates are identical across all source domains and are derived in Appendix B. The only domain-dependent part is the fast output inference step. All domains share the same descent direction, while the source domain determines the output nonlinearity and, when necessary, the dynamics of auxiliary inhibitory variables.

For each sample t and neural-dynamics iteration τ, we use the direction $d_k(t;\tau)$ from Equation 10. We further define the unconstrained pre-activation update

$$\tilde{y}_k(t;\tau+1) := y_k(t;\tau) + \eta_y(t,\tau)\, d_k(t;\tau). \tag{C.1}$$

The domain-specific architectures below are then obtained by combining Equation C.1 with the appropriate projection or proximal operator [35].

C.1 Antisparse sources

Consider the antisparse domain

$$\mathcal{B}_{\max} := \{\boldsymbol{s} \in \mathbb{R}^n : \|\boldsymbol{s}\|_\infty \le 1\}.$$

The output constraint is enforced directly through the clipping nonlinearity onto the box $[-1,1]^n$. The fast output dynamics are

$$\begin{aligned} \tilde{y}_k(t;\tau+1) &= y_k(t;\tau) + \eta_y(t,\tau)\, d_k(t;\tau), \\ y_k(t;\tau+1) &= \mathrm{clip}_{[-1,1]}\big(\tilde{y}_k(t;\tau+1)\big), \qquad k = 1,\dots,n, \end{aligned}$$

where

$$\mathrm{clip}_{[-1,1]}(u) := \begin{cases} -1, & u < -1, \\ u, & -1 \le u \le 1, \\ 1, & u > 1. \end{cases}$$

Algorithm 1 Predictive Entropy Maximization: generic online procedure

Input: streaming mixtures $\{\boldsymbol{x}(t)\}_{t=1}^{T}$, source domain $\mathcal{P}$
Hyperparameters: forgetting factor λ, feedforward step size schedule $\alpha_W(t)$, neural step size schedule $\eta_y(t,\tau)$, fast-dynamics horizon $\tau_{\max}$
Initialize $\boldsymbol{W}(0)$, $\hat{\boldsymbol{\mu}}^{\lambda}(0)$, $\{\hat{v}_i(0)\}_{i=1}^{n}$, $\{\hat{c}_{ij}(0)\}_{i\neq j}$, and $\theta_{\mathcal{P}}(0)$ if needed
for $t = 1, \dots, T$ **do**
  Initialize $\boldsymbol{y}(t;0)$ and $\theta_{\mathcal{P}}(t;0)$ if required by the domain
  for $\tau = 0, \dots, \tau_{\max} - 1$ **do**
    Compute $\boldsymbol{d}(t;\tau)$ from Equation 10
    $\boldsymbol{y}(t;\tau+1) \leftarrow \sigma_{\mathcal{P}}(\boldsymbol{y}(t;\tau) + \eta_y(t,\tau)\boldsymbol{d}(t;\tau)\,;\,\theta_{\mathcal{P}}(t;\tau))$
    Update $\theta_{\mathcal{P}}(t;\tau+1)$ if required by the domain
  end for
  Set $\boldsymbol{y}(t) \leftarrow \boldsymbol{y}(t;\tau_{\max})$ and $\boldsymbol{e}(t) \leftarrow \boldsymbol{y}(t) - \boldsymbol{W}(t-1)\boldsymbol{x}(t)$
  $\boldsymbol{W}(t) \leftarrow \boldsymbol{W}(t-1) + \alpha_W(t)\boldsymbol{e}(t)\boldsymbol{x}(t)^{\top}$
  $\hat{\boldsymbol{\mu}}^{\lambda}(t) \leftarrow \lambda\hat{\boldsymbol{\mu}}^{\lambda}(t-1) + (1-\lambda)\boldsymbol{y}(t), \quad \bar{\boldsymbol{y}}(t) \leftarrow \boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(t)$
  $\hat{v}_i(t) \leftarrow \lambda\hat{v}_i(t-1) + (1-\lambda)\bar{y}_i(t)^2, \qquad \forall i$
  $\hat{c}_{ij}(t) \leftarrow \lambda\hat{c}_{ij}(t-1) + (1-\lambda)\bar{y}_i(t)\bar{y}_j(t), \qquad \forall i \neq j$
end for

Thus, the corresponding network consists only of the output population, driven by the feedforward estimate $\boldsymbol{W}(t)\boldsymbol{x}(t)$, variance-normalized self-excitation, and covariance-dependent lateral inhibition.

C.2 Nonnegative antisparse sources

Consider the nonnegative antisparse domain

$$\mathcal{B}_{\max,+} := \mathcal{B}_{\max} \cap \mathbb{R}^n_+.$$

The dynamics are identical to the antisparse case except that the projection is now onto the box $[0,1]^n$:

$$\begin{aligned}
\tilde{y}_k(t;\tau+1) &= y_k(t;\tau) + \eta_y(t,\tau)\, d_k(t;\tau), \\
y_k(t;\tau+1) &= \mathrm{clip}_{[0,1]}\big(\tilde{y}_k(t;\tau+1)\big), \qquad k = 1, \dots, n,
\end{aligned}$$

where

$$\mathrm{clip}_{[0,1]}(u) := \begin{cases} 0, & u < 0, \\ u, & 0 \le u \le 1, \\ 1, & u > 1. \end{cases}$$

C.3 Sparse sources

Consider the sparse domain

$$\mathcal{B}_1 := \{\boldsymbol{s} \in \mathbb{R}^n : \|\boldsymbol{s}\|_1 \le 1\}.$$

We enforce the ℓ_1-constraint through a shared nonnegative variable λ_L, which acts as an adaptive threshold. Equivalently, one may view the fast-time update as a proximal-gradient step for the ℓ_1-penalized objective associated with the Lagrangian

$$\mathcal{J}_t(\boldsymbol{y}) + \lambda_L(\|\boldsymbol{y}\|_1 - 1), \qquad \lambda_L \ge 0.$$

Using the proximal mapping of the ℓ_1-norm [35], the resulting updates are

$$\begin{aligned}
\tilde{y}_k(t;\tau+1) &= y_k(t;\tau) + \eta_y(t,\tau)\, d_k(t;\tau), \\
y_k(t;\tau+1) &= \mathrm{ST}_{\lambda_L(t;\tau)}\big(\tilde{y}_k(t;\tau+1)\big), \qquad k = 1, \dots, n, \\
\lambda_L(t;\tau+1) &= \mathrm{ReLU}\left(\lambda_L(t;\tau) + \eta_\lambda(t,\tau)\left(\sum_{i=1}^{n} |y_i(t;\tau+1)| - 1\right)\right),
\end{aligned}$$

where $\mathrm{ST}_\lambda(u) := \mathrm{sign}(u)\max\{|u| - \lambda, 0\}$ is the *soft-thresholding* operator. The resulting architecture contains one additional inhibitory unit encoding λ_L, which delivers a common threshold to all output neurons.

C.4 Nonnegative sparse sources

Consider the nonnegative sparse domain

$$\mathcal{B}_{1,+} := \mathcal{B}_1 \cap \mathbb{R}^n_+.$$

Since $\boldsymbol{y} \geq 0$, the ℓ_1-constraint reduces to $\mathbf{1}^\top \boldsymbol{y} \leq 1$. As in the sparse case, we introduce a shared nonnegative variable λ_L and interpret the fast-time step as a proximal / projected-gradient update for

$$\mathcal{J}_t(\boldsymbol{y}) + \lambda_L(\mathbf{1}^\top \boldsymbol{y} - 1), \qquad \lambda_L \geq 0,$$

followed by projection onto the nonnegative orthant [35]. The resulting updates are

$$\begin{aligned}
\tilde{y}_k(t;\tau+1) &= y_k(t;\tau) + \eta_y(t,\tau)\, d_k(t;\tau), \\
y_k(t;\tau+1) &= \mathrm{ReLU}\big(\tilde{y}_k(t;\tau+1) - \lambda_L(t;\tau)\big), \qquad k = 1,\dots,n, \\
\lambda_L(t;\tau+1) &= \mathrm{ReLU}\left(\lambda_L(t;\tau) + \eta_\lambda(t,\tau)\left(\sum_{i=1}^n y_i(t;\tau+1) - 1\right)\right).
\end{aligned}$$

Thus, the nonnegative sparse dynamics correspond to a thresholded ReLU nonlinearity driven by the shared inhibitory variable λ_L.

C.5 Simplex sources

Consider the simplex domain

$$\Delta := \{\boldsymbol{s} \in \mathbb{R}^n : \boldsymbol{s} \geq \mathbf{0},\ \mathbf{1}^\top \boldsymbol{s} = 1\}.$$

The only difference with respect to the nonnegative sparse case is that the population constraint is now an equality rather than an inequality. Accordingly, the shared offset variable λ_L is no longer constrained to be nonnegative. The same proximal / projected interpretation then yields

$$\begin{aligned}
\tilde{y}_k(t;\tau+1) &= y_k(t;\tau) + \eta_y(t,\tau)\, d_k(t;\tau), \\
y_k(t;\tau+1) &= \mathrm{ReLU}\big(\tilde{y}_k(t;\tau+1) - \lambda_L(t;\tau)\big), \qquad k = 1,\dots,n, \\
\lambda_L(t;\tau+1) &= \lambda_L(t;\tau) + \eta_\lambda(t,\tau)\left(\sum_{i=1}^n y_i(t;\tau+1) - 1\right).
\end{aligned}$$

Thus, the simplex network has the same output architecture as the nonnegative sparse network, except that the inhibitory unit corresponding to λ_L is linear rather than rectified.

Summary. All five domains share the same feedforward pathway, the same covariance-driven recurrent interactions, and the same slow synaptic updates. What changes with the source domain is only the output-layer nonlinearity and, for $\mathcal{B}_1$, $\mathcal{B}_{1,+}$, and Δ, the inclusion of one shared inhibitory variable enforcing the corresponding population constraint. In particular, box-type domains require only local clipping operations, whereas sparse and simplex domains lead to piecewise-linear thresholding dynamics mediated by a single auxiliary inhibitory unit.

D Error analysis and analytical bounds for the second-order Taylor approximation

In this section, we analyze the approximation error induced by the second-order surrogate in Equation 7. The analysis proceeds in two steps. We first study the Taylor remainder pointwise for a generic covariance matrix, deriving an exact spectral representation together with sharp spectral and norm-based bounds. We then specialize these results to the online covariance sequence used by the algorithm and, finally, use the same bounds to relate the batch surrogate objective to the exact regularized determinant objective.

D.1 Pointwise remainder representation and bounds

We begin with a generic covariance matrix. Writing $\boldsymbol{C} = \boldsymbol{D} + \boldsymbol{O}$, where $\boldsymbol{D}$ collects the diagonal entries of $\boldsymbol{C}$ and $\boldsymbol{O}$ its off-diagonal part, and introducing the regularized diagonal matrix $\boldsymbol{D}^\varepsilon := \boldsymbol{D} + \varepsilon \boldsymbol{I}$, we define the normalized perturbation

$$\boldsymbol{B}^\varepsilon := (\boldsymbol{D}^\varepsilon)^{-1/2} \boldsymbol{O} (\boldsymbol{D}^\varepsilon)^{-1/2}.$$

This yields an exact spectral representation of the Taylor remainder for $\log\det(\boldsymbol{C} + \varepsilon \boldsymbol{I})$.

Theorem D.1 (Exact spectral representation of the second-order remainder)**.** *Fix $\varepsilon > 0$. Let $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ be a symmetric positive semidefinite matrix, and write*

$$\boldsymbol{C} = \boldsymbol{D} + \boldsymbol{O}, \qquad \boldsymbol{D} = \operatorname{diag}(\boldsymbol{C}),$$

where $\boldsymbol{D}$ is the diagonal matrix formed from the diagonal of $\boldsymbol{C}$ and $\boldsymbol{O}$ is the off-diagonal part of $\boldsymbol{C}$, so that $O_{ii} = 0$ for all i. Define the regularized diagonal matrix

$$\boldsymbol{D}^\varepsilon := \boldsymbol{D} + \varepsilon \boldsymbol{I},$$

the normalized off-diagonal matrix

$$\boldsymbol{B}^\varepsilon := (\boldsymbol{D}^\varepsilon)^{-1/2} \boldsymbol{O} (\boldsymbol{D}^\varepsilon)^{-1/2},$$

and let $\lambda_1, \dots, \lambda_n$ denote the eigenvalues of $\boldsymbol{B}^\varepsilon$. Then:

1. *$\boldsymbol{B}^\varepsilon$ is symmetric, $\operatorname{Tr}(\boldsymbol{B}^\varepsilon) = 0$, and $\lambda_i > -1$ for all $i = 1, \dots, n$;*
2. *the regularized log-determinant admits the exact decomposition*

$$\log\det(\boldsymbol{C} + \varepsilon \boldsymbol{I}) = \sum_{i=1}^n \log(C_{ii} + \varepsilon) - \frac{1}{2} \sum_{i=1}^n \lambda_i^2 + R_2, \tag{D.1}$$

where the remainder is given exactly by

$$R_2 = \sum_{i=1}^n \left[\log(1 + \lambda_i) - \lambda_i + \frac{1}{2} \lambda_i^2 \right]. \tag{D.2}$$

Moreover, the second-order term can be written entrywise as

$$\sum_{i=1}^n \lambda_i^2 = \operatorname{Tr}((\boldsymbol{B}^\varepsilon)^2) = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \frac{C_{ij}^2}{(C_{ii} + \varepsilon)(C_{jj} + \varepsilon)}, \tag{D.3}$$

and therefore

$$\log\det(\boldsymbol{C} + \varepsilon \boldsymbol{I}) = \sum_{i=1}^n \log(C_{ii} + \varepsilon) - \frac{1}{2} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \frac{C_{ij}^2}{(C_{ii} + \varepsilon)(C_{jj} + \varepsilon)} + R_2. \tag{D.4}$$

Proof. Since $\boldsymbol{C}$ is symmetric positive semidefinite, each diagonal entry satisfies $C_{ii} \geq 0$. Hence $\boldsymbol{D}^\varepsilon = \boldsymbol{D} + \varepsilon \boldsymbol{I}$ is positive definite and $(\boldsymbol{D}^\varepsilon)^{-1/2}$ is well-defined. Because $\boldsymbol{O} = \boldsymbol{C} - \boldsymbol{D}$ is symmetric and $(\boldsymbol{D}^\varepsilon)^{-1/2}$ is diagonal, the matrix

$$\boldsymbol{B}^\varepsilon = (\boldsymbol{D}^\varepsilon)^{-1/2} \boldsymbol{O} (\boldsymbol{D}^\varepsilon)^{-1/2}$$

is symmetric. Its diagonal entries are zero, since

$$(\boldsymbol{B}^\varepsilon)_{ii} = (D_{ii} + \varepsilon)^{-1/2} O_{ii} (D_{ii} + \varepsilon)^{-1/2} = 0, \qquad i = 1, \dots, n.$$

Therefore

$$\operatorname{Tr}(\boldsymbol{B}^\varepsilon) = 0. \tag{D.5}$$

Next, observe that

$$\boldsymbol{C} + \varepsilon \boldsymbol{I} = \boldsymbol{D}^\varepsilon + \boldsymbol{O} = (\boldsymbol{D}^\varepsilon)^{1/2} (\boldsymbol{I} + \boldsymbol{B}^\varepsilon) (\boldsymbol{D}^\varepsilon)^{1/2}. \tag{D.6}$$

Since $\boldsymbol{C} + \varepsilon \boldsymbol{I}$ is positive definite and $(\boldsymbol{D}^\varepsilon)^{1/2}$ is invertible, Equation D.6 implies that

$$\boldsymbol{I} + \boldsymbol{B}^\varepsilon = (\boldsymbol{D}^\varepsilon)^{-1/2}(\boldsymbol{C} + \varepsilon \boldsymbol{I})(\boldsymbol{D}^\varepsilon)^{-1/2}$$

is also positive definite. Hence all eigenvalues of $\boldsymbol{I} + \boldsymbol{B}^\varepsilon$ are strictly positive. If $\lambda_1, \dots, \lambda_n$ are the eigenvalues of $\boldsymbol{B}^\varepsilon$, then the eigenvalues of $\boldsymbol{I} + \boldsymbol{B}^\varepsilon$ are $1 + \lambda_1, \dots, 1 + \lambda_n$, and therefore $\lambda_i > -1$ for all $i = 1, \dots, n$.

Using Equation D.6 and multiplicativity of the determinant, we obtain

$$\det(\boldsymbol{C} + \varepsilon \boldsymbol{I}) = \det(\boldsymbol{D}^\varepsilon) \det(\boldsymbol{I} + \boldsymbol{B}^\varepsilon).$$

Taking logarithms yields

$$\log\det(\boldsymbol{C} + \varepsilon \boldsymbol{I}) = \log\det(\boldsymbol{D}^\varepsilon) + \log\det(\boldsymbol{I} + \boldsymbol{B}^\varepsilon). \tag{D.7}$$

Since $\boldsymbol{D}^\varepsilon$ is diagonal,

$$\log\det(\boldsymbol{D}^\varepsilon) = \sum_{i=1}^{n} \log(C_{ii} + \varepsilon). \tag{D.8}$$

Now let

$$\boldsymbol{B}^\varepsilon = \boldsymbol{Q} \operatorname{diag}(\lambda_1, \dots, \lambda_n) \boldsymbol{Q}^\top$$

be an eigendecomposition of $\boldsymbol{B}^\varepsilon$, with $\boldsymbol{Q}$ orthogonal. Then

$$\boldsymbol{I} + \boldsymbol{B}^\varepsilon = \boldsymbol{Q} \operatorname{diag}(1 + \lambda_1, \dots, 1 + \lambda_n) \boldsymbol{Q}^\top,$$

so

$$\log\det(\boldsymbol{I} + \boldsymbol{B}^\varepsilon) = \sum_{i=1}^{n} \log(1 + \lambda_i). \tag{D.9}$$

Substituting Equations D.8 and D.9 into Equation D.7 gives

$$\log\det(\boldsymbol{C} + \varepsilon \boldsymbol{I}) = \sum_{i=1}^{n} \log(C_{ii} + \varepsilon) + \sum_{i=1}^{n} \log(1 + \lambda_i). \tag{D.10}$$

Using Equation D.5, we have $\sum_{i=1}^{n} \lambda_i = 0$. Therefore we may add and subtract $\sum_i \lambda_i - \frac{1}{2}\sum_i \lambda_i^2$ inside Equation D.10 to obtain

$$\log\det(\boldsymbol{C} + \varepsilon \boldsymbol{I}) = \sum_{i=1}^{n} \log(C_{ii} + \varepsilon) - \frac{1}{2}\sum_{i=1}^{n} \lambda_i^2 + \sum_{i=1}^{n} \left[\log(1 + \lambda_i) - \lambda_i + \frac{1}{2}\lambda_i^2\right].$$

This proves Equations D.1 and D.2.

It remains to identify the quadratic term in coordinates. Since $\boldsymbol{B}^\varepsilon$ is symmetric,

$$\sum_{i=1}^{n} \lambda_i^2 = \operatorname{Tr}((\boldsymbol{B}^\varepsilon)^2).$$

Also, because $(\boldsymbol{B}^\varepsilon)_{ii} = 0$,

$$\operatorname{Tr}((\boldsymbol{B}^\varepsilon)^2) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (\boldsymbol{B}^\varepsilon)_{ij}^2.$$

Finally, for $i \neq j$,

$$(\boldsymbol{B}^\varepsilon)_{ij} = (D_{ii} + \varepsilon)^{-1/2} O_{ij} (D_{jj} + \varepsilon)^{-1/2} = \frac{C_{ij}}{\sqrt{(C_{ii} + \varepsilon)(C_{jj} + \varepsilon)}}.$$

Substituting this identity into the previous display proves Equation D.3, and Equation D.4 follows immediately. □

Remark D.2 (Why a second-order approximation?)**.** *Theorem D.1 identifies second order as the lowest order at which cross-covariances enter the approximation. Since* $\mathrm{Tr}(\boldsymbol{B}^\varepsilon) = 0$*, the first-order approximation to the negative log-determinant reduces to* $-\sum_i \log(C_{ii} + \varepsilon)$*. This term encourages componentwise variance expansion but does not penalize cross-covariances. Second order introduces the normalized pairwise covariance penalty in Equation D.3.*

This pairwise structure supports the biological locality criteria adopted in the introduction: local computation uses a neuron's inputs and their synaptic weights, while local plasticity uses variables available at the pre- and postsynaptic neurons [29]. Each covariance trace is learned from the corresponding pair's coactivity, together with neuron-specific variance estimates; see Equations B.6 and B.7. These traces support the direct pairwise recurrent interactions used here. The implemented activity rule also uses the separate gradient truncation in Appendix B.3: differentiation of the full second-order surrogate produces an additional term quadratic in the cross-covariances, which is omitted from that rule.

To identify the third-order correction, hold $\boldsymbol{D}^\varepsilon$ *and* $\boldsymbol{O}$ *fixed and introduce a scalar* a*. If* $\beta_1, \dots, \beta_n$ *are the eigenvalues of* $\boldsymbol{B}^\varepsilon$*, the factorization used in Equation D.6 gives, for* a *in a neighborhood of zero,*

$$F(a) := -\log\det(\boldsymbol{D}^\varepsilon + a\boldsymbol{O}) = -\log\det \boldsymbol{D}^\varepsilon - \sum_{r=1}^{n} \log(1 + a\beta_r).$$

Differentiating the scalar logarithms yields the cubic Taylor coefficient

$$\frac{F'''(0)}{3!} = -\frac{1}{3}\sum_{r=1}^{n} \beta_r^3 = -\frac{1}{3}\,\mathrm{Tr}\big((\boldsymbol{B}^\varepsilon)^3\big).$$

Let F_p *denote the degree-*p *Taylor polynomial of* F *about* $a = 0$*, evaluated at* $a = 1$*. This finite local expansion does not require convergence of an infinite series at* $a = 1$*. For the entrywise form, write* $d_i := C_{ii} + \varepsilon$*. Since* $(\boldsymbol{B}^\varepsilon)_{ii} = 0$ *and* $(\boldsymbol{B}^\varepsilon)_{ij} = C_{ij}/\sqrt{d_i d_j}$ *for* $i \neq j$*, matrix multiplication gives*

$$\mathrm{Tr}\big((\boldsymbol{B}^\varepsilon)^3\big) = \sum_{i,j,\ell} (\boldsymbol{B}^\varepsilon)_{ij}(\boldsymbol{B}^\varepsilon)_{j\ell}(\boldsymbol{B}^\varepsilon)_{\ell i}.$$

Any repeated-index term contains a zero diagonal entry, while the normalization factors in each remaining product multiply to $d_i d_j d_\ell$*. Thus the third-order correction can be written as*

$$\mathcal{T}_3 := F_3 - F_2 = -\frac{1}{3}\,\mathrm{Tr}\big((\boldsymbol{B}^\varepsilon)^3\big) = -\frac{1}{3}\sum_{\substack{i,j,\ell \\ \text{all distinct}}} \frac{C_{ij}C_{j\ell}C_{\ell i}}{d_i d_j d_\ell}.$$

To derive the activity gradient, set $\boldsymbol{C} = \hat{\boldsymbol{C}}^\lambda(t)$ *and use the steady-state statistic recursions from Appendix B. Hold the past statistics fixed while retaining the dependence of the current mean and covariance on* $\boldsymbol{y}(t)$*. Suppressing the time argument, define* $q := \lambda(1-\lambda)$ *and* $\bar{y}_i := y_i(t) - \hat{\mu}_i(t)$*. Equations B.10 and B.11 then give*

$$\frac{\partial C_{ab}}{\partial y_k} = q\big(\delta_{ak}\bar{y}_b + \delta_{bk}\bar{y}_a\big), \quad a \neq b, \qquad \frac{\partial d_i}{\partial y_k} = 2q\bar{y}_k\delta_{ik}.$$

For differentiation, regard the symmetric off-diagonal entries C_{ab}*,* $a < b$*, and the diagonal variables* d_i *as independent coordinates. Symmetry and cyclic reindexing of the triple sum yield*

$$\frac{\partial \mathcal{T}_3}{\partial C_{ab}} = -\frac{2}{d_a d_b}\sum_{j \notin \{a,b\}} \frac{C_{aj}C_{jb}}{d_j}, \qquad a < b,$$

$$\frac{\partial \mathcal{T}_3}{\partial d_k} = \frac{1}{d_k^2}\sum_{\substack{j,\ell \\ k,j,\ell \text{ distinct}}} \frac{C_{kj}C_{j\ell}C_{\ell k}}{d_j d_\ell}.$$

In the first identity, an off-diagonal entry occurs in three positions and both orientations, giving the factor $6/3 = 2$*. In the second, the three denominator positions cancel the factor* $1/3$*; differentiating*

their reciprocals reverses the sign. The chain rule therefore gives the full contribution of the cubic correction:

$$\frac{\partial \mathcal{T}_3}{\partial y_k} = \sum_{a<b} \frac{\partial \mathcal{T}_3}{\partial C_{ab}} \frac{\partial C_{ab}}{\partial y_k} + \sum_i \frac{\partial \mathcal{T}_3}{\partial d_i} \frac{\partial d_i}{\partial y_k} = g_k^{\text{path}} + \frac{2q\bar{y}_k}{d_k^2} \sum_{\substack{j,\ell \\ k,j,\ell \text{ distinct}}} \frac{C_{kj} C_{j\ell} C_{\ell k}}{d_j d_\ell}, \tag{D.11}$$

where the covariance-path contribution is

$$g_k^{\text{path}} := -2q \sum_{\ell \neq k} \frac{\bar{y}_\ell}{d_k d_\ell} \sum_{j \notin \{k,\ell\}} \frac{C_{kj} C_{j\ell}}{d_j}. \tag{D.12}$$

In the sums over (j, ℓ), both orders of each admissible pair are included. Thus Equation D.12 follows from differentiating the covariance factors, while the second term in Equation D.11 accounts for the variance denominators.

In Equation D.12, the coefficient multiplying $\bar{y}_\ell$ contains products $C_{kj}C_{j\ell}$ along paths through an intermediate neuron j. Evaluating this coefficient at the direct ℓ-to-k synapse requires covariance traces beyond that synapse and its endpoint variables. Although the individual traces can be learned locally, combining them would require additional communication or auxiliary circuitry. Second order therefore provides the lowest-order cross-covariance correction compatible with the direct pairwise construction used here, together with the separate gradient truncation described above. The remainder bounds below quantify the error of the second-order objective approximation.

Theorem D.1 shows that the approximation error is determined by the spectrum of the regularized normalized perturbation $\boldsymbol{B}^\varepsilon = (\boldsymbol{D}^\varepsilon)^{-1/2}\boldsymbol{O}(\boldsymbol{D}^\varepsilon)^{-1/2}$. This identifies $\boldsymbol{B}^\varepsilon$ as the natural perturbation variable for assessing the accuracy of the regularized Taylor surrogate. The next result turns the exact representation above into a sharp two-sided bound on the remainder.

Theorem D.3 (Two-sided spectral bound)**.** *Under the assumptions and notation of Theorem D.1, the remainder R_2 satisfies*

$$-\frac{1}{3} \sum_{\lambda_i < 0} \frac{|\lambda_i|^3}{1+\lambda_i} \;\le\; R_2 \;\le\; \frac{1}{3} \sum_{\lambda_i \ge 0} \lambda_i^3. \tag{D.13}$$

Consequently,

$$|R_2| \le \max\left\{ \frac{1}{3} \sum_{\lambda_i \ge 0} \lambda_i^3,\; \frac{1}{3} \sum_{\lambda_i < 0} \frac{|\lambda_i|^3}{1+\lambda_i} \right\}. \tag{D.14}$$

Proof. Define the scalar function

$$r(x) := \log(1+x) - x + \frac{1}{2}x^2, \qquad x > -1.$$

By Equation D.2,

$$R_2 = \sum_{i=1}^n r(\lambda_i). \tag{D.15}$$

Also,

$$r(0) = 0, \qquad r'(x) = \frac{x^2}{1+x}.$$

Hence

$$r(x) = \int_0^x \frac{t^2}{1+t}\, dt, \qquad x > -1. \tag{D.16}$$

If $x \ge 0$, then $1 + t \ge 1$ for all $t \in [0, x]$, and Equation D.16 gives

$$0 \le r(x) = \int_0^x \frac{t^2}{1+t}\, dt \le \int_0^x t^2\, dt = \frac{x^3}{3}.$$

Thus

$$0 \le r(x) \le \frac{x^3}{3}, \qquad x \ge 0. \tag{D.17}$$

If $-1 < x < 0$, then

$$r(x) = -\int_x^0 \frac{t^2}{1+t}\,dt \le 0.$$

Moreover, for $t \in [x, 0]$, we have $1 + t \ge 1 + x > 0$, so

$$\frac{1}{1+t} \le \frac{1}{1+x}.$$

Therefore

$$|r(x)| = -r(x) = \int_x^0 \frac{t^2}{1+t}\,dt \le \frac{1}{1+x}\int_x^0 t^2\,dt = \frac{|x|^3}{3(1+x)}.$$

Hence

$$-\frac{|x|^3}{3(1+x)} \le r(x) \le 0, \qquad -1 < x < 0. \tag{D.18}$$

Applying Equation D.17 to the nonnegative eigenvalues and Equation D.18 to the negative eigenvalues, and summing the resulting inequalities in Equation D.15, yields Equation D.13. Equation D.14 follows immediately. □

The preceding spectral bound immediately yields a simpler estimate expressed directly in terms of the size and conditioning of the regularized normalized perturbation $\boldsymbol{B}^\varepsilon$.

Corollary D.4 (Simpler norm-based bound)**.** *Under the assumptions and notation of Theorem D.1, let $\lambda_{\min}(\boldsymbol{B}^\varepsilon)$ denote the smallest eigenvalue of the normalized off-diagonal matrix $\boldsymbol{B}^\varepsilon$. Then*

$$|R_2| \le \frac{1}{3\big(1+\lambda_{\min}(\boldsymbol{B}^\varepsilon)\big)} \sum_{i=1}^n |\lambda_i|^3 \le \frac{\|\boldsymbol{B}^\varepsilon\|_F^2 \|\boldsymbol{B}^\varepsilon\|_2}{3\big(1+\lambda_{\min}(\boldsymbol{B}^\varepsilon)\big)}. \tag{D.19}$$

Proof. From Theorem D.3, we have

$$|R_2| \le \max\left\{ \frac{1}{3}\sum_{\lambda_i \ge 0} \lambda_i^3,\ \frac{1}{3}\sum_{\lambda_i < 0} \frac{|\lambda_i|^3}{1+\lambda_i} \right\}. \tag{D.20}$$

We bound the two terms on the right-hand side separately.

First, since $\mathrm{Tr}(\boldsymbol{B}^\varepsilon) = 0$, either $\boldsymbol{B}^\varepsilon = \boldsymbol{0}$ or $\lambda_{\min}(\boldsymbol{B}^\varepsilon) \le 0$. In either case,

$$1 + \lambda_{\min}(\boldsymbol{B}^\varepsilon) \le 1.$$

Therefore, for every eigenvalue with $\lambda_i \ge 0$,

$$\lambda_i^3 \le \frac{\lambda_i^3}{1+\lambda_{\min}(\boldsymbol{B}^\varepsilon)} \le \frac{|\lambda_i|^3}{1+\lambda_{\min}(\boldsymbol{B}^\varepsilon)}.$$

Hence

$$\frac{1}{3}\sum_{\lambda_i \ge 0} \lambda_i^3 \le \frac{1}{3\big(1+\lambda_{\min}(\boldsymbol{B}^\varepsilon)\big)} \sum_{\lambda_i \ge 0} |\lambda_i|^3. \tag{D.21}$$

Second, for every eigenvalue with $\lambda_i < 0$, we have $\lambda_i \ge \lambda_{\min}(\boldsymbol{B}^\varepsilon)$, so

$$1 + \lambda_i \ge 1 + \lambda_{\min}(\boldsymbol{B}^\varepsilon) > 0.$$

Consequently,

$$\frac{|\lambda_i|^3}{1+\lambda_i} \le \frac{|\lambda_i|^3}{1+\lambda_{\min}(\boldsymbol{B}^\varepsilon)}.$$

Summing over all negative eigenvalues gives

$$\frac{1}{3}\sum_{\lambda_i < 0} \frac{|\lambda_i|^3}{1+\lambda_i} \le \frac{1}{3\big(1+\lambda_{\min}(\boldsymbol{B}^\varepsilon)\big)} \sum_{\lambda_i < 0} |\lambda_i|^3. \tag{D.22}$$

Combining Equations D.21 and D.22 with Equation D.20, we obtain

$$|R_2| \le \frac{1}{3\big(1+\lambda_{\min}(\boldsymbol{B}^{\varepsilon})\big)} \sum_{i=1}^{n} |\lambda_i|^3.$$

For the second inequality, note that

$$\sum_{i=1}^{n} |\lambda_i|^3 = \sum_{i=1}^{n} |\lambda_i|^2 |\lambda_i| \le \left(\max_{1\le i\le n} |\lambda_i| \right) \sum_{i=1}^{n} \lambda_i^2 = \|\boldsymbol{B}^{\varepsilon}\|_2 \|\boldsymbol{B}^{\varepsilon}\|_F^2.$$

Substituting this into the previous inequality proves Equation D.19. □

We now specialize the preceding pointwise analysis to the online covariance sequence generated by the algorithm.

Corollary D.5 (Online version)**.** *Consider the online Predictive Entropy Maximization algorithm, and suppose that at time t the exponentially weighted output covariance $\hat{\boldsymbol{C}}^{\lambda}(t)$ is symmetric positive semidefinite. Let*

$$\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t) := \hat{\boldsymbol{D}}^{\lambda}(t) + \varepsilon \boldsymbol{I}, \qquad \hat{\boldsymbol{C}}^{\lambda}(t) = \hat{\boldsymbol{D}}^{\lambda}(t) + \hat{\boldsymbol{O}}^{\lambda}(t),$$

and define the normalized off-diagonal matrix

$$\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t) := \big(\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t)\big)^{-1/2} \hat{\boldsymbol{O}}^{\lambda}(t) \big(\hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t)\big)^{-1/2}.$$

Let $\hat{\lambda}_1(t), \dots, \hat{\lambda}_n(t)$ denote the eigenvalues of $\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t)$. Then the remainder $R_2(t)$ of the second-order surrogate in Equation 7 satisfies

$$-\frac{1}{3} \sum_{\hat{\lambda}_i(t)<0} \frac{|\hat{\lambda}_i(t)|^3}{1+\hat{\lambda}_i(t)} \;\le\; R_2(t) \;\le\; \frac{1}{3} \sum_{\hat{\lambda}_i(t)\ge 0} \hat{\lambda}_i(t)^3,$$

and therefore

$$|R_2(t)| \le \max\left\{ \frac{1}{3} \sum_{\hat{\lambda}_i(t)\ge 0} \hat{\lambda}_i(t)^3,\; \frac{1}{3} \sum_{\hat{\lambda}_i(t)<0} \frac{|\hat{\lambda}_i(t)|^3}{1+\hat{\lambda}_i(t)} \right\}.$$

Proof. Apply Theorem D.1 and Theorem D.3 with

$$\boldsymbol{C} = \hat{\boldsymbol{C}}^{\lambda}(t), \qquad \boldsymbol{D} = \hat{\boldsymbol{D}}^{\lambda}(t), \qquad \boldsymbol{D}^{\varepsilon} = \hat{\boldsymbol{D}}^{\lambda,\varepsilon}(t), \qquad \boldsymbol{O} = \hat{\boldsymbol{O}}^{\lambda}(t).$$

Since $\hat{\boldsymbol{C}}^{\lambda}(t)$ is symmetric positive semidefinite and $\varepsilon > 0$, all assumptions are satisfied. □

Corollary D.5 is the bound used in our numerical diagnostics. It shows that the approximation error is controlled by the spectrum of the regularized normalized off-diagonal covariance matrix $\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t)$. In particular, the surrogate is accurate when the diagonally normalized off-diagonal covariance is small and the eigenvalues of $\hat{\boldsymbol{B}}^{\lambda,\varepsilon}(t)$ stay well away from the singular value -1.

D.2 Batch surrogate optimality relative to the exact determinant objective

The previous subsection controls the Taylor remainder pointwise for a fixed covariance matrix. We now use that control at the objective level. Although the surrogate in Section 3 is introduced through the online exponentially weighted covariance $\hat{\boldsymbol{C}}^{\lambda}(t)$, the expansion in Theorem D.1 and the remainder bounds above are pointwise statements about an arbitrary covariance matrix. It is therefore useful to consider the corresponding *batch* counterpart, obtained by replacing $\hat{\boldsymbol{C}}^{\lambda}(t)$ with the ordinary centered sample covariance. This auxiliary analysis does not address the convergence of the online dynamics; rather, it clarifies what minimizing the surrogate preserves from the exact regularized determinant objective itself.

Let $\mathcal{Y}$ be a nonempty family of feasible output matrices $\boldsymbol{Y} = [\boldsymbol{y}(1), \dots, \boldsymbol{y}(T)] \in \mathbb{R}^{n \times T}$ such that $\boldsymbol{y}(t) \in \mathcal{P}$ for all t, and such that the centered sample covariance

$$\hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y}) := \frac{1}{T} \sum_{t=1}^{T} \big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(\boldsymbol{Y})\big)\big(\boldsymbol{y}(t) - \hat{\boldsymbol{\mu}}^{\lambda}(\boldsymbol{Y})\big)^{\top}, \qquad \hat{\boldsymbol{\mu}}^{\lambda}(\boldsymbol{Y}) := \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{y}(t),$$

is symmetric positive semidefinite for every $\boldsymbol{Y} \in \mathcal{Y}$. For each $\boldsymbol{Y} \in \mathcal{Y}$, write

$$\boldsymbol{D}(\boldsymbol{Y}) := \operatorname{diag}\big(\hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y})\big), \qquad \boldsymbol{D}^{\varepsilon}(\boldsymbol{Y}) := \boldsymbol{D}(\boldsymbol{Y}) + \varepsilon \boldsymbol{I}, \qquad \boldsymbol{O}(\boldsymbol{Y}) := \hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y}) - \boldsymbol{D}(\boldsymbol{Y}),$$

$$\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y}) := \boldsymbol{D}^{\varepsilon}(\boldsymbol{Y})^{-1/2} \boldsymbol{O}(\boldsymbol{Y}) \boldsymbol{D}^{\varepsilon}(\boldsymbol{Y})^{-1/2}, \qquad \hat{v}_i(\boldsymbol{Y}) := [\hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y})]_{ii}, \qquad \hat{c}_{ij}(\boldsymbol{Y}) := [\hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y})]_{ij}.$$

We consider the exact batch determinant objective

$$\mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}) := -\log\det\big(\hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y}) + \varepsilon \boldsymbol{I}\big), \tag{D.23}$$

and its second-order batch surrogate

$$\mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}(\boldsymbol{Y}) := -\sum_{i=1}^{n} \log\big(\hat{v}_i(\boldsymbol{Y}) + \varepsilon\big) + \frac{1}{2} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\hat{c}_{ij}(\boldsymbol{Y})^2}{\big(\hat{v}_i(\boldsymbol{Y}) + \varepsilon\big)\big(\hat{v}_j(\boldsymbol{Y}) + \varepsilon\big)}.$$

The next result shows that if the regularized normalized off-diagonal perturbation remains uniformly controlled over the feasible family $\mathcal{Y}$, then every global minimizer of the surrogate is nearly optimal for the exact regularized determinant objective.

Theorem D.6 (Uniform approximation and exact-objective near-optimality)**.** *Define*

$$\bar{\varepsilon}_{\mathcal{Y}} := \sup_{\boldsymbol{Y} \in \mathcal{Y}} \frac{\|\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y})\|_F^2 \|\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y})\|_2}{3\big(1 + \lambda_{\min}(\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y}))\big)}. \tag{D.24}$$

Assume $\bar{\varepsilon}_{\mathcal{Y}} < \infty$. Then:

1. *for every $\boldsymbol{Y} \in \mathcal{Y}$,*
$$\big|\mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}(\boldsymbol{Y}) - \mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y})\big| \leq \bar{\varepsilon}_{\mathcal{Y}}; \tag{D.25}$$
2. *if $\boldsymbol{Y}_{\mathrm{sur}}$ is a global minimizer of $\mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}$ over $\mathcal{Y}$, then*
$$\mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}_{\mathrm{sur}}) \leq \inf_{\boldsymbol{Y} \in \mathcal{Y}} \mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}) + 2\bar{\varepsilon}_{\mathcal{Y}}. \tag{D.26}$$

Proof. By Equation D.4, applied with $\boldsymbol{C} = \hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y})$, the difference between the surrogate and exact batch objectives is exactly the Taylor remainder:

$$\mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}(\boldsymbol{Y}) - \mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}) = R_2(\boldsymbol{Y}).$$

Applying Corollary D.4 with $\boldsymbol{C} = \hat{\boldsymbol{C}}_{\boldsymbol{y}}(\boldsymbol{Y})$ yields

$$|R_2(\boldsymbol{Y})| \leq \frac{\|\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y})\|_F^2 \|\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y})\|_2}{3\big(1 + \lambda_{\min}(\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y}))\big)} \leq \bar{\varepsilon}_{\mathcal{Y}},$$

which proves Equation D.25.

Now let $\boldsymbol{Y}_{\mathrm{sur}}$ be a global minimizer of $\mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}$ over $\mathcal{Y}$. Then

$$\mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}_{\mathrm{sur}}) \leq \mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}(\boldsymbol{Y}_{\mathrm{sur}}) + \bar{\varepsilon}_{\mathcal{Y}} = \inf_{\boldsymbol{Y} \in \mathcal{Y}} \mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}(\boldsymbol{Y}) + \bar{\varepsilon}_{\mathcal{Y}}.$$

Using Equation D.25 once more,

$$\inf_{\boldsymbol{Y} \in \mathcal{Y}} \mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}(\boldsymbol{Y}) \leq \inf_{\boldsymbol{Y} \in \mathcal{Y}} \big(\mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}) + \bar{\varepsilon}_{\mathcal{Y}}\big) = \inf_{\boldsymbol{Y} \in \mathcal{Y}} \mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}) + \bar{\varepsilon}_{\mathcal{Y}}.$$

Combining the last two displays proves Equation D.26. □

Theorem D.6 shows that the Taylor surrogate does more than approximate the exact regularized determinant objective pointwise: if the regularized normalized off-diagonal perturbation is uniformly small on the feasible family, then minimizing the surrogate yields an output matrix whose exact determinant objective value is close to the batch optimum.

Corollary D.7 (Explicit spectral form)**.** *Assume that there exists* $\rho \in (0, 1)$ *such that*

$$\|\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y})\|_2 \le \rho, \qquad \forall \boldsymbol{Y} \in \mathcal{Y}. \tag{D.27}$$

Then

$$\bar{\varepsilon}_{\mathcal{Y}} \le \frac{n\rho^3}{3(1-\rho)}.$$

Consequently, every global minimizer $\boldsymbol{Y}_{\mathrm{sur}}$ *of* $\mathcal{J}_{\mathrm{sur}}^{\mathrm{batch}}$ *over* $\mathcal{Y}$ *satisfies*

$$\mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}_{\mathrm{sur}}) \le \inf_{\boldsymbol{Y}\in\mathcal{Y}} \mathcal{J}_{\mathrm{det}}^{\mathrm{batch}}(\boldsymbol{Y}) + \frac{2n\rho^3}{3(1-\rho)}. \tag{D.28}$$

Proof. Under Equation D.27, we have

$$\lambda_{\min}(\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y})) \ge -\rho, \qquad \|\boldsymbol{B}^{\varepsilon}(\boldsymbol{Y})\|_F^2 \le n\rho^2, \qquad \forall \boldsymbol{Y} \in \mathcal{Y}.$$

Substituting these inequalities into Equation D.24 gives

$$\bar{\varepsilon}_{\mathcal{Y}} \le \frac{n\rho^2 \cdot \rho}{3(1-\rho)} = \frac{n\rho^3}{3(1-\rho)}.$$

The bound in Equation D.28 then follows directly from Theorem D.6. □

E Supplementary on numerical experiments

This section provides technical implementation details, extended experimental results, and additional information on the experimental protocols used throughout the paper.

E.1 Performance evaluation metric

To evaluate source recovery performance, we report the mean signal-to-noise ratio (mSNR) of the estimated sources. Let $\boldsymbol{s}_i$ and $\tilde{\boldsymbol{y}}_i$ denote the i-th ground-truth source and its corresponding permutation- and sign-corrected estimate, respectively. The mSNR is defined as

$$\mathrm{mSNR}\ (\mathrm{dB}) = \frac{1}{n}\sum_{i=1}^{n} 10\log_{10}\left(\frac{\|\boldsymbol{s}_i\|_2^2}{\|\boldsymbol{s}_i - \tilde{\boldsymbol{y}}_i\|_2^2}\right). \tag{E.1}$$

To assess statistical variability, all reported metrics are aggregated over $N = 30$ independent realizations (unless otherwise stated) with different random mixing matrices and noise seeds. We report the sample mean $\bar{\mu}$ together with the 95% confidence interval. Let σ denote the sample standard deviation and $SE = \sigma/\sqrt{N}$ the standard error of the mean. The confidence interval is computed as

$$\mathrm{CI}_{95\%} = \bar{\mu} \pm t_{\alpha/2,N-1}\left(\frac{\sigma}{\sqrt{N}}\right), \tag{E.2}$$

where $t_{\alpha/2,N-1}$ is the critical value of the Student t-distribution with $\alpha = 0.05$ and $N - 1$ degrees of freedom. For $N = 30$, this gives $t_{0.025,29} \approx 2.045$. Unless otherwise stated, all shaded envelopes in the reported plots correspond to this 95% confidence interval.

E.2 Nonnegative feature learning with predictive entropy maximization

We evaluate the nonnegative PEM network as an unsupervised feature learner on two image datasets. We assess reconstruction on *Olivetti* face images and both reconstruction and downstream classification on *Fashion-MNIST* dataset. In these experiments, we use $\varepsilon = 0.002$.

Olivetti faces. We use the 400 grayscale images in the Olivetti faces dataset[2]. Each image is downsampled from 64×64 to 32×32 pixels by average pooling. We randomly divide the images into 350 training and 50 held-out examples. The split is performed at the image level, so the same person may appear in both sets.

PEM learns 49 nonnegative feedforward filters over 150 epochs, with the training images shuffled in each epoch. Weights are initialized randomly with nonnegative values. For each image, the $\mathcal{B}_{\max,+}$ dynamics infer bounded activities $\boldsymbol{y} \in [0,1]^{49}$. We then update the feedforward weights as

$$\boldsymbol{W} \leftarrow \left[\boldsymbol{W} + \eta_t(\boldsymbol{y} - \boldsymbol{W}\boldsymbol{x})\boldsymbol{x}^\top\right]_+,$$
$$\eta_t = \frac{5 \times 10^{-5}}{1 + t/60000},$$

and update the running activity statistics. Here, $[\cdot]_+$ denotes elementwise clipping at zero. Figure 6 shows all learned feedforward filters. The learned filters emphasize facial structures such as the eyes, mouth, and face contours. Their linear responses provide the feedforward drive for the recurrent dynamics that infer the bounded nonnegative representation.



Figure 6: **Nonnegative feedforward filters learned from Olivetti faces.** All 49 rows of $\boldsymbol{W}$ are shown, reshaped to 32×32 pixels.

After training, we freeze the weights and running statistics and infer activity codes for both image sets. We fit a linear decoder to the training codes and images using ridge regularization with parameter 10^{-6}, then apply this decoder to both sets. This decoder is separate from the feedforward filters shown in Figure 6. For comparison, we fit 49-component PCA[3] and NMF[4] models to the training images using scikit-learn. Table 1 reports the relative reconstruction error, $\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F / \|\boldsymbol{X}\|_F$. PEM achieves a held-out error of 0.097, compared with 0.084 for PCA and 0.089 for NMF.

Table 1: **Reconstruction of Olivetti faces.** Relative reconstruction errors for 49-dimensional representations, using a single split of 350 training and 50 held-out images.

Algorithm	Training	Held-out
PEM	0.084	0.097
PCA	0.072	0.084
NMF	0.077	0.089

Fashion-MNIST. We randomly select 15000 images from the Fashion-MNIST [43] training split and divide them into 10000 training and 5000 held-out images. The official test split is unused. Pixel intensities are rescaled to $[0,1]$.

Using the same nonnegative learning rule, PEM learns 64 features without labels over 10 shuffled epochs. We use a constant learning rate of 10^{-4}, and at most 150 neural iterations per image. After training, we freeze the weights and running statistics and infer codes for both sets. We evaluate reconstruction using a new linear decoder fitted to the Fashion-MNIST training codes and images, with the ridge parameter 10^{-6}. We also fit 64-component PCA and NMF models to the training images.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_olivetti_faces.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

To evaluate classification, we standardize each representation using its training-set mean and standard deviation, then fit an ℓ_2-regularized logistic-regression classifier ($C = 1$, at most 1000 iterations) using the training labels. We apply the fitted standardization and classifier to the held-out images. This procedure is used for PEM, PCA, NMF, and the 784-dimensional raw pixel representation.

Table 2 reports reconstruction errors and held-out classification accuracy. The 64-dimensional PEM codes achieve 82.1% accuracy, compared with 82.6% for PCA, 82.9% for NMF, and 79.6% for raw pixels. These single-split results show that the learned PEM representation retains information useful for classification, although its reconstruction error is higher than those of PCA and NMF.

Figure 7 provides a qualitative visualization of the held-out PEM codes using t-SNE [45]. Class labels determine only the point colors and are not used to compute the embedding.

Table 2: Fashion-MNIST reconstruction error and held-out classification accuracy. Feature dimensions are 64 for PEM/PCA/NMF and 784 for raw pixels.

	Rel. error		Accuracy
Algorithm	Train	Test	(%)
PEM	0.271	0.273	82.1
PCA	0.222	0.225	82.6
NMF	0.248	0.249	82.9
Raw pixels	–	–	79.6

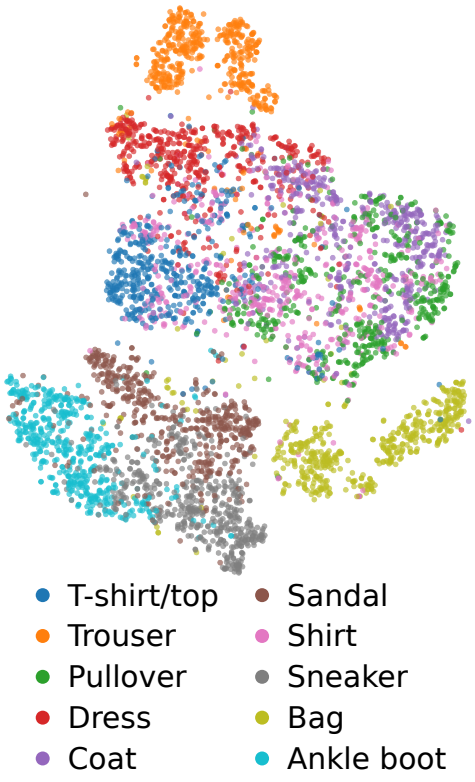


Figure 7: t-SNE of all 5000 held-out 64-dimensional PEM codes, colored by class.

E.3 Quantitative analysis of sparse dictionary learning

We assess how closely the 144 learned 12×12 receptive fields in Figure 3 resemble Gabor functions. Two-dimensional Gabor models are widely used to characterize the spatial structure of V1 simple-cell receptive fields [38]. We fit each learned filter with the model

$$G(x, y) = A \exp\left(-\frac{x_r^2}{2\sigma_x^2} - \frac{y_r^2}{2\sigma_y^2}\right) \cos(2\pi f x_r + \varphi) + c,$$

where

$$x_r = (x - x_0)\cos\theta + (y - y_0)\sin\theta,$$
$$y_r = -(x - x_0)\sin\theta + (y - y_0)\cos\theta.$$

The parameters (x_0, y_0) and θ specify the center and rotation of the filter. The remaining parameters are the amplitude A, offset c, envelope widths (σ_x, σ_y), spatial frequency f, and phase φ.

We estimate the parameters by bounded nonlinear least squares. Initial estimates of the center and envelope widths are obtained from the centroid and second moments of the squared filter values. The dominant nonzero peak of the two-dimensional power spectrum provides the initial frequency and orientation. For each filter, we retain the fit with the smallest residual sum of squares across eight initializations, combining four phases with two amplitude signs.

We measure goodness of fit using $R^2 = 1 - \mathrm{SS}_{\mathrm{res}}/\mathrm{SS}_{\mathrm{tot}}$, where $\mathrm{SS}_{\mathrm{res}}$ is the sum of squared fitting residuals and $\mathrm{SS}_{\mathrm{tot}}$ is the sum of squared deviations from the filter's spatial mean. To assess the fit achievable on unstructured inputs, we apply the same procedure to 144 random 12×12 filters with independent standard Gaussian entries. We also evaluate 144 receptive fields learned by Online

CorInfoMax [28] on the same natural-image patch task, providing a comparison with another learned dictionary.

Table 3: **Gabor fits to learned and random filters.** Statistics are computed across 144 filters per dictionary. The final two columns report the number of filters exceeding each R^2 threshold, out of 144.

Dictionary	Mean R^2	Median R^2	$R^2 > 0.75$	$R^2 > 0.80$
PEM	0.917	0.925	143/144	141/144
Online CorInfoMax	0.908	0.920	140/144	136/144
Gaussian random filters	0.123	0.113	0/144	0/144

Table 3 shows that Gabor functions closely approximate both learned dictionaries. PEM and Online CorInfoMax have similar mean and median R^2 values, substantially exceeding those of the Gaussian random filters.

We further examine the fitted rotation angles $\theta \bmod \pi$ of the 143 PEM filters with $R^2 > 0.75$. These angles occupy all 12 equal-width bins over $[0, \pi)$(Figure 8), demonstrating a broad range of orientations. Together, these measurements provide quantitative support for the Gabor-like structure and orientation diversity of the learned receptive fields.

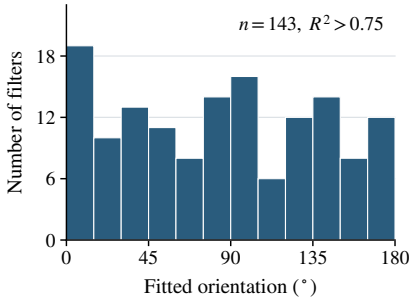


Figure 8: **Orientation diversity of PEM receptive fields.** Fitted angles $\theta \bmod \pi$ for the 143 filters with $R^2 > 0.75$, grouped into 15° bins.

E.4 Transform invariance in auditory source separation

In Section 4, we applied *Predictive Entropy Maximization* to audio mixtures in a sparse wavelet domain. The justification is straightforward. In the noise-free setting, let

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}, \qquad \boldsymbol{X} \in \mathbb{R}^{m\times T}, \qquad \boldsymbol{S} \in \mathbb{R}^{n\times T},$$

and let $\boldsymbol{\Phi} \in \mathbb{R}^{T\times T}$ denote a linear transform acting on the sample axis, such as the discrete wavelet transform. The transformed mixtures satisfy

$$\tilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{\Phi} = (\boldsymbol{A}\boldsymbol{S})\boldsymbol{\Phi} = \boldsymbol{A}(\boldsymbol{S}\boldsymbol{\Phi}) = \boldsymbol{A}\tilde{\boldsymbol{S}},$$

where $\tilde{\boldsymbol{S}} := \boldsymbol{S}\boldsymbol{\Phi}$. Thus, the same mixing matrix $\boldsymbol{A}$ governs the linear relation in the transform domain. This preserves the mixing geometry while allowing us to exploit the approximate sparsity of wavelet coefficients.

Accordingly, we learn a separator $\boldsymbol{W}$ such that $\boldsymbol{W}\tilde{\boldsymbol{X}} \approx \tilde{\boldsymbol{S}}$ using the $\mathcal{B}_1$ architecture. After separation, the time-domain sources are recovered through the inverse transform:

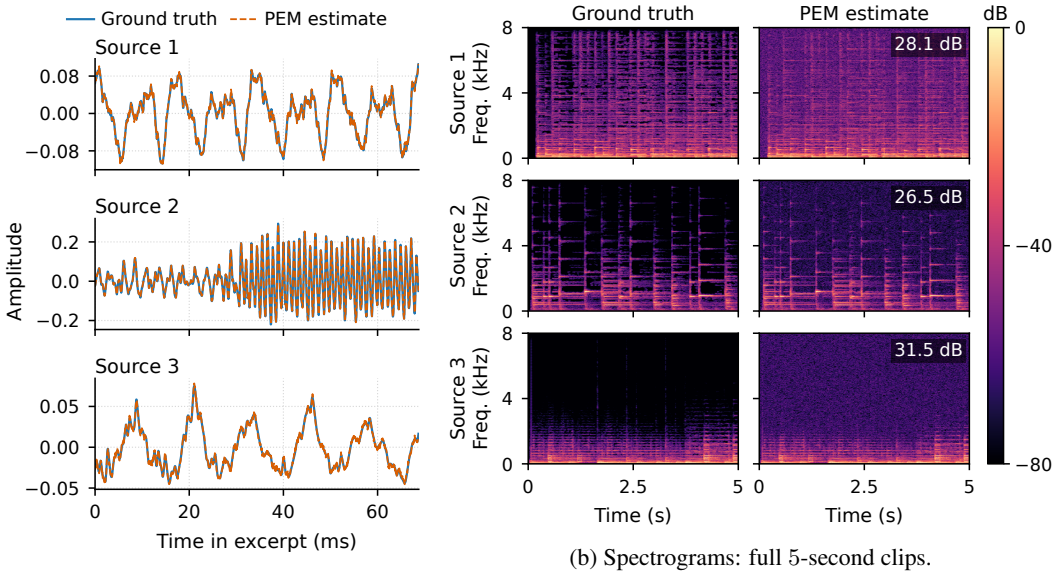
$$\hat{\boldsymbol{S}} = (\boldsymbol{W}\tilde{\boldsymbol{X}})\boldsymbol{\Phi}^{-1} \approx \tilde{\boldsymbol{S}}\boldsymbol{\Phi}^{-1} = \boldsymbol{S}.$$

For the specific trial illustrated in Figure 9, the mixing matrix was

$$\boldsymbol{A} = \begin{bmatrix} -1.131 & 0.696 & -0.432 \\ 0.741 & -0.478 & 1.386 \\ 0.125 & 1.149 & -2.350 \\ 0.183 & -0.311 & -0.294 \\ 0.400 & 1.006 & 0.502 \end{bmatrix},$$

and the resulting source-wise SNR values were 25.76 dB, 24.35 dB, and 30.53 dB.

The recovered spectrograms in Figure 9b retain the principal time–frequency structure of the three reference clips, with residual background energy visible in low-energy regions, complementing the waveform alignment in Figure 9a. For magnitude Short-Time Fourier Transform (STFT) arrays shown in Figure 9b M_i and $\hat{M}_i$ of each reference and aligned estimate, respectively, the magnitude-spectral SNR $10\log_{10}(\|M_i\|_F^2/\|M_i - \hat{M}_i\|_F^2)$ is 28.07, 26.49, and 31.51 dB, quantifying spectral-magnitude agreement.



(a) Waveforms: short excerpt.

(b) Spectrograms: full 5-second clips.

Figure 9: **Representative auditory source-separation result.** Temporal and spectral comparisons use the same permutation-, sign-, and scale-aligned estimates. Spectrograms use a 1024-sample Hann window/FFT, 256-sample hop, and 16 kHz sampling rate.

E.5 Antisparse, nonnegative sparse, and simplex source separation examples

Complementing the representative results in Figure 2, Figure 10 reports three additional synthetic domains.

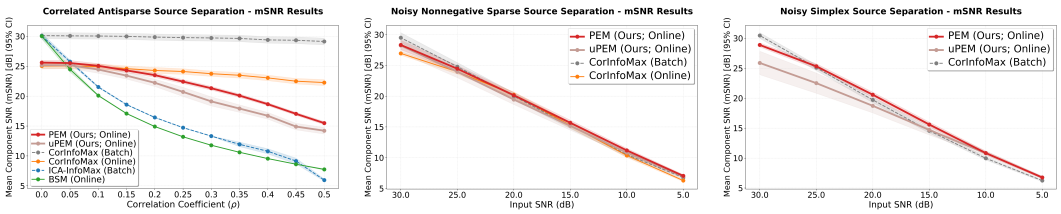
For the antisparse experiment, we use the domain $\mathcal{B}_{\max}$. As in the nonnegative antisparse case from the main text, sources are generated from a Copula-t model with four degrees of freedom, the source correlation level varies over $\rho \in \{0, 0.05, \dots, 0.5\}$, and the input SNR is fixed at 30 dB. Figure 10a shows that both *PEM* and *u-PEM* remain robust as correlation increases. In particular, they maintain strong performance relative to the biologically plausible baselines, while the batch *LD-InfoMax* method again provides the strongest overall reference performance. While *u-PEM* performs similarly to *PEM*, its performance declines slightly faster with increasing source correlation.

For the nonnegative sparse experiment, we use the domain $\mathcal{B}_{1,+}$ and vary the observation noise over $\mathrm{SNR}_{\mathrm{in}} \in \{30, 25, \dots, 5\}$ dB. Sources are generated uniformly from $\mathcal{B}_{1,+}$. Figure 10b shows that *PEM* and *u-PEM* remain competitive with the batch *LD-InfoMax* baseline across the full noise range, confirming that the surrogate-based online formulation extends naturally to nonnegative sparse domains as well.

For the simplex experiment, we use the domain Δ and vary the observation noise over $\mathrm{SNR}_{\mathrm{in}} \in \{30, 25, \dots, 5\}$ dB. Sources are generated uniformly from Δ. Figure 10c shows that *PEM* remains competitive with the batch *LD-InfoMax* baseline across the full noise range, confirming that the surrogate-based online formulation extends beyond box-constrained and sparse domains to simplex-structured sources as well. u-PEM also follows the other models closely, although it performs noticeably worse at high SNR values.

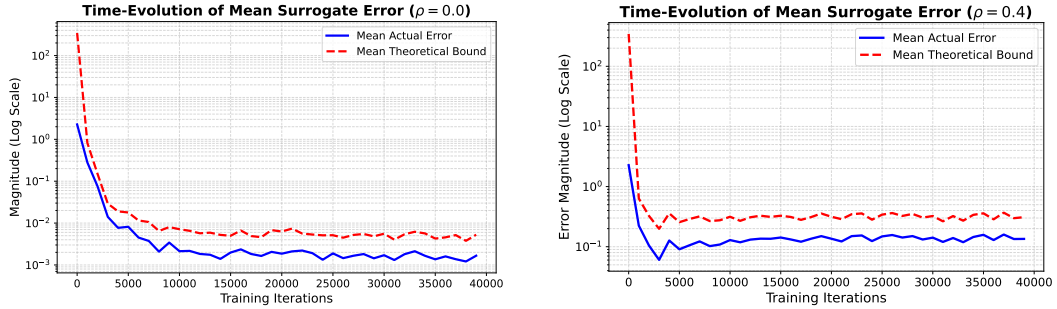
E.6 Transient diagnostics for the Taylor approximation

Figure 11 reports the time evolution of the exact Taylor remainder and the corresponding spectral upper bound for two representative correlation levels, $\rho = 0$ and $\rho = 0.4$. In both cases, the approximation error decreases during training as the dynamics suppress normalized off-diagonal covariance structure. The correlated case exhibits a larger error scale, as expected, but remains controlled by the theoretical bound throughout training. These trajectories complement the aggregate Taylor-surrogate diagnostics reported in the main text.



(a) Correlated antisparse ($\mathcal{B}_{\max}$) (b) Noisy nonneg. sparse ($\mathcal{B}_{1,+}$) (c) Noisy simplex (Δ)

Figure 10: **Additional performance comparisons. (a)** Mean component SNR (mSNR) as a function of the source correlation level ρ for antisparse sources. **(b)** Mean component SNR as a function of the input SNR for nonnegative sparse sources. **(c)** Mean component SNR as a function of the input SNR for simplex sources. These complementary experiments confirm the same pattern observed in the main text: *Predictive Entropy Maximization* remains robust under source correlation in antisparse domains and remains competitive with batch Det-Max baselines under increasing observation noise in nonnegative sparse- and simplex-structured domains. Shaded envelopes show 95% confidence intervals over 30 independent realizations.



(a) Uncorrelated sources ($\rho = 0$) (b) Correlated sources ($\rho = 0.4$)

Figure 11: **Transient Taylor-surrogate diagnostics.** Exact Taylor remainder and corresponding spectral upper bound over training iterations for two representative source-correlation levels.

E.7 Simulation hyperparameters

This section summarizes the hyperparameters used in all reported simulations. We first describe the generic implementation choices for *Predictive Entropy Maximization*, and then list the experiment-specific settings.

Generic implementation details for Predictive Entropy Maximization. Unless otherwise stated, we trained *Predictive Entropy Maximization* with a single online pass over the data, using the regularization constant $\varepsilon = 10^{-5}$. For each incoming sample $\boldsymbol{x}(t)$, the fast inference dynamics were initialized at $\boldsymbol{y}(t;0) = \mathbf{0}$. If no custom initialization was supplied, the feedforward matrix was initialized as

$$\boldsymbol{W}(0) = \boldsymbol{I} + \boldsymbol{\xi}_W, \tag{E.3}$$

where $\boldsymbol{I}$ denotes the rectangular identity and $\boldsymbol{\xi}_W$ has i.i.d. Gaussian entries with standard deviation 0.01. Likewise, if not specified explicitly, the running mean and covariance states were initialized as

$$\hat{\boldsymbol{\mu}}^{\lambda}(0) = \mathbf{0}, \qquad \hat{\boldsymbol{C}}^{\lambda}(0) = 0.2\,\boldsymbol{I}. \tag{E.4}$$

The slow feedforward update uses the step size schedule selected by `lr_W_rule`. In the experiments below, we used either:

$$\alpha_W(t) = \alpha_W^0 \qquad (\texttt{constant}), \tag{E.5}$$

$$\alpha_W(t) = \max\left\{\frac{\alpha_W^0}{t/T_W + 1},\, 10^{-8}\right\} \qquad (\texttt{divide_by_index}), \tag{E.6}$$

or

$$\alpha_W(t) = \max\left\{\frac{\alpha_W^0}{1+\log(t/T_W+2)},\, 10^{-8}\right\} \qquad \texttt{(divide_by_log_index)}, \tag{E.7}$$

where T_W is the decay divider.

The fast neural inference loop uses the step size schedule selected by `neural_lr_rule`. In the experiments below, we used either:

$$\eta_y(\tau) = \eta_y^0 \qquad \texttt{(constant)}, \tag{E.8}$$

$$\eta_y(\tau) = \max\left\{\frac{\eta_y^0}{\tau+1},\, \eta_y^{\min}\right\} \qquad \texttt{(divide_by_loop_index)}, \tag{E.9}$$

or

$$\eta_y(\tau) = \max\left\{\frac{\eta_y^0}{\tau T_y+1},\, \eta_y^{\min}\right\} \qquad \texttt{(divide_by_slow_loop_index)}, \tag{E.10}$$

where T_y is the neural learning-rate decay divider. For sparse, nonnegative sparse, and simplex domains, the scalar threshold/inhibitory variable λ_L was updated with step size η_λ, implemented in code as `stlambda_lr`.

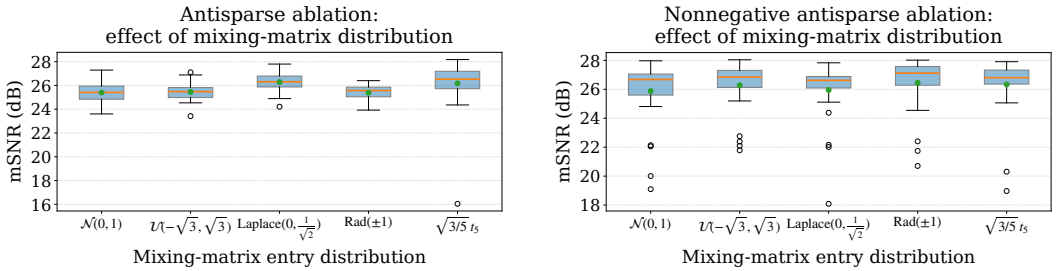
Predictive Entropy Maximization hyperparameters. The synthetic source-separation experiments use $n = 5$ sources, $m = 10$ mixtures, and $T = 10^5$ samples. The auditory experiment uses $n = 3$ sources and $m = 5$ mixtures. The sparse receptive-field experiment uses $n = m = 144$.

- **Correlated antisparse sources ($\mathcal{B}_{\max}$).** We used $\lambda_{\text{lat}} = 0.99$, $\gamma_{\text{pred}} = 250$, $\alpha_W^0 = 5 \times 10^{-2}$, $\eta_y^0 = 0.5$, $\eta_y^{\min} = 10^{-6}$, $\tau_{\max} = 250$, and tolerance 10^{-7}. The feedforward learning rate used `divide_by_index` with divider $T_W = 5000$, and the neural step size used `divide_by_loop_index`. No threshold variable was used.
- **Correlated nonnegative antisparse sources ($\mathcal{B}_{\max,+}$).** We used $\lambda_{\text{lat}} = 0.95$, $\gamma_{\text{pred}} = 750$, $\alpha_W^0 = 5 \times 10^{-2}$, $\eta_y^0 = 0.05$, $\eta_y^{\min} = 10^{-4}$, $\tau_{\max} = 500$, $\varepsilon = 10^{-4}$ and tolerance 10^{-6}. The feedforward learning rate used `divide_by_index` with divider $T_W = 20000$, and the neural step size used `divide_by_loop_index`. In this experiment we overrode the default initialization and set

  $$\hat{\boldsymbol{C}}^{\lambda}(0) = 2\boldsymbol{I}, \qquad \boldsymbol{W}(0) = 0.01\,\boldsymbol{I} + \frac{1}{15}\,\boldsymbol{\xi}_W, \tag{E.11}$$

  with $\boldsymbol{\xi}_W$ i.i.d. Gaussian.
- **Noisy sparse sources ($\mathcal{B}_1$).** We used $\lambda_{\text{lat}} = 0.99$, $\gamma_{\text{pred}} = 150$, $\alpha_W^0 = 5 \times 10^{-2}$, $\eta_y^0 = 0.05$, $\eta_y^{\min} = 10^{-4}$, $\eta_\lambda = 0.5$, $\tau_{\max} = 100$, and tolerance 10^{-6}. The feedforward learning rate used `divide_by_index` with divider $T_W = 5000$, and the neural step size used `divide_by_loop_index`.
- **Noisy nonnegative sparse sources ($\mathcal{B}_{1,+}$).** We used $\lambda_{\text{lat}} = 0.99$, $\gamma_{\text{pred}} = 250$, $\alpha_W^0 = 5 \times 10^{-2}$, $\eta_y^0 = 0.1$, $\eta_y^{\min} = 10^{-4}$, $\eta_\lambda = 0.5$, $\tau_{\max} = 100$, and tolerance 10^{-7}. The feedforward learning rate used `divide_by_index` with divider $T_W = 2000$, and the neural step size used `divide_by_loop_index`.
- **Noisy simplex sources (Δ).** We used $\lambda_{\text{lat}} = 0.99$, $\gamma_{\text{pred}} = 150$, $\alpha_W^0 = 5 \times 10^{-2}$, $\eta_y^0 = 0.1$, $\eta_y^{\min} = 10^{-4}$, $\eta_\lambda = 0.05$, $\tau_{\max} = 100$, and tolerance 10^{-7}. The feedforward learning rate used `divide_by_log_index` with divider $T_W = 5000$, and the neural step size used `divide_by_loop_index`.
- **Auditory source separation.** The sparse-domain audio experiment was run after transforming the mixtures to a wavelet domain using a db4 wavelet with decomposition level 3. The *Predictive Entropy Maximization* hyperparameters were $\lambda_{\text{lat}} = 0.95$, $\gamma_{\text{pred}} = 150$, $\alpha_W^0 = 9.5 \times 10^{-1}$, $\eta_y^0 = 0.01$, $\eta_y^{\min} = 10^{-4}$, $\eta_\lambda = 0.5$, $\tau_{\max} = 100$, and tolerance 10^{-6}. The feedforward learning rate used `divide_by_index` with divider $T_W = 2000$, and the neural step size used `divide_by_loop_index`.
- **Sparse receptive-field learning on natural images.** For the 12×12 patch experiment we used the sparse-domain architecture with

  $$\lambda_{\text{lat}} = 1 - \frac{10^{-3}}{7}, \qquad \gamma_{\text{pred}} = 3, \qquad \alpha_W^0 = 10^{-4}, \tag{E.12}$$



(a) Antisparse sources ($\mathcal{B}_{\max}$)

(b) Nonnegative antisparse sources ($\mathcal{B}_{\max,+}$)

Figure 12: **Ablation with respect to the distribution of the mixing-matrix entries.** Box plots summarize the distribution of mSNR over 30 seeds in the uncorrelated setting ($\rho = 0$) at input SNR 30 dB. All candidate mixing laws are centered and scaled to unit variance: $\mathcal{N}(0,1)$, $\mathcal{U}(-\sqrt{3},\sqrt{3})$, $\mathrm{Laplace}(0,1/\sqrt{2})$, $\mathrm{Rad}(\pm 1)$, and $\sqrt{3/5}\,t_5$. The performance of *Predictive Entropy Maximization* remains broadly stable across these choices, indicating that the method is not sensitive to the precise distributional shape of the mixing coefficients.

together with $\eta_y^0 = 0.05$, $\eta_y^{\min} = 10^{-6}$, $\eta_\lambda = 5 \times 10^{-2}$, $\tau_{\max} = 500$, and tolerance 10^{-6}. The feedforward learning rate was kept constant, while the neural step size used `divide_by_slow_loop_index` with divider $T_y = 100$. We also used custom initializations

$$\hat{\boldsymbol{C}}^{\lambda}(0) = \boldsymbol{I} + \frac{1}{250}\,\boldsymbol{\xi}_C, \qquad \boldsymbol{W}(0) = \boldsymbol{I} + \frac{1}{250}\,\boldsymbol{\xi}_W, \tag{E.13}$$

where $\boldsymbol{\xi}_C$ and $\boldsymbol{\xi}_W$ have i.i.d. Gaussian entries.

E.8 Ablation Studies

We report three additional ablations to better understand the robustness of *Predictive Entropy Maximization*. The first studies the sensitivity of the method to the distribution of the mixing-matrix entries. The second studies how performance varies with the number of mixtures while keeping the number of sources fixed. The third briefly investigates the underdetermined regime outside our theoretical assumptions.

E.8.1 Sensitivity to the mixing-matrix distribution

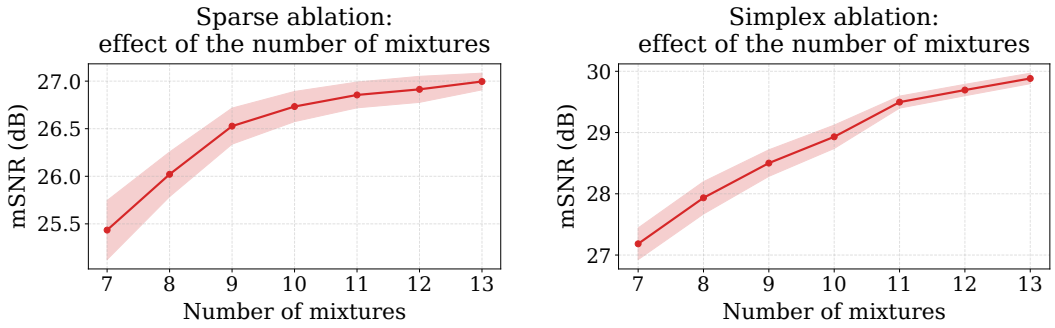
We first study whether the method is sensitive to the particular law used to generate the mixing matrix. For this ablation, we consider the antisparse and nonnegative antisparse domains, fix the source correlation level at $\rho = 0$, and fix the input SNR at 30 dB. In both cases, we use $n = 5$ sources, $m = 10$ mixtures, and $T = 10^5$ samples. Results are aggregated over 30 random seeds.

For each seed, we keep the source realization and the observation noise fixed and vary only the distribution of the entries of the mixing matrix $\boldsymbol{A}$. We consider five centered distributions, all scaled to have unit variance:

$$\mathcal{N}(0,1), \qquad \mathcal{U}(-\sqrt{3},\sqrt{3}), \qquad \mathrm{Laplace}(0,1/\sqrt{2}), \qquad \mathrm{Rad}(\pm 1), \qquad \sqrt{3/5}\,t_5.$$

Here, $\mathrm{Rad}(\pm 1)$ denotes a Rademacher random variable taking values ± 1 with equal probability, and t_5 denotes a Student-t random variable with 5 degrees of freedom. The scaling by $\sqrt{3/5}$ ensures unit variance, so that the comparison isolates the effect of distributional shape rather than trivial scale differences.

Figure 12 shows the resulting box plots for the mean component SNR. Across both domains, the central performance remains broadly stable across the five mixing laws. This indicates that the behavior of *Predictive Entropy Maximization* is not tied to Gaussian mixing matrices and transfers well to heavier-tailed, bounded, and discrete mixing distributions. The nonnegative antisparse case exhibits slightly larger variability across seeds, but again no systematic degradation under non-Gaussian or discrete mixing is observed.



(a) Sparse sources ($\mathcal{B}_1$)

(b) Simplex sources (Δ)

Figure 13: **Ablation with respect to the number of mixtures.** Mean mSNR is plotted against the number of mixtures $m \in \{7, \dots, 13\}$ while keeping the number of sources fixed at $n = 5$. The shaded envelopes show one standard error over 30 seeds. For each seed, a single Gaussian 13×5 mixing matrix is sampled and the experiment with m mixtures uses its first m rows, so the comparison isolates the effect of adding observation channels. In both the sparse and simplex domains, performance improves as the number of mixtures increases.

E.8.2 Effect of the number of mixtures

We next study how performance varies with the number of observed mixtures. For this ablation, we consider the sparse and simplex domains, fix the number of sources at $n = 5$, fix the input SNR at 30 dB, and vary the number of mixtures over

$$m \in \{7, 8, 9, 10, 11, 12, 13\}.$$

As before, each experiment uses $T = 10^5$ samples and is repeated over 30 random seeds.

For a fixed seed, we keep the source realization fixed and generate a single Gaussian mixing matrix of size 13×5. The experiment with m mixtures then uses the first m rows of this matrix. This nested construction makes the comparison more controlled, since increasing the number of mixtures corresponds to adding new observation channels rather than resampling a completely different mixing matrix.

Figure 13 reports the mean mSNR together with a standard-error envelope across seeds. In both domains, performance improves as the number of mixtures increases. This is consistent with the intuition that additional observation channels make the inverse problem easier by providing richer linear views of the same latent sources. The trend is visible already in the sparse case and is even more pronounced in the simplex setting. These results confirm that the method scales favorably with observation dimension in the overdetermined regime.

E.8.3 Underdetermined mixtures

Our theoretical analysis assumes $m \geq n$ and a full-column-rank mixing matrix $\boldsymbol{A}$. When $m < n$, any linear readout satisfies $\operatorname{rank}(\boldsymbol{W}\boldsymbol{A}) \leq m < n$. Thus, even without noise, a fixed linear readout cannot recover all n sources for arbitrary source realizations, up to sign, permutation and nonzero scaling. Specialized approaches to underdetermined separation exploit additional structure, including sparse representations [46, 47, 48], time–frequency structure [49, 50], or spatial covariance models [51].

To examine PEM's behavior outside the assumed mixing regime, we evaluate its linear readout $\hat{\boldsymbol{S}} = \boldsymbol{W}\boldsymbol{X}$ on an exploratory underdetermined experiment. We generate $n = 5$ independent $\mathcal{U}[-1, 1]$ sources and form $m \in \{3, 4\}$ mixtures using i.i.d. standard Gaussian mixing coefficients. Each trial contains $T = 10^5$ samples with additive white Gaussian noise at 30 dB input SNR. We train the antisparse network for one pass through the samples and repeat each setting over 30 random seeds.

As a reference, we compute the empirical linear minimum mean-square error (LMMSE) filter

$$\boldsymbol{W}_{\mathrm{LMMSE}} = \widehat{\boldsymbol{R}}_{sx}\widehat{\boldsymbol{R}}_{xx}^{-1}, \qquad \widehat{\boldsymbol{R}}_{sx} = \frac{\boldsymbol{S}\boldsymbol{X}^\top}{T}, \qquad \widehat{\boldsymbol{R}}_{xx} = \frac{\boldsymbol{X}\boldsymbol{X}^\top}{T}.$$

This ground-truth-assisted filter minimizes the squared reconstruction error over linear readouts without an intercept. It is fitted and evaluated on the same samples, providing an in-sample upper bound on source-wise SNR for such readouts.

We evaluate PEM on the same samples after a one-to-one assignment of outputs to sources that maximizes the sum of absolute Pearson correlations. Each matched output is then multiplied by its least-squares scalar gain, correcting its sign and scale. The reference outputs retain their known source order and receive no further alignment or rescaling.

Table 4: **Linear reconstruction from underdetermined mixtures.** For each one of the 30 trials, a mixing matrix and source are sampled. After running each method (LMMSE and PEM), the five sources are sorted (separately for the two methods) in decreasing order of source-wise SNR (dB). Entries report the mean $\pm$ sample standard deviation at each rank over 30 trials. Rank 1 denotes the highest SNR source (out of five); matching ranks need not correspond to the same source across methods or trials.

(a) $m = 4, n = 5$.

Rank	Empirical LMMSE	PEM
1	18.9 ± 4.5	16.9 ± 5.4
2	13.6 ± 3.6	11.7 ± 4.3
3	10.0 ± 3.1	7.4 ± 3.5
4	6.3 ± 1.7	4.2 ± 2.0
5	2.5 ± 1.1	1.0 ± 0.8

(b) $m = 3, n = 5$.

Rank	Empirical LMMSE	PEM
1	11.5 ± 3.3	8.5 ± 2.9
2	7.0 ± 2.8	5.3 ± 2.2
3	4.3 ± 1.3	3.1 ± 1.3
4	2.4 ± 0.8	1.6 ± 0.7
5	1.4 ± 0.7	0.7 ± 0.6

Table 4 shows lower mean SNRs at every rank when the number of mixtures decreases from four to three. Reconstruction quality varies substantially across ranks, and PEM remains below the empirical LMMSE reference. These results characterize the degradation of PEM's linear readout in the underdetermined setting. They do not establish underdetermined identifiability or compare PEM with methods specifically designed to exploit additional source structure.

E.9 Unnormalized Predictive Entropy Maximization

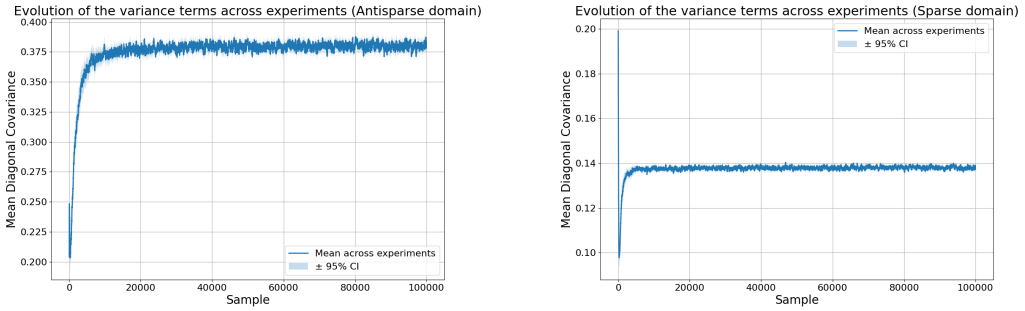
The *Predictive Entropy Maximization* (*PEM*) model includes cross-covariance terms that are variance normalized in both the energy function and the activity gradient update (Eqs. 9-10). If we remove this normalization term, the objective becomes

$$\mathcal{J}_t = -\sum_{i=1}^{n} \log(\hat{v}_i(t) + \varepsilon) + \frac{1}{2}\gamma_{lateral} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \hat{c}_{ij}(t)^2 + \gamma \|\boldsymbol{y}(t) - \boldsymbol{W}(t-1)\boldsymbol{x}(t)\|_2^2, \tag{E.14}$$

where we also introduced an additional hyperparameter $\gamma_{lateral}$ that controls the strength of the covariance term, corresponding to lateral connections. We call this unnormalized variant of the model *u-PEM*. Furthermore, the coordinate gradient update becomes (cf. Eq. 10)

$$d_k(t,\tau) = \frac{\bar{y}_k(t,\tau)}{\hat{v}_k(t) + \varepsilon} - \gamma_{lateral} \sum_{\substack{j=1 \\ j \neq k}}^{n} \hat{c}_{kj}(t)\, \bar{y}_j(t,\tau) - \gamma \left(y_k(t,\tau) - \sum_{\ell=1}^{m} W_{k\ell}(t-1) x_\ell(t) \right). \tag{E.15}$$

The update is now directly given by the product of the lateral weight and the neural activity, with no variance scaling, making *u-PEM* arguably even more biologically plausible than *PEM*. The removal of the variance normalization is further motivated by the observation that the mean variance of the *PEM* model appears to stabilise during training, as shown in Figure 14. Benchmark results for *u-PEM* and *PEM* are reported in Figures 2 and 10. The hyperparameter values $\gamma_{lateral}$ for the antisparse, non-negative antisparse, sparse, non-negative sparse and simplex domains were 10, 300, 50, 3200 and 100, respectively. These were selected via a hyperparameter sweep. The stabilised variance values v from *PEM* were used to calculate $1/v^2$ which served as a starting point to select the range of values to test for $\gamma_{lateral}$.



(a) Antisparse sources ($\mathcal{B}_{\max}$)

(b) Sparse sources ($\mathcal{B}_1$)

Figure 14: **Evolution of the variance term in *PEM* during training.** The evolution of the mean variance (averaged across the sources) is plotted across training samples for the *PEM* model. Solid lines and shaded bands indicate the mean and the 95% confidence interval over 30 seeds, respectively. For each seed, a single Gaussian 10×5 mixing matrix is sampled. These experiments demonstrate that the variance terms stabilise to a small range of values, motivating our simplification further. The values which the variance terms stabilise to can provide a starting point for the range of values we should sweep to select the best value for our hyperparameter $\gamma_{lateral}$.

E.10 Variance-covariance regularisation in self-supervised learning

An influential approach to self-supervised learning is to learn embeddings or latent representations of inputs (e.g. images) which are invariant to various transformations (e.g. crops). However, self-supervised objectives are prone to learning constant or degenerate embeddings (e.g. not using all the available dimensions), a phenomenon known as "representation collapse" [52, 53, 54]. Two main methods have been developed to prevent or mitigate against such collapse: (i) contrastive methods [55], which involve training with both positive and negative examples; and (ii) non-contrastive or regularized approaches, which directly modify the loss function and which are most closely related to the *Predictive Entropy Maximization* model.

For example, the objective of "Barlow Twins" [53] pushes the cross-correlation matrix between the embeddings of two identical networks towards the identity. This promotes both invariance to distortions (via the diagonal terms) and decorrelation (via the off-diagonal terms). The method was named after the neuroscientist H. Barlow for introducing the idea of redundancy-reduction, hypothesizing that the visual system minimizes the statistical dependency between neural firing rates [56] (which relates to our experiment on sparse receptive fields; see Figure 3).

Building on the Barlow Twins, "VICReg" [34] introduced a variance term that encourages spread in each embedding dimension. The overall loss function therefore contains a variance, a covariance and an invariance term. This is similar to the objective of the *Predictive Entropy Maximization* model, which also features variance-maximization and covariance-minimization terms under structural constraints (see Eq. 9). Although the models used for self-supervised learning are not biologically plausible and address a different problem than source separation, they provide a useful point of comparison given the similarity of the objectives (and perhaps a common goal of encouraging sufficient spread in the domain).

A form of variance-covariance regularization has also been used in recent large-scale models such as Video JEPA [57]. Overall, extending *Predictive Entropy Maximization* to self-supervised or supervised learning tasks could be an interesting future direction.

E.11 Computational complexity of the proposed method

In this section, we characterize the computational cost of *Predictive Entropy Maximization*. Let n denote the number of sources, m the number of mixtures, and $\tau_{\max}$ the maximum number of fast neural-dynamics iterations used to compute the output for a single input sample.

The dominant cost comes from the fast inference step. For one neural-dynamics iteration, the feedforward prediction $\boldsymbol{W}(t)\boldsymbol{x}(t)$ requires $O(nm)$ operations, while the covariance-driven lateral term

in Equation 10 requires $O(n^2)$ operations. The domain-dependent output nonlinearity adds only $O(n)$ work, and for $\mathcal{B}_1$, $\mathcal{B}_{1,+}$, and Δ, the additional update of the shared inhibitory variable λ_L also remains $O(n)$. Therefore, one fast iteration costs

$$O(nm + n^2),$$

and the full inference stage costs

$$O\big(\tau_{\max}(nm + n^2)\big).$$

After the output has converged, the slow updates are cheaper. The feedforward update

$$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) + \alpha_W(t)\boldsymbol{e}(t)\boldsymbol{x}(t)^\top$$

costs $O(nm)$, the mean update costs $O(n)$, the diagonal variance update costs $O(n)$, and the off-diagonal covariance update costs $O(n^2)$. Hence the slow plasticity and statistics updates contribute

$$O(nm + n^2)$$

per sample.

Combining both stages, the overall worst-case cost per sample is

$$O\big((\tau_{\max} + 1)(nm + n^2)\big).$$

In the determined or overdetermined regime $m \geq n$, this simplifies to

$$O(\tau_{\max} mn),$$

since the recurrent inference stage dominates.

Thus, as in other biologically plausible recurrent BSS algorithms, the computational bottleneck is the iterative output inference rather than the synaptic updates. This is the same asymptotic order reported for the antisparse CorInfoMax network and for the determinant-maximization WSM networks, both of which are likewise dominated by recurrent output computation in digital simulations.

Wall-clock time. Table 5 reports CPU wall-clock runtimes for the experimental configurations used in this paper. Batch CorInfoMax uses 10^4 samples to limit the cost of 5000 full-batch updates and their associated linear solves; the other methods use 10^5 samples. Timings include initialization and, where applicable, just-in-time compilation and internal diagnostics. They depend on the implementation, learning rates, inner-loop iteration limits, and stopping criteria, with relative runtimes varying across source domains. The results therefore describe execution costs under the reported configurations. PEM hyperparameters are given in Appendix E.7; baseline settings are available in the released code.

Table 5: Recorded runtimes in seconds (mean $\pm$ sample standard deviation), for $m = 10$ mixtures and $n = 5$ sources. Antisparse entries pool 30 seeds over 11 correlation levels (330 runs per method); other domains pool 30 seeds over 6 input-SNR levels (180 runs). Sample budgets and algorithm configurations differ. A dash denotes an unreported setting.

Method	Samples	Antisparse	Sparse	Nonnegative sparse	Simplex
PEM	10^5	14.48 ± 1.41	14.60 ± 2.41	9.71 ± 1.58	17.40 ± 1.98
CorInfoMax (Online)	10^5	5.58 ± 0.42	31.21 ± 8.49	9.02 ± 0.86	–
CorInfoMax (Batch)	10^4	10.26 ± 0.88	23.38 ± 3.02	27.45 ± 2.32	21.19 ± 2.04
BSM	10^5	3.51 ± 0.62	–	–	–
ICA-InfoMax	10^5	6.21 ± 0.30	–	–	–

Compute resources. The implementation is CPU-compatible and does not require GPU acceleration or high-performance computing; the reported experiments can be run on a standard personal computer with sufficient memory. The runs reported here were performed with Python 3.12.12 on a Linux x86_64 machine (Linux 6.8.0-107-generic, glibc 2.39) using a single-node, single-task CPU Slurm job with 10GB requested memory.

E.12 Existing assets and licences

For sparse dictionary learning, we use the pre-whitened natural images associated with Olshausen and Field [4], distributed through Bruno Olshausen's Sparsenet website[5]; the distribution page does not specify a dataset licence. The Olivetti faces dataset [42] is credited to AT&T Laboratories Cambridge, as requested on its original distribution page[6], which does not specify a named licence. Fashion-MNIST [43] is distributed by Zalando Research under the MIT licence[7].

[5] https://www.rctn.org/bruno/sparsenet/

[6] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[7] https://github.com/zalandoresearch/fashion-mnist/blob/master/LICENSE

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main claims and contributions, including the Taylor approximation, predictive neural architecture, spectral error bounds, and empirical validation. These claims are supported by the model derivation in Section 3, the theoretical analysis in Appendix D, and the experiments in Section 4 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations paragraph in Section 5 discusses hyperparameter sensitivity, dependence on the number of observed mixtures, and computational scaling with fast neural-dynamics iterations. Supporting analyses are provided in Appendix E.7, Appendix E.8, and Appendix E.11.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theoretical assumptions and complete proofs for the Taylor remainder representation, spectral bounds, online specialization, and batch near-optimality result are provided in Appendix D. The update-rule derivations and descent property of the truncated direction are given in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 specifies the main synthetic and application settings, baselines, metrics, and number of runs. Appendix E provides the evaluation metric, confidence-interval formula, transform-domain details, hyperparameters, ablations, and supplementary experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce

the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: GitHub link to our Python implementation is given in the main text. The data used in the experiments are either synthetic or publicly available, with experimental protocols and hyperparameters documented in Appendix E.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes the main experimental setup, including mixture dimensions, source domains, sample size, noise levels, baselines, and performance metric. Appendix E.7 gives the simulation hyperparameters and implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Section 4 states that results are averaged over 30 independent realizations and that shaded bands denote 95% confidence intervals unless otherwise stated. Appendix E defines the mSNR metric and gives the confidence-interval computation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E.11 reports the per-sample computational complexity and the compute environment used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper presents foundational algorithmic work on biologically plausible blind source separation rather than a deployed system, human-facing application, or new dataset involving people. No direct societal-risk pathway is introduced.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk model, generative system, scraped dataset, or other asset requiring controlled release safeguards. The released asset is an implementation of the proposed algorithm.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits and cites the main external assets used in Section 4 and Appendix E, including the *librosa* audio sources and the Sparsenet natural-image dataset. Furthermore, Appendix E.12 documents the image-dataset sources, Fashion-MNIST's MIT licence, and the attribution requested for Olivetti faces. However, the Sparsenet and Olivetti distribution pages do not specify explicit dataset licences.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper releases a Python implementation with the supplementary material with instructions to reproduce the results. The algorithm, update rules, domain-specific realizations, and hyperparameters are documented in Section 3, Appendix C, and Appendix E.7.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or participant instructions. All experiments use synthetic data, public audio examples, or public natural-image datasets which we cite.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research or direct interaction with participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the core methodology, theory, or experiments. They were used only for editing (grammar, spelling, word choice), understanding technical concepts, and assisting with visualization/presentation of results for submission.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.