# Sparse Hyperparametric Itakura-Saito NMF via Bi-Level Optimization

Laura Selicato<sup>a,b</sup>, Flavia Esposito<sup>b,\*</sup>, Andersen Ang<sup>c</sup>, Nicoletta Del Buono<sup>b</sup>, Rafał Zdunek<sup>d</sup>

<sup>a</sup> Water Research Institute – National Research Council (IRSA-CNR), Italy
 <sup>b</sup> Department of Mathematics, University of Bari Aldo Moro, Italy
 <sup>c</sup> School of Electronics and Computer Science, University of Southampton, UK
 <sup>d</sup> Faculty of Electronics, Photonics, and Microsystems, Wrocław University of Science and Technology, Poland

#### Abstract

The selection of penalty hyperparameters is a critical aspect in Nonnegative Matrix Factorization (NMF), since these values control the trade-off between the reconstruction accuracy and the adherence to desired constraints. In this work, we focus on an NMF problem involving the Itakura-Saito (IS) divergence, effective for extracting low spectral density components from spectrograms of mixed signals, enhanced with sparsity constraints. We propose a new algorithm called SHINBO, which introduces a bi-level optimization framework to automatically and adaptively tune the row-dependent penalty hyperparameters, enhancing the ability of IS-NMF to isolate sparse, periodic signals against noise. Experimental results showed SHINBO ensures precise spectral decomposition and demonstrates superior performance in both synthetic and real-world applications. For the latter, SHINBO is particularly useful, as noninvasive vibration-based fault detection in rolling bearings, where the desired signal components often reside in high-frequency subbands but are obscured by stronger, spectrally broader noise. By addressing the critical issue of hyperparameter selection, SHINBO advances the state-ofthe-art in signal recovery for complex, noise-dominated environments.

Keywords: NMF, Itakura-Saito divergence, Hyperamater Optimization, Sparsity, Bi-level Optimization, Dynamical System

Email address: flavia.esposito@uniba.it (Flavia Esposito)

<sup>\*</sup>Corresponding author

#### 1. Introduction

Nonnegative Matrix Factorization (NMF) is a dimensionality reduction technique that approximates a nonnegative data matrix as the product of two (lower-dimensional) nonnegative matrices. A key challenge in NMF is setting penalty coefficients when additional constraints, such as sparsity or smoothness, are imposed [18]. These coefficients control the trade-off between reconstruction accuracy and constraint adherence, but their optimal values are highly dataset- and application-dependent, making the selection process non-trivial. For example, [6] proposes a variant of NMF that incorporates data-dependent penalties and introduces auxiliary constraints to enhance performance in tasks such as face recognition. Additionally, [20] presents multiplicative algorithms for NMF that enforce non-negativity and flow preservation constraints while introducing regularizations to ensure smoothness or sparsity. Finally, [11] adapts a minimization scheme for functions with nondifferentiable constraints, known as PALM, to solve NMF problems, yielding solutions that can be both smooth and sparse—two highly desirable properties. In this work, we rely on prior studies [8, 9] in which the penalized problem is reformulated in a general form, and a strategy is proposed to tune the penalty coefficient automatically.

Formally, let  $\boldsymbol{X} \in \mathbb{R}_{+}^{m \times n}$  with  $m, n \in \mathbb{N}$  be a data matrix, NMF aims to approximate it as the product of  $\boldsymbol{W} \in \mathbb{R}_{+}^{m \times r}$  and  $\boldsymbol{H} \in \mathbb{R}_{+}^{r \times n}$  with  $r \leq \min\{m, n\}$ , so that  $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$ . In our problem, we want to solve

$$(\boldsymbol{W}^*, \boldsymbol{H}^*) \in \underset{\boldsymbol{W} \geqslant \boldsymbol{0}, \boldsymbol{H} \geqslant \boldsymbol{0}}{\operatorname{argmin}} D_{\beta}(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}) + \mathcal{P}(\operatorname{Diag}(\boldsymbol{\lambda})\boldsymbol{H})$$
 (1)

with the objective function  $D_{\beta}(\cdot, \cdot)$  being the  $\beta$ -divergence [13, 19], assessing the quality of the reconstruction  $\boldsymbol{W}\boldsymbol{H}$  in fitting  $\boldsymbol{X}$ . We remind that the  $\beta$ -divergence for matrices is defined as  $D_{\beta}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i} \sum_{j} d_{\beta}(a_{ij}, b_{ij})$ , where the function  $d_{\beta}$  for each  $x, y \in \mathbb{R}$  is defined as

$$d_{\beta}(x,y) = \begin{cases} \frac{1}{\beta(\beta-1)} (x^{\beta} + (\beta-1)y^{\beta} - \beta xy^{\beta-1}) & \beta \in \mathbb{R} \setminus \{0,1\}; \\ x \log(\frac{x}{y}) - x + y & \beta = 1; \\ \frac{x}{y} - \log(\frac{x}{y}) - 1 & \beta = 0. \end{cases}$$

The function  $\mathscr{P}: \mathbb{R}^{r \times n} \to \mathbb{R}$  is a penalty term on  $\boldsymbol{H}$  that enforces a particular constraint. The vector  $\boldsymbol{\lambda} \in \mathbb{R}^r_+$  in (1) contains a nonnegative penalty hyperparameters  $[\lambda_1, \ldots, \lambda_r]$  associated to each row of  $\boldsymbol{H}$ .

In this study, we are interested in a special case of the  $\beta$ -divergence. When  $\beta = 0$ , the  $\beta$ -divergence boils down to the Itakura-Saito (IS) divergence which is a scale-invariant measure of dissimilarity, useful for extracting low spectral density components from spectrograms of mixed signals. In many practical applications, the desired components are located in a high frequency subband and are often masked by much stronger and spectrally wider noisy disturbances. This is the case of a noninvasive vibration-based fault detection in rolling bearings [23, 24, 25]. The vibrational diagnostic signals measured from faulty bearings on the laboratory test rig [16] are used in our study to validate the proposed algorithm.

The signal of interest (SOI) in this application is represented by a periodic and impulsive signal. The observed signal should contain the SOI and other perturbations, usually expressed by a strong independent and identically distributed (i.i.d.) Gaussian noise. Applying NMF to the spectrogram of the measured signal, we expect that the representative of the SOI will be expressed by one of the temporal components of NMF, i.e., one row of matrix  $\boldsymbol{H}$ . This component, which presents a periodic spiky signal, is obviously very sparse. The other components, which represent the noisy perturbations, should not be sparse. To enforce NMF to search for the desired component, we introduced the penalty for the rows of  $\boldsymbol{H}$  using the term  $\mathcal{P}(\mathrm{Diag}(\boldsymbol{\lambda})\boldsymbol{H})$  in (1). The proposed bi-level approach should perform data-driven adaptation of the hyperparameters (vector  $\boldsymbol{\lambda}$ ) to the desired nature of the estimated components.

Contribution. The contribution of this work is two folded.

- 1. New model. In this work, we present a new model for minimizing the Itakura-Saito divergence (Problem (1) with  $\beta=0$ ) while penalizing rows of  $\boldsymbol{H}$ . In particular
  - The penalty hyperparameter is not known in advance. This parameter is formulated as an optimization variable (in form of bilevel optimization model) and is solved by a bi-level optimization method.
  - The penalty hyperparameter is row-dependent. Note that Problem (1) is not the standard penalized NMF [18], which applies the

same penalty coefficient to all rows in the matrix  $\mathbf{H}$ . In Problem (1) each row of  $\mathbf{H}$  is penalized by its own penalty parameter.

2. **New algorithm**. For Problem (1), we present a new multiplicative update (see Equation (12)), and a way to automatically tune the penalty hyperparameter based on bi-level strategy (see details in Section 3).

**Paper Organization**. We introduce the problem and overall algorithm framework in Section 2. In Section 3 we first review the bi-level optimization, then we discuss the details of the bi-level approach proposed in this work for solving the Problem 1 in Section 4. Experimental results on synthetic and real datasets are presented in Section 5. We conclude the paper in Section 6, giving an outline of possible future directions.

**Notation**. The symbol  $\mathbf{0}$  denote the zero matrix, the symbol  $\mathbf{E}_{a \times b}$  is all-one matrix sizing  $a \times b$  with  $a, b \in \mathbb{N}$ . The notation  $\mathbf{v}$  denotes a column vector and the notation  $\underline{\mathbf{v}}$  means  $\mathbf{v}$  is a row vector.

On matrix,  $A^{\top}$  is the transpose of A, and  $A^2 = AA$ . The symbol  $A \odot B$  refers to the Hadamard (element-wise) product between A and B of conformal dimensions, and the symbol  $A \oslash B$  with  $B \neq 0$  refers to the Hadamard division, and we denote  $A^{\odot k}$  as the Hadamard power-k of A.

Given  $n \in \mathbb{N}$ , we denote  $[n] := \{1, 2, ..., n\}$ . The symbols  $k, t \in \mathbb{N}$  indicate iteration counters. The symbol  $\mathbf{A}^k$  refers to the variable  $\mathbf{A}$  at the iteration k,  $A_{ij}$  or  $a_{ij}$  is the (i, j)-th element of  $\mathbf{A}$ . Lastly  $\mathbf{A}_{i:}$  and  $\mathbf{A}_{:j}$  are the i-th and j-th row and column of  $\mathbf{A}$ , respectively.

## 2. The overall optimization framework of SHINBO

In this section, we discuss the main focus of the paper, Problem (1) and the overall framework of the proposed algorithm.

The optimization problem. We focus on Problem (1) with  $\beta = 0$ . As  $\mathcal{P}$  we chose a particular penalty function, effective in increasing sparsity, that is the diversity measure J [7, 9]. In a particular case, if the matrix is nonnegative, the diversity measure can be written in terms of the trace operator as

$$J(\mathbf{A}) = \sum_{i=1}^{n} \|\mathbf{A}_{i:}\|_{1}^{2} = Tr(\mathbf{A}\mathbf{E}\mathbf{A}^{\top}),$$
(2)

where  $\mathbf{E}$  is the squared matrix of ones. Thus, by properties of trace, Problem (1) becomes

$$(\boldsymbol{W}^*, \boldsymbol{H}^*, \boldsymbol{\lambda}^*) \in \underset{\substack{\boldsymbol{W} \geqslant \boldsymbol{0}, \boldsymbol{H} \geqslant \boldsymbol{0} \\ \boldsymbol{\lambda} \geqslant \boldsymbol{0}}}{\operatorname{argmin}} D_0(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}) + \operatorname{Tr}(\operatorname{Diag}(\boldsymbol{\lambda})^2 \boldsymbol{H} \boldsymbol{E} \boldsymbol{H}^\top).$$
 (3)

We remark that (3) is a nonconvex minimization problem, in which finding global minima is NP-hard, therefore we are interested in finding local minima for the triple  $(\mathbf{W}^*, \mathbf{H}^*, \boldsymbol{\lambda}^*)$ .

The proposed optimization algorithm SHINBO. We propose an algorithm, called SHINBO, to find a local minima for Problem (3) as follows.

```
Algorithm 1: SHINBO
```

```
1 Input: X \in \mathbb{R}_{+}^{m \times n} and factorization rank r.
  2 Initialize W^0 \in \mathbb{R}_+^{m \times r}, H^0 \in \mathbb{R}_+^{r \times n} and \lambda^0 \in \mathbb{R}_+^r
      for k = 1, 2, ... do
              \mathbf{W}^k = \text{update}(\mathbf{X}, \mathbf{W}^{k-1}, \mathbf{H}^{k-1})
                                                                                                            % classic MU-update
  4
              for l \in [r] do
  5
                    \underline{\boldsymbol{h}}_{l}^{k-1,0} = \underline{\boldsymbol{h}}_{l}^{k-1}, \lambda_{l}^{k-1,0} = \lambda_{l}^{k-1}
                                                                                                                         % initialization
  6

\underline{\boldsymbol{h}}_{l}^{k,t} = \text{update}(\underline{\boldsymbol{h}}_{l}^{k,t-1}, \boldsymbol{W}^{k}, \boldsymbol{X}, \lambda_{l}^{k}) \text{ as in (12)}

\frac{\partial \mathcal{R}}{\partial \lambda_{l}} \text{ as in (10)}

  7
  8
                                                                                                                      % hypergradient
  9
10
                     \lambda^{k,T} = \text{update}(\lambda^{k,T}, \nabla_{\lambda} \Re(\lambda)) % projected gradient update
11
              end
12
13 end
14 Return W, H, \lambda at the last iteration.
```

Note that SHINBO is composed by two main parts: one devoted to the update of W (reviewed in the following) and the other one based on bi-level strategy to optimize simultaneously on H and  $\lambda$ .

**Update on W.** The update of **W** can be done simply by the following Multiplicative Update [12] as  $\mathbf{W} = \mathbf{W} \odot (\mathbf{X}\mathbf{H}^{\top}) \oslash (\mathbf{W}\mathbf{H}\mathbf{H}^{\top})$ .

The next section introduces the update on  $\boldsymbol{H}$  and  $\boldsymbol{\lambda}$  by a bi-level method, which is the main contribution of the paper.

## 3. Bi-level Optimization for the subproblem

In this section, we discuss the steps for updating H and  $\lambda$  in Algorithm 1. Given a fixed  $W^k$ , we have the following optimization subproblem

$$(\boldsymbol{H}^*, \boldsymbol{\lambda}^*) \in \underset{\boldsymbol{H} \geqslant \mathbf{0}, \boldsymbol{\lambda} \geqslant \mathbf{0}}{\operatorname{argmin}} D_0(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}) + \operatorname{Tr}(\operatorname{Diag}(\boldsymbol{\lambda})^2 \boldsymbol{H} \boldsymbol{E} \boldsymbol{H}^\top).$$
 (4)

The core idea of this paper is to solve Problem (4) as a bi-level problem, in which we incorporate the problem of tuning hyperparameter  $\lambda$  simultaneously into the update of  $\mathbf{H}$ . To do so, first we review the general theory of bi-level optimization and its application to Problem (4).

Section organization and general overview of the approach. Under a fixed and given  $W^k$ , the goal is to obtain an updated version of H and  $\lambda$  that approximately solves Problem (4). The bi-level approach has the following steps:

- 1. First we replace the constrained optimization problem (4) by a bi-level optimization problem for  $(\underline{h}_l, \lambda_l)$ , with  $l \in [r]$ , see (5).
- 2. The inner problem (IP) in Problem (5) is then approximated by the solution of a dynamical system. See Problem (IVP- $\Phi$ ).
- 3. We then solve Problem (IVP- $\Phi$ ) to obtain a solution for  $\underline{\boldsymbol{h}}_l$ , and also the hypergradient at the last time point T. See (hypergrad).
- 4. Lastly we use the hypergradient to obtain the solution  $\lambda$  by a gradient descent approach. See (6).

We now proceed to discuss each of the steps below.

1. Bi-level formulation. In Algorithm 1, the update of H and  $\lambda$  is performed in r steps, where each step is aimed to update the l-th component  $(\underline{h}_l, \lambda_l)$ . This is achieved by solving the following bi-level problem

$$\min_{\lambda_{l} \geq 0} \left\{ \begin{array}{ll} \boldsymbol{r}(\lambda_{l}) = & \inf_{\underline{\boldsymbol{h}}_{l}(\lambda_{l})} D_{2}(\boldsymbol{X}, \boldsymbol{R} + \boldsymbol{w}_{l}\underline{\boldsymbol{h}}_{l}(\lambda_{l})) & \text{(OP)} \\ & \text{s.t. } \underline{\boldsymbol{h}}_{l}(\lambda_{l}) \in \operatorname*{argmin}_{\boldsymbol{u} \in \mathbb{R}^{n}_{+}} D_{0}(\boldsymbol{X}, \boldsymbol{R} + \boldsymbol{w}_{l}\underline{\boldsymbol{u}}) + \lambda_{l}^{2} \|\underline{\boldsymbol{u}}\|_{1}^{2} & \text{(IP)} \end{array} \right\}, \tag{5}$$

where the matrix  $\mathbf{R}$  is the residual obtained isolating in  $\mathbf{W}\mathbf{H}$  the l-th component of  $\mathbf{w}_l \in \mathbb{R}^m$  (column of  $\mathbf{W}$ ) and  $\underline{\mathbf{h}}_l \in \mathbb{R}^n$  (row of  $\mathbf{H}$ ), which is

$$oldsymbol{R} = oldsymbol{X} - \sum_{j 
eq \ell} oldsymbol{w}_j oldsymbol{h}_j.$$

The function  $r : \mathbb{R} \to \mathbb{R}$  is called Response function of the outer problem related to  $\underline{h}_l$ . In the outer problem, the objective  $D_2$  is the  $\beta$ -divergence with  $\beta = 2$ , which is the Frobenius norm. The inner problem is represented by the  $\beta$ -divergence with  $\beta = 0$ , which is the Itakura-Saito divergence  $D_0$ , regularized by the row-wise diversity measure defined in (2).

- **Remark 1.** Note that the squared  $\ell_1$ -norm in the inner problem on  $\mathbf{u}$  can be seen as a non-smooth regularization term and therefore proximal gradient descent can be applied on  $\mathbf{u}$  (see [10] and [1, Lemma 6.70]), however, their approach is applied to convex objective function, where here in (5) the objective function (the IS-divergence) is possibly nonconvex [12] thus proximal gradient descent do not have convergence guarantee.
- 2. Dynamical system approach on H. An approach to solve the bilevel Problem (5) over H is to replace the inner problem with a dynamical system [14, 15, 22] and compute an approximation solution.

We now omit the k index in  $\underline{\boldsymbol{h}}_{l}^{k,t}$ , due to the fact that the update focuses on the iteration over t under a constant k.

Given  $\underline{\boldsymbol{h}}_{l}^{0}$  (which depends implicitly on  $\lambda_{l}$ ), we build a dynamical system (IVP- $\Phi$ ) in the form of a discrete initial value problem as

$$\begin{cases} \underline{\boldsymbol{h}}_{l}^{t} = \Phi_{t}(\underline{\boldsymbol{h}}_{l}^{t-1}, \lambda_{l}), & t \in [T] \\ \underline{\boldsymbol{h}}_{l}^{0} = \Phi_{0}(\lambda_{l}) \end{cases}$$
(IVP- $\Phi$ )

where  $\Phi_t : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^r$  is a smooth map for  $t \in [T]$ .

The idea of the bi-level optimization strategy is to use the IVP- $\Phi$  to approximate the solution of the Problem (5). We do so by solving the following minimization

$$\lambda_l^* = \underset{\lambda_l}{\operatorname{argmin}} \quad \boldsymbol{r}(\lambda_l) \\
\text{s.t.} \quad \underline{\boldsymbol{h}}_l^t = \Phi_t(\underline{\boldsymbol{h}}_l^{t-1}, \lambda_l) \quad \text{for } t \in [T]$$
(6)

in which we approximate the solution of the inner problem with the solution of the dynamical system. This is possible because problem (6) satisfies the existence and convergence theorems as proved in [8].

As a preview, we will derive  $\Phi_t(\underline{\boldsymbol{h}}_l^{t-1}, \lambda_l)$  in Section 4.

3. Hypergradient. To find the solution  $\lambda^*$  for Problem (4), we solve the problem formed by joining all the Response functions  $r(\lambda_l)$  in (5) as

$$\underset{\boldsymbol{\lambda} \in \mathbb{R}_{+}^{r}}{\operatorname{argmin}} \left\{ \Re(\boldsymbol{\lambda}) := \sum_{j} r(\lambda_{l}) \right\}. \tag{7}$$

We would like to compute the hypergradient, i.e. the gradient of  $\mathcal{R}$  with respect to  $\lambda$ , in order to use a gradient descent approach on  $\lambda$ . The hypergradient  $\nabla_{\lambda}\mathcal{R}$ , by using chain rule, is

$$\frac{\partial \mathcal{R}}{\partial \lambda_l} = \frac{\partial \mathbf{r}}{\partial \lambda_l} + \frac{\partial \mathbf{r}}{\partial \mathbf{h}_l^T} \cdot \frac{d\mathbf{h}_l^T}{d\lambda_l}, \quad l \in [r].$$
 (hypergrad)

Note that  $\underline{\boldsymbol{h}}_{l}^{T}$  denotes the row-vector  $\underline{\boldsymbol{h}}_{l}$  at the time T.

It is well known that the computation of the hypergradient can be done using Reverse-Mode Differentiation (RMD) or Forward-Mode Differentiation (FMD). Since RMD requires storing specific variables across all iterations and indices in memory, in this work we use FMD, making it more suitable for scenarios where the total quantity of interest is small. For details we refer the reader to [8, 15].

Forward-Mode. FMD computes the differentiation in (hypergrad) using the chain rule. Function  $\Phi_t$  for  $t \in [T]$  depends on  $\lambda_l$  explicit and on  $\underline{\boldsymbol{h}}_l^{t-1}$  implicitly, then we have the derivative

$$\frac{d\underline{\boldsymbol{h}}_{l}^{t}}{d\lambda_{l}} = \frac{\partial \Phi_{t}(\underline{\boldsymbol{h}}_{l}^{t-1}, \lambda_{l})}{\partial \boldsymbol{h}_{l}^{t-1}} \cdot \frac{d\underline{\boldsymbol{h}}_{l}^{t-1}}{d\lambda_{l}} + \frac{\partial \Phi_{t}(\underline{\boldsymbol{h}}_{l}^{t-1}, \lambda_{l})}{\partial \lambda_{l}}.$$

Let  $\mathbf{s}^t = \frac{d\underline{\mathbf{h}}_l^t}{d\lambda_l}$ , then each FMD iterate behaves as

$$\begin{cases} s^t = A_t s^{t-1} + b_t, & t \in [T] \\ s^0 = b_0 \end{cases}$$
(8)

where 
$$\mathbf{A}_t = \frac{\partial \Phi_t(\underline{\mathbf{h}}_l^{t-1}, \lambda_l)}{\partial \underline{\mathbf{h}}_l^{t-1}} \in \mathbb{R}^{n \times n}$$
 and  $\mathbf{b}_t = \frac{\partial \Phi_t(\underline{\mathbf{h}}_l^{t-1}, \lambda_l)}{\partial \lambda_l} \in \mathbb{R}^n$ . Now

(hypergrad) can be expressed as

$$\frac{\partial \mathcal{R}}{\partial \lambda_l} = \langle \underline{\boldsymbol{g}}^T, \boldsymbol{s}^T \rangle \in \mathbb{R} \quad \text{where} \quad \underline{\boldsymbol{g}}^T = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{h}_l} \in \mathbb{R}^n.$$
 (9)

We note that  $\underline{g}^T$  is denoting the row-vector  $\underline{g}$  at the time T. Lastly, the solution of Problem (8) solves the following equation

$$\frac{\partial \mathcal{R}}{\partial \lambda_l} = \frac{\partial \boldsymbol{r}}{\partial \underline{\boldsymbol{h}}_l^T} \left( \boldsymbol{b}_T + \sum_{t=0}^{T-1} \left( \prod_{s=t+1}^T \boldsymbol{A}_s \right) \boldsymbol{b}_t \right). \tag{10}$$

4. Update of  $\lambda$ . Given all the components  $\underline{h}_1^T, \underline{h}_2^T, \dots, \underline{h}_r^T$  at the last time point of the dynamic system and the hypergradient (discussed previously), we update  $\lambda$  with a projected gradient descent approach as  $\lambda = [\lambda - \alpha \nabla_{\lambda} \Re(\lambda)]_{+}$ , for a pre-defined stepsize  $\alpha > 0$ .

### 4. Derivation of the algorithm SHINBO

We now introduce the overall bi-level optimization approach, discussed in the previous section, to the Subproblem (4). We use the method of partial Lagrangian multiplier, which is applied only on  $\boldsymbol{H}$ . First, let  $\boldsymbol{\Psi} \in \mathbb{R}_{+}^{r \times n}$  be the matrix of Lagrangian multipliers associated to the nonnegative constraints of  $\boldsymbol{H}$ , then the expression of the (partial) Lagrangian of Subproblem (4) is

$$\mathcal{L}(\boldsymbol{H}) = D_0(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}) + \text{Tr}(\text{Diag}(\boldsymbol{\lambda})^2 \boldsymbol{H}\boldsymbol{E}\boldsymbol{H}^\top) + \text{Tr}(\boldsymbol{\Psi}\boldsymbol{H}^\top).$$

Recall that  $M^{\odot 2}$  denotes the Hadarmard power of M, then the partial derivative of  $\mathcal L$  with respect to H is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{H}} = \boldsymbol{W}^{\top} \Big( (\boldsymbol{W} \boldsymbol{H})^{\odot - 2} (\boldsymbol{W} \boldsymbol{H} - \boldsymbol{X}) \Big) + 2 \mathrm{Diag}(\boldsymbol{\lambda})^{2} \boldsymbol{H} \boldsymbol{E} + \boldsymbol{\Psi}.$$

Now denote  $H_{ij}$  the (i, j)-th element of the matrix  $\mathbf{H}$ , and recall that  $\odot$  is the Hadarmard product, then by the complementary slackness  $\Psi_{ij}H_{ij} = 0$  in the KKT conditions, we get

$$\left[\boldsymbol{W}^{\top}\Big((\boldsymbol{W}\boldsymbol{H})^{\odot-2}\odot(\boldsymbol{W}\boldsymbol{H}-\boldsymbol{X})\Big)\right]_{ij}H_{ij}+2\Big[\mathrm{Diag}(\boldsymbol{\lambda})^2\boldsymbol{H}\boldsymbol{E}\Big]_{ij}H_{ij}+\Psi_{ij}H_{ij}=0.$$

These equations lead to the multiplicative update

$$H_{ij} = H_{ij} \left[ \mathbf{W}^{\top} (\mathbf{W} \mathbf{H})^{\odot -2} \mathbf{X} \right]_{ij} / \left[ \mathbf{W}^{\top} (\mathbf{W} \mathbf{H})^{\odot -1} + 2 \operatorname{Diag}(\boldsymbol{\lambda})^{2} \mathbf{H} \mathbf{E} \right]_{ij}, (11)$$

where the division is performed element-wise as the Hadamard division  $\oslash$ . By fixing  $l \in \{1, \ldots, r\}$ , the update in row-wise format for the l-th row of  $\boldsymbol{H}$  can be rewritten equivalently as

$$\underline{\boldsymbol{h}}_{l}^{t} = \underline{\boldsymbol{h}}_{l}^{t-1} \odot \frac{\left(\boldsymbol{W}^{\top} (\boldsymbol{W} \boldsymbol{H})^{\odot - 2} \boldsymbol{X}\right)_{l:}^{t-1}}{\left(\boldsymbol{W}^{\top} \odot (\boldsymbol{W} \boldsymbol{H})^{\odot - 1}\right)_{l:} + 2(\lambda_{l}^{t-1})^{2} \|\underline{\boldsymbol{h}}_{l}^{t-1}\|_{1} \boldsymbol{E}_{l:}}.$$
(12)

We remark that Equation (12) is one of the main contribution of this paper.

Remark 2. The key idea in this methodology is that  $\Phi_t(\underline{h}_l^{t-1}, \lambda_l)$  in the dynamical system (IVP- $\Phi$ ) is the right hand side of the update (12).

## 4.1. The implementation of the bi-level approach in SHINBO

Following the procedure and the discussion in Section 3, we consider  $\Phi_t(\underline{\boldsymbol{h}}_l^{t-1}, \lambda_l)$  in the dynamical system (IVP- $\Phi$ ) (represented by the multiplicative update (12)) and we compute  $\boldsymbol{A}^t = \frac{\partial \Phi_t}{\partial \underline{\boldsymbol{h}}_l^{t-1}}$  and  $\boldsymbol{b}^t = \frac{\partial \Phi_t}{\partial \lambda_l}$  for  $t \in [T]$  required for the FMD.

• The computation of  $m{A}^t = rac{\partial \Phi_t}{\partial m{h}_l^{t-1}}$  gives a diagonal matrix

$$A_{jj}^{t} = \frac{N_{lj}}{D_{lj}} - h_{lj} \left( \frac{2\sum_{i}^{n} w_{il}^{2} \frac{x_{ij}}{(\boldsymbol{W}\boldsymbol{H})_{ij}^{3}} D_{lj} - N_{lj} \sum_{i}^{n} \frac{w_{il}^{2}}{(\boldsymbol{W}\boldsymbol{H})_{lj}^{2}} + 2\lambda_{l}^{2} N_{lj}}{D_{lj}^{2}} \right),$$

where the derivative is computed with respect to the (l, j)-th element of  $\mathbf{H}$ , and

$$N_{lj} = \left( \boldsymbol{W}^{\top} (\boldsymbol{W} \boldsymbol{H})^{\odot - 2} \boldsymbol{X} \right)_{lj}, \ D_{lj} = \left( \boldsymbol{W}^{\top} (\boldsymbol{W} \boldsymbol{H})^{\odot - 1} + 2 \mathrm{Diag}(\boldsymbol{\lambda})^2 \boldsymbol{H} \boldsymbol{E} \right)_{lj}.$$

• The vector 
$$\boldsymbol{b}_t$$
 is given by  $\frac{\partial \Phi_t}{\partial \lambda_l} = -4\lambda_l h_{lj}^2 \frac{N_{lj}}{D_{lj}^2}$ .

Finally, having chosen as an outer problem the Frobenius norm,  $\underline{\boldsymbol{g}}^T$  in (9) for computing (hypergrad) is  $\underline{\boldsymbol{g}}^T = -2\boldsymbol{w}_l^{\top}(\boldsymbol{X} - \boldsymbol{R} - \boldsymbol{w}_l\boldsymbol{h}_l)$ .

## 5. Experimental Results

In this section, we present the numerical results of comparing the proposed algorithm SHINBO with the multiplicative update (MU) algorithm [12]. We also compare SHINBO with the penalized MU, which is the update (12) under a fixed penalty hyperparameter  $\lambda$  with  $\lambda_1 = \lambda_2 = \ldots = \lambda_r$  for every row of  $\mathbf{H}$ . We summarize their difference in the table below.

Table 1: The method compared in the experiment.

Algorithm	$\lambda$	$\lambda$ setting
MU [12]	0	constant
Penalized MU	0.1	constant for each column
Penalized MU	0.5	constant for each column
SHINBO	not required	automatically optimized,
		different penalization for each column

**Datasets**. We evaluate the algorithms on two datasets: one generated synthetically, and the other one obtained by processing real vibrational signals measured in a laboratory condition from damage rolling bearing elements.

- The synthetic dataset is generated starting from a full-rank decomposition  $X \approx WH$ , where the factor matrix W contains 10% nonzeros in the mr entries and the factor matrix H contains 70% nonzeros in the rn entries. In this dataset, we have (m, n, r) = (100, 7, 3).
- The vibrational signals were measured on the test rig presented in Fig. 1. The platform is equipped with an electric motor, gearbox, couplings, and two bearings. One of the latter was deliberately damaged. Diagnostic signals were measured with the accelerometer (KISTLER Model 8702B500), stacked horizontally to the shaft bearing. The 40-second-long vibration signal was recorded with the sampling rate of 50 kHz. For easier visualization, one second excerpt was selected, and

then transformed to a spectrogram using the 128-window length, 100 sample overlapping, and 512 frequency bins. The raw observed signal and its spectrogram are shown in Fig. 2. Assuming r = 4 (four main components of the mixed signal), we have (m, n, r) = (257, 1782, 4).

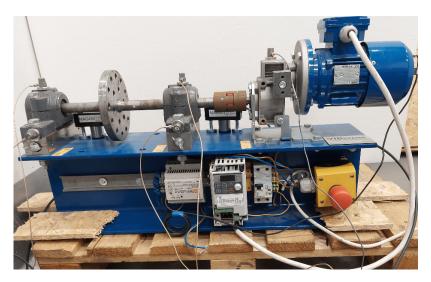


Figure 1: The test rig used in the experiment [16].

### 5.1. Experimental setup

**Initialization**. To ensure a fair comparison, all algorithms were initialized using the same initial factor matrices, which were generated from an unpenalized MU-based NMF warm start, that is initialized via nonnegative double singular value decomposition (NNDSVD) of the matrix  $\boldsymbol{X}$  [4].

**Simulation**. We perform 100 Monte Carlo simulations, where each run uses a different random data matrix. At the start of each run, the initial value of  $\lambda^0$  was selected randomly following a uniform distribution  $\mathcal{U}[0,1]$ .

**Normalization**. We perform a normalization step on each column of W,  $w_k$  for all k = 1, ..., r:

$$w_k = w_k / \max(w_k);$$
  $\underline{h}_k = \underline{h}_k * \max(w_k).$ 

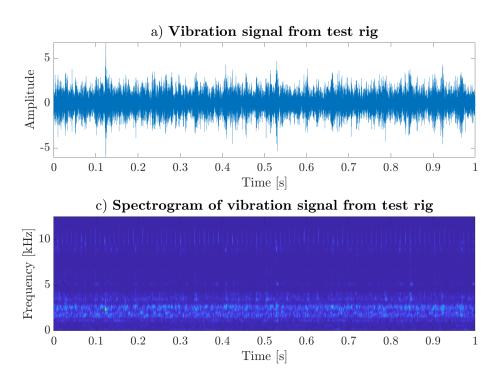


Figure 2: Recorded vibration signal and its spectrogram.

**Termination**. All the algorithms were allowed to run a maximum number of iterations of 500 for the synthetic dataset and 100 for the real one, with an early termination tolerance of  $10^{-6}$  on the relative fitting error using the IS-divergence  $D_0$ , defined as

$$|D_0(\mathbf{X}, \mathbf{W}^{k+1}\mathbf{H}^{k+1}) - D_0(\mathbf{X}, \mathbf{W}^k\mathbf{H}^k)| / |D_0(\mathbf{X}, \mathbf{W}^k\mathbf{H}^k)| \le 10^{-6},$$

where k indicates the iteration of the outer loop of the algorithm. For the inner loop of the bi-level approach on iteration counter t, we stop at the maximum number of (inner) T=4 iterations.

**Evaluation**. We evaluate the algorithms according to different criteria on synthetic and real-world datasets. For both datasets, we plot the convergence of the algorithms by looking at the behavior of the Response function.

For the synthetic dataset, we evaluate the quality of the factorization with respect to the identification problem for the synthetic dataset using the Signal-to-Interference Ratio (SIR) measure [5] between the estimated signals and the true signals. This is a log-scale similarity measure (expressed in

decibels), often used in signal processing applications, and its higher value indicates a higher similarity level. To highlight the effectiveness of the proposed method, we conduct a statistical comparison of SIR values across algorithms using the Kruskal-Wallis test, followed by a post-hoc multiple comparison based on the Mann-Whitney test [21] with Benjamini-Hochberg (BH) correction [2], maintaining a significance level of  $\alpha = 0.05$ . We also investigate the sparsity of the results obtained for the synthetic dataset using the following measure of sparsity for a generic matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ :

$$Sp(\mathbf{A}) := (1 - \|\mathbf{A} - 10^{-6}\mathbf{E}\|_{0}/mn)100\%,$$

where  $\|\cdot\|_0$  denotes the number of non-zeros elements. To demonstrate the advantages of the proposed method, we analyze sparsity values across algorithms using the same statistical test as for the evaluation of the SIR (Kruskal-Wallis test, post-hoc Mann-Whitney test with BH and  $\alpha = 0.05$ ).

On real-world dataset we check the goodness of the factorization by quantifying the impulsive and cyclic behavior of the signal under analysis, using a modified envelope spectrum based indicator (ENVSI) [17, 3] on time profiles. ENVSI can be expressed as spectrum based indicator (SBI), defined as:

$$SBI := \sum_{i=1}^{M_1} AIS_i^2 / \sum_{k=1}^{M_2} S_k^2,$$

where  $AIS_i$  is the magnitude of the *i*-th harmonic of the estimated signal in the frequency domain,  $S_k$  is the magnitude in the *k*-th frequency bin in the spectrum of the time profile,  $M_1$  is the number of harmonics to be analyzed (assuming a periodic signal), and  $M_2$  is the number of frequency bins to calculate the total energy.

SBI is zero if there are no impulsive components in the time profile, whereas a larger SBI occurs when the impulses in the time profile are stronger (which corresponds to the amplitudes in the spectrum) and the noise is weaker. In the experiments, the number of harmonics  $M_1$  is set to 6, and it was experimentally found to be sufficient for demonstrating the impulsive and period behavior of the SOI representing time profile in the analyzed application. To demonstrate the superiority of the proposed method, we statistically compare ENVSI values across algorithms using a Kruskall-Wallis test and a post-hoc multiple comparison based on Mann-Whitney test with a BH correction, with a statistical significance level set at  $\alpha = 0.05$ .

**Computer.** All the experiments were conducted in MATLAB 2021a and executed on a machine with an i7 octa-core processor and 16GB of RAM.

## 5.2. Results on synthetic dataset

Refers to Figure 3, the proposed algorithm demonstrates superior performance in terms of convergence rate of Response function compared to the other methods. We remark that, despite SHINBO exhibited similar and rapid convergence as unpenalized MU for the first 450 iterations (on average), the obtained decompositions have a different SIR value. See the discussion below.

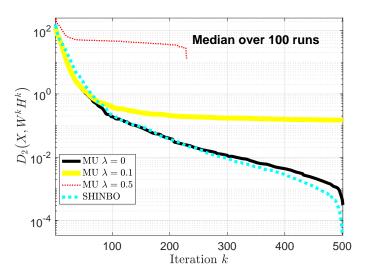


Figure 3: Behavior of the Response function (outer problem) for the synthetic dataset.

Refers to Figure 4, we can see that SHINBO obtain the best SIR values on both matrices and the higher sparsity on  $\boldsymbol{H}$  compared to the other methods. On average, SHINBO achieved 10% better SIR and 5% better sparsity on  $\boldsymbol{H}$  than other algorithms.

These results are also confirmed by the statistical comparisons. The Kruskall-Wallis tests present p-values lesser that  $10^{-14}$  and the details of the pair-wise comparison with a BH comparison are presented in the tables 2 and 3.

#### 5.3. Results on real-world dataset

As shown in Figure 5, the proposed algorithm outperforms other methods in terms of the convergence of the Response function. From Figure 6, we

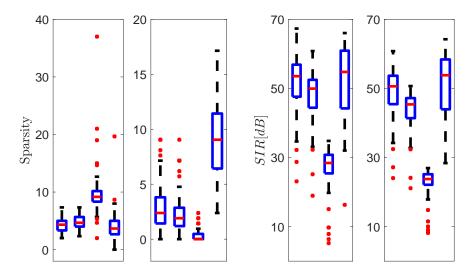


Figure 4: Results of sparsity and SIR on the factor matrices on the synthetic dataset. Left two columns are the sparsity of  $\boldsymbol{W}$  (left) and  $\boldsymbol{H}$  (right) of the four algorithms: from the left to right are MU with  $\lambda=0$ , MU with  $\lambda=0.1$ , MU with  $\lambda=0.5$  and SHINBO. The right two columns are the SIR of  $\boldsymbol{W}$  (left) and  $\boldsymbol{H}$  (right) of the four algorithms: from the left to right are MU with  $\lambda=0$ , MU with  $\lambda=0.1$ , MU with  $\lambda=0.5$  and SHINBO.

Table 2: p-values results of pairwise comparisons for SIR coefficients on (W, H). In the table  $\epsilon = 10^{-16}$ .

	MU	MU $\lambda = 0.1$	MU $\lambda = 0.5$
$MU \lambda = 0.1$	$8.1 \cdot 10^{-5}, 6.3 \cdot 10^{-10}$	-	-
MU $\lambda = 0.5$	$< 2\epsilon, < 2\epsilon$	$< 2\epsilon, < 2\epsilon$	-
SHINBO	0.14,0.0042	$5.4 \cdot 10^{-5}, 1.2 \cdot 10^{-9}$	$< 2\epsilon, < 2\epsilon$

Table 3: p-values results of pairwise comparisons for sparsity coefficients on (W, H).

	MU	MU $\lambda = 0.1$	MU $\lambda = 0.5$
$MU \lambda = 0.1$	0.00036, 0.0075	-	-
MU $\lambda = 0.5$	$< 2\epsilon, < 2\epsilon$	$< 2\epsilon, < 2\epsilon$	-
SHINBO	$0.023, < 2\epsilon$	$3.12 \cdot 10^{-6}, < 2\epsilon$	$< 2\epsilon, < 2\epsilon$

observe that SHINBO achieves the highest ENVSI on  $\boldsymbol{H}$ . A higher value of the ENVSI score (above 0.77) and a lower number of outliers confirm that the proposed algorithm finds the SOI in the observed diagnostic signal, which is consistent with our prior knowledge about the physical state of the tested rolling bearing.

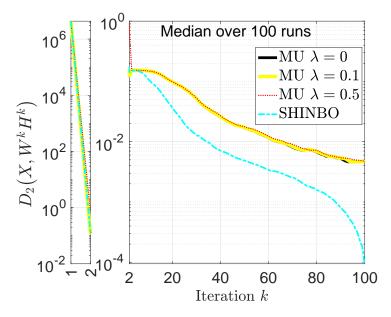


Figure 5: Behavior of the Response function (outer problem) for the real dataset.

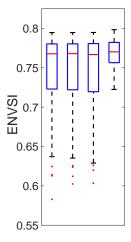


Figure 6: The ENVSI score of  $\boldsymbol{H}$  of the four algorithms: from the left to right are MU with  $\lambda=0$ , MU with  $\lambda=0.1$ , MU with  $\lambda=0.5$  and SHINBO.

These results are also confirmed by the statistical comparisons. The Kruskall-Wallis tests present p-values lesser than the fixed significant level (0.04) and the details of the pair-wise comparison with a BH comparison are presented in Table 4.

The concept is that SHINBO is designed to effectively identify which

Table 4: p-values results of pairwise comparisons for ENVSI coefficients.

		MU $\lambda = 0.1$	MU $\lambda = 0.5$
MU $\lambda = 0.1$	0.98	-	-
MU $\lambda = 0.5$	0.98	0.98	-
$\begin{array}{c} \text{MU } \lambda = 0.1 \\ \text{MU } \lambda = 0.5 \\ \text{SHINBO} \end{array}$	0.04	0.04	0.04

components of  $\lambda$  are associated with noise and which are linked to the true sparse signal. The goal is to apply greater penalization to certain components, thereby filtering out the noise while preserving the component representing the SOI. The results indicate that SHINBO performs very well in this context, as it successfully isolates and preserves the meaningful signal while suppressing irrelevant noise.

#### 6. Conclusion

In this work, we addressed the critical challenge of selecting penalty hyperparameters in NMF by introducing SHINBO, a novel algorithm that employs a bi-level optimization framework to adaptively tune row-dependent penalties. By focusing on the IS divergence, SHINBO proves highly effective for extracting low spectral density components in spectrograms, particularly in the presence of noise. The ability of the algorithm to enforce sparsity constraints and dynamically adjust penalties ensures a more precise separation of meaningful signals from noisy disturbances.

Through experiments on both synthetic and real-world datasets, SHINBO demonstrated its superior performance compared to traditional NMF methods. For real-world applications, such as noninvasive vibration-based fault detection in rolling bearings, SHINBO excelled at isolating sparse, periodic signal components in high-frequency subbands, even when heavily masked by broader noise.

The results highlight SHINBO's potential to significantly enhance signal recovery in complex, noise-dominated environments. By tackling the hyperparameter selection problem with an adaptive, data-driven approach, SHINBO not only advances the field of NMF but also provides a robust tool for applications requiring precise spectral decomposition and noise suppression. Future work will explore the scalability of SHINBO for larger datasets and its adaptability to other domains with similar challenges.

## Acknowledgment

The authors would like to express their sincere appreciation to Professor Radoslaw Zimroz from Faculty of Geoengineering, Mining and Geology at Wroclaw University of Science and Technology, for providing the real data collected in his laboratory.

N.D.B., F.E., L.S. are members of the Gruppo Nazionale Calcolo Scientifico - Istituto Nazionale di Alta Matematica (GNCS-INdAM).

## **Funding**

N.D.B., F.E., and L.S. are partially supported by "INdAM - GNCS Project", CUP: E53C24001950001.

N.D.B. and F.E. are supported by Piano Nazionale di Ripresa e Resilienza (PNRR), Missione 4 "Istruzione e Ricerca"-Componente C2 Investimento 1.1, "Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale", Progetto PRIN-2022 PNRR, P2022BLN38, Computational approaches for the integration of multi-omics data. CUP: H53D23008870001.

### Authors contribution

All authors contributed equally to this work.

#### Conflict of interest

The authors have no relevant financial interest to disclose.

## References

- [1] A. Beck. First-order methods in optimization. SIAM, 2017.
- [2] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.
- [3] Y. Berrouche, G. Vashishtha, S. Chauhan, and R. Zimroz. Local damage detection in rolling element bearings based on a single ensemble empirical mode decomposition. *Knowledge-Based Systems*, 301:112265, 2024.

- [4] C. Boutsidis and E. Gallopoulos. Svd based initialization: A head start for nonnegative matrix factorization. *Pattern recognition*, 41(4):1350–1362, 2008.
- [5] A. Cichocki, R. Zdunek, and A. H. Phan. Amari Si. Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons, 2009.
- [6] M. Corsetti and E. Fokoué. Nonnegative matrix factorization with zell-ner penalty. arXiv preprint arXiv:2012.03889, 2020.
- [7] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on signal processing*, 53(7):2477–2488, 2005.
- [8] N. Del Buono, F. Esposito, L. Selicato, and R. Zdunek. Bi-level algorithm for optimizing hyperparameters in penalized nonnegative matrix factorization. *Applied Mathematics and Computation*, 457:128184, 2023.
- [9] N. Del Buono, F. Esposito, L. Selicato, and R. Zdunek. Penalty hyper-parameter optimization with diversity measure for nonnegative low-rank approximation. *Applied Numerical Mathematics*, 208:189–204, 2025.
- [10] T. Evgeniou, M. Pontil, D. Spinellis, N. Nassuphis, J. Suykens, M. Signoretto, and A. Argyriou. Regularized robust portfolio estimation. Regularization, optimization, kernels, and support vector machines, Chapman & Hall/CRC Mach. Learn. Pattern Recogn. Ser., CRC Press, Boca Raton, FL, pages 237–256, 2015.
- [11] R. Fabregat, N. Pustelnik, P. Gonçalves, and P. Borgnat. Solving nmf with smoothness and sparsity constraints using palm. arXiv preprint arXiv:1910.14576, 2019.
- [12] C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.
- [13] C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the  $\beta$ -divergence. Neural computation, 23(9):2421–2456, 2011.

- [14] L. Franceschi. A Unified Framework for Gradient-based Hyperparameter Optimization and Meta-learning. PhD thesis, UCL (University College London), 2021.
- [15] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil. Forward and reverse gradient-based hyperparameter optimization. In *ICML*, pages 1165–1173. PMLR, 2017.
- [16] M. Gabor, R. Zdunek, R. Zimroz, J. Wodecki, and A. Wylomanska. Non-negative tensor factorization for vibration-based local damage detection. *Mechanical Systems and Signal Processing*, 198:110430, 2023.
- [17] M. Gabor, R. Zdunek, R. Zimroz, and A. Wylomanska. Bearing damage detection with orthogonal and nonnegative low-rank feature extraction. *IEEE Transactions on Industrial Informatics*, 2023.
- [18] N. Gillis. Nonnegative matrix factorization. SIAM, 2020.
- [19] R. Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural computation*, 19(3):780–791, 2007.
- [20] H. Lantéri, C. Theys, and C. Richard. Nonnegative matrix factorization with regularization and sparsity-enforcing terms. In 2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pages 97–100. IEEE, 2011.
- [21] E. L. Lehmann, J. P. Romano, and G. Casella. *Testing statistical hypotheses*, volume 3. Springer, 1986.
- [22] D. Maclaurin, D. Duvenaud, and R. Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proc. of ICML*, pages 2113–2122, 2015.
- [23] J. Wodecki, P. Kruczek, A. Bartkowiak, R. Zimroz, and A. Wyłomańska. Novel method of informative frequency band selection for vibration signal using nonnegative matrix factorization of spectrogram matrix. Mechanical systems and signal processing, 130:585–596, 2019.
- [24] J. Wodecki, A. Michalak, R. Zimroz, T. Barszcz, and A. Wyłomańska. Impulsive source separation using combination of nonnegative matrix factorization of bi-frequency map, spatial denoising and monte carlo simulation. *Mechanical Systems and Signal Processing*, 127:89–101, 2019.

[25] J. Wodecki, A. Michalak, R. Zimroz, and A. Wyłomańska. Separation of multiple local-damage-related components from vibration data using nonnegative matrix factorization and multichannel data fusion. *Mechanical Systems and Signal Processing*, 145:106954, 2020.