# Accelerated Deep Nonlinear Dictionary Learning

Benying Tan, Jie Lin, Yang Qin, Shuxue Ding, and Yujie Li*

School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin, 541000, China
*Corresponding author: yujieli@guet.edu.cn

**Abstract.** Most of the existing dictionary learning models are based on linearly learned dictionaries, which have weak performance in nonlinear signal representation, thus driving a research boom in nonlinear dictionary learning (NLDL). In this paper, we propose a deep nonlinear dictionary learning model for dictionaries and coefficients with full-layer sparse regularizations to access deep latent information. We apply $\ell_1$ regularization on the model to improve the efficiency of extracting key features hierarchically. We investigated the proposed algorithm using the Lifted Proximal Operator Machine (LPOM), by which a nonlinear model is transformed into a linear convex optimization problem to be solved. Then Nesterov's acceleration is introduced to speed up the convergence, called Accelerated DNLDL_$\ell_1$. We verify the feasibility of the proposed algorithm through numerical and application experiments. The results show that the acceleration scheme improves the convergence speed of the algorithm, and the proposed method has excellent performance on image classification and image denoising tasks.

**Keywords:** Deep nonlinear dictionary learning · Lifted Proximal Operator Machine · Nesterov's acceleration.

## 1 Introduction

Sparse representation can map high-dimensional vectors to a specific set of basis vectors, thereby achieving a sparse representation of signals. It has been widely applied in signal processing, machine learning, and computer vision [8, 27, 28]. Dictionary learning (DL) involves learning an overcomplete dictionary from a training dataset to represent input data succinctly [3, 13, 6].

For the design of dictionary learning algorithms, linear single-layer models serve as the cornerstone, with most existing algorithms based on this framework. However, single-layer DL models struggle to extract deep-seated information from the data [5, 14], thus driving research into multi-layer dictionary structures, i.e., deep dictionary learning (DDL) [17, 19]. Data is abundant and complex in the real world, containing much nonlinear information. Linear dictionary learning algorithms are limited to capturing only the linear structures within the data and are thus ineffective at handling nonlinear information. To address this issue, scholars have embarked on a research trend focusing on nonlinear dictionary learning (NLDL) [22, 26]. The conventional method of handling is to

use kernel techniques to map the nonlinear data to the high-dimensional space, where these features are linearly differentiable, and then dictionary learning is performed [10, 9]. However, the shortcoming of the kernel method is difficult to extend and cannot be displayed to get the nonlinear mapping function [1, 24]. In addition, another novel NLDL method can directly extract the hidden nonlinear information, but this method only inverts the signal and transforms the nonlinear problem into solving a linear task. Additionally, it only considers single-layer nonlinear dictionary structures and ignores deep nonlinear features.

Therefore, to address the existing issues with nonlinear dictionary learning, this paper proposes a novel deep nonlinear dictionary Learning (DNLDL) model, which is capable of uncovering deep nonlinear information and effectively extracting key features. Inspired by deep dictionary learning, we extend the single-layer NLDL to multiple layers, exploring the depth representation of data by learning multiple nonlinear dictionaries. To enhance the model's feature extraction capability and achieve a sparse representation of the data, we imposed $\ell_1$ sparsity constraints on the dictionaries and coefficients for each layer. To avoid directly inverting nonlinear functions in regularization optimization problems, we use the lifted proximal operator machine (LPOM) to transform DNLDL into a linear convex optimization problem. To speed up the convergence of the algorithm, we introduce Nesterov's acceleration technique and propose an accelerated version called Accelerated DNLDL_$\ell_1$. We validate the feasibility of the proposed algorithm and its applicability to mainstream nonlinear functions through numerical simulations. The use of the acceleration scheme is demonstrated to speed up the convergence of the algorithm significantly. Additionally, we explore the impact of the $\ell_1$ constraint on the algorithm, and the results from real tasks such as image classification and denoising demonstrate the effectiveness and superiority of the proposed algorithm.

The contributions of this work are summarised as follows:

- In order to extract nonlinear features from data in a hierarchical manner, we have developed a new model called deep nonlinear dictionary learning (DNLDL). This model uses $\ell_1$ regularization on the coefficients and dictionaries of all layers to extract hidden nonlinear features in the data, which we refer to as DNLDL_$\ell_1$.
- In addressing the challenge of nonconvex optimization in nonlinear models, we employ the LPOM concept to convert it into a linear convex optimization problem. Additionally, we integrate Nesterov's acceleration technique to attain an accelerated version.
- To validate the effectiveness of the proposed approach, numerical and application experiments were conducted, demonstrating its broad applicability and superiority in tasks such as image classification and denoising.

We standardize the notation in this paper: bold uppercase letters (e.g., $\boldsymbol{A}$) denote matrices, bold lowercase letters (e.g., $\boldsymbol{a}$) denote vectors, and regular lowercase letters (e.g., $a$) denote scalars.

## 2 Related work

**Dictionary learning (DL)** aims to find the optimal dictionary $\boldsymbol{A} \in \mathbb{R}^{n\times o}$ and coefficient $\boldsymbol{Z} \in \mathbb{R}^{o\times l}$ whose linear combination can sparsely represent the signal $\boldsymbol{S} \in \mathbb{R}^{n\times l}$ [18, 7]. We denote the dictionary as $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_o]$, where vector $\boldsymbol{a}_{j(1\leq j\leq o)}$ is the dictionary atom of size $n$. The sparse coefficient matrix is $\boldsymbol{Z} = [\boldsymbol{z_1}, \boldsymbol{z_2}, \cdots, \boldsymbol{z_o}]^T$, and $l$ is the number of training data samples. In the sparse coding task, $\boldsymbol{S}$ is the data consisting of o samples, which can be approximated as $\boldsymbol{S} \approx \boldsymbol{AZ}$. Dictionary learning is defined as:

$$\min_{\boldsymbol{A},\boldsymbol{Z}} \|\boldsymbol{S} - \boldsymbol{AZ}\|_F^2 + \lambda Sp(\boldsymbol{Z}), \tag{1}$$

where the $Sp(\boldsymbol{Z})$ is the sparse penalty, and $\lambda$ is the regularisation factor to balance the relationship between the error and the sparsity penalty. To prevent an imbalance in the ratio of $\boldsymbol{A}$ and $\boldsymbol{Z}$ due to an arbitrarily large dictionary, it is usually necessary to normalize the $\boldsymbol{A}$ with constraint $\{\|\boldsymbol{a}_j\|_2 \leq 1\}_{j=1}^{o}$. The traditional DL approach lays the cornerstone of the study, and $Sp(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_1$ is usually chosen as the sparse constraint [20, 25].

**Deep dictionary learning (DDL)** differs from single-layer DL in that it captures a deep representation of the data by learning multiple dictionaries of sparse coefficients. It is worth noting that at the outermost layer, the dictionary remains a linear structure, where the nonlinear functions are used to avoid collapsing the dimension into a matrix. Its structure is described as:

$$\min_{\boldsymbol{A}_i,\boldsymbol{Z}} \|\boldsymbol{S} - \boldsymbol{A}_1\rho(\boldsymbol{A}_2...\rho(\boldsymbol{A}_l\boldsymbol{Z}))\|_F^2 + \lambda Sp(\boldsymbol{Z}), \tag{2}$$

where $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_l$ are different layers of dictionaries, and $\rho$ is a nonlinear function.

**Nonlinear dictionary learning (NLDL)** method learns the nonlinear information in the data, it maps the product of the dictionary and coefficient through a nonlinear function $\rho$. The model is constructed as follows:

$$\min_{\boldsymbol{A},\boldsymbol{Z}} \|\boldsymbol{S} - \rho(\mathbf{A}\boldsymbol{Z})\|_{\boldsymbol{F}}^2 + \lambda Sp(\boldsymbol{Z}), \quad \text{s.t.}\ \|\boldsymbol{a}_j\|_2 = 1, 1 \leq j \leq n. \tag{3}$$

Based on the objective function Eq. (3), Chen et al. proposed NL-MOD and NL-KSVD algorithms [4] based on $\ell_0$ norm. Lin et al. proposed NLDL using MCP and GMC regularisation and optimized with a difference of convex function (DC) programming [16].

Inspired by the DDL model, this work attempts to expand the single-layer NLDL framework into a multi-layer structure to obtain deeply hierarchical nonlinear features of the data, and introduces $\ell_1$ norms to the proposed framework to further extract the key features.

## 3 Formulation

We propose a novel deep nonlinear dictionary learning model that achieves a sparse approximation of the dictionaries and coefficients at each layer and then

solves the problem by transforming it into a series of LPOM-based convex subproblems, where we avoid the optimization challenges associated with inverse functions. While nonlinear dictionary learning and fully connected neural networks are distinct concepts, in this paper, our proposed deep nonlinear dictionary learning can be viewed as a special case of deep neural networks. We add coefficient constraints at each layer to motivate capturing the essential features of the original signal, resulting in a compact and informative representation.

### 3.1 Formulation of DNLDL model

For our proposed DNLDL model, data is decomposed into multiple layers of features, where deeper coefficients are learned from the nonlinear mappings of several atoms from the previous layers. We consider each layer of the model as an auxiliary variable block, and the constrained problem proposed is formulated as follows:

$$\min_{\boldsymbol{A}_i,\boldsymbol{Z}_i} \|\boldsymbol{S}-\rho(\boldsymbol{A}_n\dots\rho(\boldsymbol{A}_2\rho(\boldsymbol{A}_1\boldsymbol{Z}_1)))\|_F^2,\ \text{s.t.}\begin{cases}\|\boldsymbol{A}_i\|_1\le\varepsilon, i=1,\dots,n\\ \|\boldsymbol{Z}_i\|_1\le\varepsilon, i=2,\dots,n+1,\end{cases} \tag{4}$$

where $\boldsymbol{A}_i$ and $\boldsymbol{Z}_i$ represent the *i-th* layer dictionary and coefficient, $\rho$ is a non-decreasing nonlinear function. In the middle layer, we define $\boldsymbol{Z}_2=\rho(\boldsymbol{A}_1\boldsymbol{Z}_1), \boldsymbol{Z}_3=\rho(\boldsymbol{A}_2\boldsymbol{Z}_2),\dots,\boldsymbol{Z}_{i+1}=\rho(\boldsymbol{A}_i\boldsymbol{Z}_i)$. The nonlinear mapping of the product of the last layer dictionary $\boldsymbol{A}_n$ and coefficients $\boldsymbol{Z}_n$ serves as an approximation to the signal $\boldsymbol{S}$, i.e. $\boldsymbol{S}=\rho((\boldsymbol{A}_n\boldsymbol{Z}_n))$. $\|\boldsymbol{Z}_i\|_1$ and $\|\boldsymbol{A}_i\|_1$ denote the $\ell_1$ norm regularization applied to the coefficients and dictionary, respectively. We then impose coefficient constraints on the dictionaries and coefficients of each layer, and we obtain a deep nonlinear dictionary learning model with all $\ell_1$ regularization layers.

**Introduction of the LPOM** Lifted proximal operator machine (LPOM) [12] converts a nonlinear function $\rho$ into an equivalent proximal operator and adds it as a penalty to the objective function, thus converting nonlinear constrained optimization into a convex minimization problem.

We start by solving a relatively simple single constraint problem consisting of scalars $r,t$:

$$\min_{r,t} s(r),\ \text{s.t.}\, r=\rho(t). \tag{5}$$

LPOM introduces a function $h(r,t)$, the optimal solution of which satisfies the constraints outlined in Eq. (5), i.e., $r=\rho(t)=\operatorname{argmin}_r h(r,t)$, Therefore, the constraints on a single variable are relaxed to

$$\min_{r,t} s(r)+\theta h(r,t). \tag{6}$$

In addition, the approximation operator is also a commonly used technique in optimization algorithms. For Eq. (5), the expression can be expressed in terms of the proximal operator as follows:

$$\operatorname{prox}_f(t)=\underset{r}{argmin}\, f(r)+\frac{1}{2}(r-t)^2. \tag{7}$$

Let $f(r) = \int_0^r (\rho^{-1}(t) - t)dv$, thus implying the existence of a convex set $\rho^{-1}(r) = \{t|r = \rho(t)\}$. The optimality condition for Eq. (7) is $0 \in (\rho^{-1}(r) - r) + (r - t)$, its solution is precisely the constraint $r = \rho(t)$. So we define $h(r, t) = f(r) + \frac{1}{2}(r - t)^2$.

**The DNLDL problem based on LPOM** To solve the DNLDL problem in this paper, we rewrite $h(r, t)$ into the matrix form $h(\boldsymbol{R}, \boldsymbol{T})$. The matrices $\boldsymbol{R}$ and $\boldsymbol{T}$ are then replaced by $\boldsymbol{Z}_i$ and $\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}$ to obtain the corresponding matrix expressions of the approximation operator:

$$\underset{\boldsymbol{Z}_i}{argmin}\, h(\boldsymbol{Z}_i, \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}) \equiv \mathbf{1}^T f(\boldsymbol{Z}_i)\mathbf{1} + \frac{1}{2}\|\boldsymbol{Z}_i - \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}\|_F^2. \tag{8}$$

Similarly, the optimality condition for Eq. (8) is $\mathbf{0} \in \rho^{-1}(\boldsymbol{Z}_i) - \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}$ and the corresponding optimal solution is $\boldsymbol{Z}_i = \rho(\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1})$, that is, the structural constraint problem for multilayer dictionary learning in Eq. (4).

We relax the original problem with multiple constraints in Eq. (4), transforming it into the following unconstrained problem:

$$\begin{aligned} \min_{\boldsymbol{A}_i, \boldsymbol{Z}_i} \frac{1}{2}\|\boldsymbol{S} - \rho(\boldsymbol{A}_n\boldsymbol{Z}_n)\|_F^2 + \sum_{i=2}^{n} \theta_i \Big( \mathbf{1}^T f(\boldsymbol{Z}_i)\mathbf{1} + \\ \frac{1}{2}\|\boldsymbol{Z}_i - \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}\|_F^2 \Big) + \sum_{i=2}^{n+1} \lambda_Z \|\boldsymbol{Z}_i\|_1 + \sum_{i=1}^{n} \lambda_A \|\boldsymbol{A}_i\|_1. \end{aligned} \tag{9}$$

However, unlike the independent single-variable constraints in Eq. (5), Eq. (4) involves recursive constraints. In the $i$-th layer, Eq. (4) is required to simultaneously satisfy both $\boldsymbol{Z}_i = \rho(\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1})$ and $\boldsymbol{Z}_{i+1} = \rho(\boldsymbol{A}_i\boldsymbol{Z}_i)$. From this, we need to use both $h(\boldsymbol{Z}_i, \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1})$ and $h(\boldsymbol{Z}_{i+1}, \boldsymbol{A}_i Z_i)$, it corresponds to the optimality condition of the following expression:

$$\min_{\boldsymbol{Z}_i} \theta_i h(\boldsymbol{Z}_i, \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}) + \theta_{i+1} h(\boldsymbol{Z}_{i+1}, \boldsymbol{A}_i\boldsymbol{Z}_i). \tag{10}$$

For $\boldsymbol{Z}_i$, the optimality condition for the alternative form in Eq. (9) is $\mathbf{0} \in \theta_i(\rho^{-1}(\boldsymbol{Z}_i) - \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}) + \theta_{i+1}(\boldsymbol{A}_i)^T(\boldsymbol{A}_i\boldsymbol{Z}_i - \boldsymbol{Z}_{i+1}), i = 2, \ldots, n$. However, its drawback lies in the inability to simultaneously satisfy both constraints for $\boldsymbol{Z}_i = \rho(\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1})$ and $\boldsymbol{Z}_{i+1} = \rho(\boldsymbol{A}_i\boldsymbol{Z}_i)$. To address this issue, we modify the optimization conditions as follows:

$$\mathbf{0} \in \theta_i(\rho^{-1}(\boldsymbol{Z}_i) - \boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}) + \theta_{i+1}(\boldsymbol{A}_i)^T(\rho(\boldsymbol{A}_i\boldsymbol{Z}_i) - \boldsymbol{Z}_{i+1}), i = 2, \ldots, n. \tag{11}$$

Simultaneously, similar to the function $f(r)$, we construct $g(r) = \int_0^r (\rho(t) - t)dt$ to ensure the iterative update of sparse coefficients in multi-layer nonlinear dictionary learning.

Finally, we employ LPOM to relax the constrained deep nonlinear dictionary learning problem into a convex optimization problem. The unconstrained

optimization expression for Eq. (4) as follows:

$$\min_{\boldsymbol{A}_i,\boldsymbol{Z}_i} \frac{1}{2}\|\boldsymbol{S}-\rho(\boldsymbol{A}_n\boldsymbol{Z}_n)\|_F^2+\sum_{i=2}^{n}\theta_i\Big(\mathbf{1}^T f(\boldsymbol{Z}_i)\mathbf{1}+\mathbf{1}^T g(\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1})\mathbf{1}+ \frac{1}{2}\|\boldsymbol{Z}_i-\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}\|_F^2\Big)+\sum_{i=1}^{n}\lambda_A\|\boldsymbol{A}_i\|_1+\sum_{i=2}^{n+1}\lambda_Z\|\boldsymbol{Z}_i\|_1, \tag{12}$$

where the first term is the approximation error, the second term is the internal structure of the multilayer dictionary learning model, and the last two terms are the $\ell_1$ regularization of the coefficients and dictionaries, respectively. $\theta_i$ is the balancing factor for each layer, and $\lambda_A$ and $\lambda_Z$ correspond to the positive regularization parameters of dictionaries and coefficients, respectively.

### 3.2 Optimization

Most dictionary learning models commonly use an alternating optimization scheme, which comprises two steps: sparse coding and updating dictionaries. We update the coefficients $\{\boldsymbol{Z}_i\}_{i=2}^{n+1}$ and dictionaries $\{\boldsymbol{A}_i\}_{i=1}^{n}$ separately, keeping the other layer variables fixed when updating either $\boldsymbol{Z}_i$ or $\boldsymbol{A}_i$.

**Updating coefficients $\{\boldsymbol{Z_i}\}_{\boldsymbol{i=2}}^{\boldsymbol{n+1}}$** We fix the dictionary of the current layer and all variables from other layers. Simplifying problem Eq. (12), we obtain:

$$\min_{\boldsymbol{A}_i,\boldsymbol{Z}_i}\sum_{i=2}^{n}\{\theta_i\Big(\mathbf{1}^T f(\boldsymbol{Z}_i)\mathbf{1}+\mathbf{1}^T g(\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1})\mathbf{1}+\frac{1}{2}\|\boldsymbol{Z}_i-\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}\|_F^2\Big)+\lambda_Z\|\boldsymbol{Z}_i\|_1\}. \tag{13}$$

For the update of $\boldsymbol{Z}_i(i=2,3,\ldots,n)$, Eq. (13) can be reformulated as the following optimization subproblems:

$$\begin{aligned}\min_{\boldsymbol{Z}_i}\theta_i&\left(\mathbf{1}^T f(\boldsymbol{Z}_i)\mathbf{1}+\frac{1}{2}\|\boldsymbol{Z}_i-\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}\|_F^2\right)\\ +\theta_{i+1}&\left(\mathbf{1}^T g(\boldsymbol{A}_i\boldsymbol{Z}_i)\mathbf{1}+\frac{1}{2}\|\boldsymbol{Z}_{i+1}-\boldsymbol{A}_i\boldsymbol{Z}_i\|_F^2\right)+\lambda_Z\|\boldsymbol{Z}_i\|_1,\end{aligned} \tag{14}$$

the optimality condition for Eq. (14) is

$$\mathbf{0}\in\theta_i(\rho^{-1}(\boldsymbol{Z}_i)-\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1})+\theta_{i+1}((\boldsymbol{A}_i)^T(\rho(\boldsymbol{A}_i\boldsymbol{Z}_i)-\boldsymbol{Z}_{i+1}))+\lambda_Z sign(\boldsymbol{Z}_i). \tag{15}$$

To avoid explicit use of $\rho^{-1}$, based on the fixed-point principle, we derive the following iterative expression:

$$\boldsymbol{Z}_i{}^{k+1}=\rho\left(\boldsymbol{A}_{i-1}\boldsymbol{Z}_{i-1}-\frac{\theta_{i+1}}{\theta_i}((\boldsymbol{A}_i)^T\rho(\boldsymbol{A}_i\boldsymbol{Z}_i{}^k)-\boldsymbol{Z}_{i+1})-\frac{\lambda_Z}{\theta_i}sign(\boldsymbol{Z}_i^k)\right). \tag{16}$$

For the last layer $\boldsymbol{Z}_{n+1}$, has $g(\boldsymbol{A}_{n+1}\boldsymbol{Z}_{n+1})=0$. Problem 12 is reduced to

$$\min_{\boldsymbol{Z}_{n+1}}\frac{1}{2}\|\boldsymbol{S}-\boldsymbol{Z}_{n+1}\|_F^2+\theta_{n+1}\Big(\mathbf{1}^T f(\boldsymbol{Z}_{n+1})\mathbf{1}+\frac{1}{2}\|\boldsymbol{Z}_{n+1}-\boldsymbol{A}_n\boldsymbol{Z}_n\|_F^2)+\lambda_Z\|\boldsymbol{Z}_i\|_1, \tag{17}$$

the optimality condition for Eq. (17) is

$$\mathbf{0} \in (\boldsymbol{Z}_{n+1} - \boldsymbol{S}) + \theta_{n+1}(\rho^{-1}(\boldsymbol{Z}_{n+1}) - \boldsymbol{A}_n\boldsymbol{Z}_n) + \lambda_Z sign(\boldsymbol{Z}_{n+1}). \tag{18}$$

Based on the fixed-point principle, the updating expression for $\boldsymbol{Z}_{n+1}$ at the $k$-th iteration is obtained as:

$$\boldsymbol{Z}_{n+1}^{k+1} = \rho\Big(\boldsymbol{A}_n\boldsymbol{Z}_n - \frac{1}{\theta_{n+1}}(\boldsymbol{Z}_{n+1}^k - \boldsymbol{S}) - \frac{\lambda_Z}{\theta_{n+1}} sign(\boldsymbol{Z}_{n+1}^k)\Big). \tag{19}$$

**Updating dictionary $\{\boldsymbol{A_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$** For $\{\boldsymbol{A}_i\}_{i=1}^n$, keeping the sparse coefficients within the same layer and other variables from different layers fixed. When $i = 1, 2, \ldots, n$, we simplify Eq. (12) to

$$\min_{\boldsymbol{A}_i} \mathbf{1}^T g(\boldsymbol{A}_i\boldsymbol{Z}_i)\mathbf{1} + \frac{1}{2}\|\boldsymbol{Z}_{i+1} - \boldsymbol{A}_i\boldsymbol{Z}_i\|_F^2 + \lambda_A\|\boldsymbol{A}_i\|_1, \tag{20}$$

and Eq. (20) can be rewritten as:

$$\min_{\boldsymbol{A}_i} \mathbf{1}^T \tilde{g}(\boldsymbol{A}_i\boldsymbol{Z}_i)\mathbf{1} - \langle \boldsymbol{Z}_{i+1}, \boldsymbol{A}_i\boldsymbol{Z}_i\rangle + \lambda_A\|\boldsymbol{A}_i\|_1, \tag{21}$$

where $\tilde{g}(r) = \int_0^r \rho(t)dt$, which is similar to $g(r)$. Here, a variant of the Accelerated Proximal Gradient (APG) method [12] is utilized to solve problem Eq. (21) through local linearization $\hat{g}(\boldsymbol{A}) = \tilde{g}(\boldsymbol{AZ})$, simultaneously, the use of $\rho^{-1}$ is avoided, enhancing the efficiency of the algorithm. We transform Eq. (21) as follows, see [12] for details:

$$\begin{aligned} {\boldsymbol{A}_i}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{A}} \langle \rho({\boldsymbol{T}_i}^k\boldsymbol{Z}_i), (\boldsymbol{A} - {\boldsymbol{T}_i}^k)\boldsymbol{Z}_i\rangle \\ &+ \frac{\beta}{2}\|(\boldsymbol{A} - {\boldsymbol{T}_i}^k)\boldsymbol{Z}_i\|_F^2 - \langle \boldsymbol{Z}_{i+1}, \boldsymbol{AZ}_i\rangle + \lambda_A\|\boldsymbol{A}_i\|_1, \end{aligned} \tag{22}$$

where ${\boldsymbol{T}_i}^k = \eta^k {\boldsymbol{A}_i}^k - \sqrt{\eta^k}(\eta^{k-1}{\boldsymbol{A}_i}^{k-1} - {\boldsymbol{A}_i}^k)$, $\eta^k$ can be calculated from $1 - \eta^k = \sqrt{\eta^k}(1 - \eta^{k-1})$, and $\beta$ denotes a Lipschitz constant.

Eq. (22) has a closed-form solution, and its least squares solution is given by:

$${\boldsymbol{A}_i}^{k+1} = {\boldsymbol{T}_i}^k - \frac{1}{\beta}\Big(\rho({\boldsymbol{T}_i}^k\boldsymbol{Z}_i) - \boldsymbol{Z}_{i+1} + \lambda_A sign(\boldsymbol{A}_i^k)((\boldsymbol{Z}_i)^\dagger)^T\Big)\boldsymbol{Z}_i^\dagger. \tag{23}$$

Here $\boldsymbol{Z}_i^\dagger$ denotes the pseudo-inverse of $\boldsymbol{Z}_i$.

### 3.3 Algorithms

The formulated objective function in Eq. (12) involves $\ell_1$ regularization on multi-layer dictionaries and coefficients. In the structure of DNLDL, we globally update the dictionary and coefficients alternately. Unlike other optimization methods, we avoid explicitly using the inverse of a nonlinear function, which reduces the restriction on its having invertibility. We summarize as Algorithm 1.

**Algorithm 1** Multilayer sparse regularised DNLDL (DNLDL_$\ell_1$)

**Input:** parameters $\beta, \lambda, \{\theta_i\}_{i=1}^n$, iteration $K$;
Randomly initialize $\{\boldsymbol{A}_i\}_{i=1}^n$, $\{\boldsymbol{Z}_i\}_{i=2}^{n+1}$
**for** $k = 1$ to $K$ **do**
  Updating ${\boldsymbol{Z}_i}^k$ from Eq. (16), $i = 2, 3, \ldots, n$.
  Updating ${\boldsymbol{Z}_{n+1}}^k$ from Eq. (19).
  Updating ${\boldsymbol{A}_i}^k$ from Eq. (23), $i = 1, 2, \ldots, n$.
**end for**
**Output:** $\{\boldsymbol{A}_i\}_{i=1}^n$, $\{\boldsymbol{Z}_i\}_{i=2}^{n+1}$

**Algorithm 2** Accelerated DNLDL_$\ell_1$

**Input:** parameters $\beta, \lambda, \{\theta_i\}_{i=1}^n$, iteration $K$, $\mu^1 = 1$;
Randomly initialize $\{\boldsymbol{A}_i\}_{i=1}^n$, $\{\boldsymbol{Z}_i\}_{i=2}^{n+1}$
**for** $k = 1$ to $K$ **do**
  $\mu^{k+1} = \dfrac{1 + \sqrt{(1 + 4(\mu^k)^2)}}{2}$.
  $\omega = \dfrac{1 - \mu^k}{\mu^{k+1}}$.
  Updating ${\boldsymbol{Z}_i}^k$ from Eq. (16), $i = 2, 3, \ldots, n$.
  Updating ${\boldsymbol{Z}_{n+1}}^k$ from Eq. (19).
  Updating ${\boldsymbol{A}_i}^k$ from Eq. (23).
  $\boldsymbol{A}_i^{k+1} = (1 - \omega)\boldsymbol{A}_i^k + \omega \boldsymbol{A}_i^{k-1}, i = 1, 2, \ldots, n$.
**end for**
**Output:** $\{\boldsymbol{A}_i\}_{i=1}^n$, $\{\boldsymbol{Z}_i\}_{i=2}^{n+1}$

Furthermore, we aim to accelerate the convergence speed of the algorithm by identifying an optimal computation point that expedites reaching the convergence threshold. Nesterov's acceleration technique [2, 21] involves, in each iteration, determining the next optimal update point $\xi^{k+1}$ based on both the current $\xi^k$ and the previous values $\xi^{k-1}$, rather than solely relying on the current suboptimal update point:

$$\xi^{k+1} = (1 - \omega)\xi^k + \omega\xi^{k-1}, \tag{24}$$

where

$$\omega = \frac{1 - \mu^k}{\mu^{k+1}}, \quad \mu^{k+1} = \frac{1 + \sqrt{(1 + 4(\mu^k)^2)}}{2}. \tag{25}$$

Therefore, by incorporating Nesterov's acceleration technique, we summarize the acceleration scheme for DNLDL_$\ell_1$ as Algorithm 2.

## 4 Experiment

In this section, we evaluate the performance of the proposed algorithm through numerical and application experiments. All experiments are conducted on an 11th generation Intel(R) Core(TM) i9-11900K @ 3.50GHz, and the results are the average of repeated runs.

In Numerical experiments, we compare the impact of the DNLDL_$\ell_1$ algorithm under various nonlinear functions and demonstrate its excellent signal recovery capability. Then, to address the issue of slow convergence speed for individual activation functions, we adopt an acceleration algorithm and prove its effectiveness. We compare the proposed model with state-of-the-art methods in different DL frameworks and show the superiority of our approach.

In application experiments, the DNLDL_$\ell_1$ algorithm is applied to image classification and denoising tasks. In image classification, we investigate the effect of sparse constraints on algorithm performanc. In image denoising, we compare the performance with mainstream algorithms under different noise levels.

### 4.1 Experimental settings and evaluation metrics

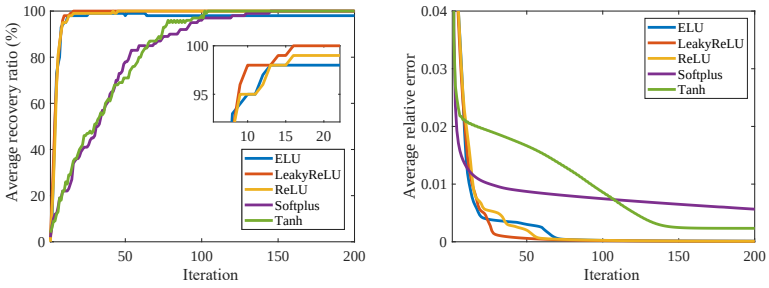
**Numerical experiments** As in the experimental setup of traditional DL methods, we randomly generate multiple dictionaries with independent and identically distributed columns and normalize each column using $\ell_2$ normalization. We generate $L$ samples $\{s_j\}_j^L$ from the multi-layer dictionary as the ground truth signal $\boldsymbol{S}$, and each synthesized signal is composed of a nonlinear mapping of the linear combination of different atoms from the multilayer dictionaries. We expect the synthesized signals to approximate the real simulated signals and recover the dictionary to the maximum extent possible.

The effectiveness of a two-layer dictionary model is demonstrated here, and it is easier to extend to deeper models when facing more complex tasks. We set dictionaries of size $\boldsymbol{A}_1 \in \mathbb{R}^{20\times 50}$ and $\boldsymbol{A}_2 \in \mathbb{R}^{50\times 20}$, and the sparse coding matrix $\boldsymbol{Z}_1 \in \mathbb{R}^{50\times 1500}$. Following the structure of deep nonlinear dictionary learning, we recursively generate simulated signals $\boldsymbol{S} \in \mathbb{R}^{50\times 1500}$ through the nonlinear mapping of each layer's dictionary and the sparse linear combination, i.e., $\boldsymbol{S} = \rho(\boldsymbol{A}_2\rho(\boldsymbol{A}_1\boldsymbol{Z}_1)$.

Dictionary recovery ratio and relative error are used as evaluation metrics. Each column in the dictionary is referred to as an atom. We scan each column in the learned dictionary for each atom in the ground truth dictionary and calculate their distance. If there exists $|\mathbf{a}_u^T\hat{\mathbf{a}}_j| > 0.99$, it is considered that the learned atom is successfully recovered. The proportion of recovered atoms to the total number of atoms determines the dictionary recovery ratio. To assess the discrepancy between the approximated signal and the true synthesized signal, we employ the relative error, expressed as follows:

$$\text{Relative error} = \frac{\|\boldsymbol{S} - \rho(\boldsymbol{A_2}\rho(\boldsymbol{A_1}\boldsymbol{Z}_1))\|_{\mathrm{F}}^2}{\|\boldsymbol{S}\|_{\mathrm{F}}^2}. \tag{26}$$

**Application experiments** In image classification, we use the MNIST dataset, which is composed of 10 types of handwritten digits with a size of $28 \times 28$ and contains 6,000 training images and 1,000 test images. A four-layer deep dictionary learning structure is set, namely $\boldsymbol{A_1} \in \mathbb{R}^{500\times 784}$, $\boldsymbol{A_2} \in \mathbb{R}^{300\times 500}$, $\boldsymbol{A_3} \in \mathbb{R}^{200\times 300}$, and $\boldsymbol{A_4} \in \mathbb{R}^{10\times 200}$. Therefore, the parameter count for our algorithm is $500\times 784+300\times 500+200\times 300+10\times 200 = 604,000$. Classification accuracy and loss are used as evaluation metrics for image classification.




(a) (b)

Fig. 1: Convergence with different nonlinear functions. (a) is the recovery rate of the dictionary, and (b) is the average relative error between the approximated signal and the true signal.

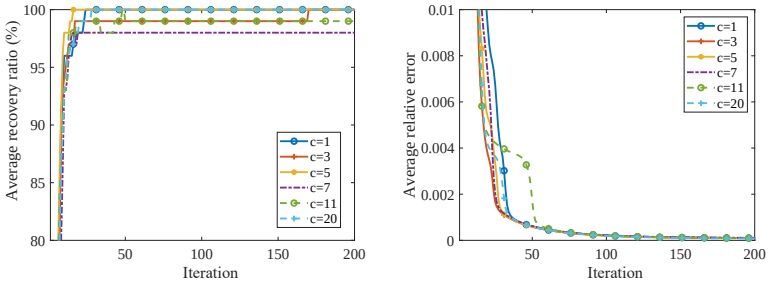
In image denoising, the dictionaries are set to $\boldsymbol{A_1} \in \mathbb{R}^{80\times64}$ and $\boldsymbol{A_2} \in \mathbb{R}^{64\times80}$ with parameters $80\times64+64\times80 = 10240$. The Peak signal-to-noise ratio (PSNR) and Structural similarity (SSIM) are used to evaluate the denoising performance.

### 4.2 Results of numerical experiments

**Effectiveness of the proposed algorithm** We choose the mainstream activation function to explore the generalization of the proposed algorithm. The parameter $c$ is the number of non-zero elements in each column of the sparse encoding matrix, and these non-zero entries' positions are randomly chosen. Fig. 1 shows the convergence of the proposed algorithm for the different nonlinear functions with $c = 5$. We are mainly concerned with the recovery rate of the last layer of the dictionaries. The experimental results indicate that ELU, LeakyReLU, and ReLU exhibit similar and rapid convergence, while Softplus and Tanh require more time to reach convergence. The recovery rates of the different nonlinear functions are close to 100%, and the reconstruction loss is close to 0. Then, we tested the convergence of DNLDL_$\ell_1$ for different values of $c$. Fig. 2 shows that the algorithm can converge stably at about 30 iterations for different $c$ values.

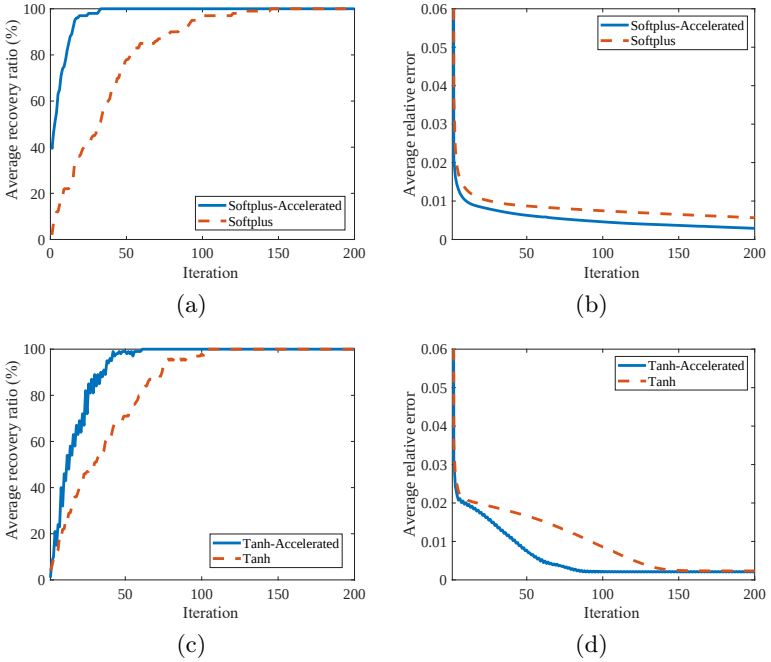
**Acceleration schemes for the proposed algorithm** Given the poor convergence performance of Softplus and Tanh, we use them to explore the impact of the acceleration scheme on the convergence of the algorithm. In Fig. 3, we observe a significant improvement in the dictionary recovery ratio and a smaller error. This suggests that the acceleration scheme finds better update points, improves the algorithm's convergence, and achieves lower reconstruction errors.

**Performance comparison of different algorithms** To demonstrate the superiority of our proposed model, we compared the DNLDL_$\ell_1$ algorithm with various state-of-the-art algorithms in the field of dictionary learning, including NLKSVD [4], NLMOD [4], MCP [15], GMC [23], and DDL [11]. This comparison




(a) (b)

Fig. 2: Convergence results for different numbers of nonzero element $c$. (a) is the dictionary recovery ratio, and (b) is the relative error with different $c$.

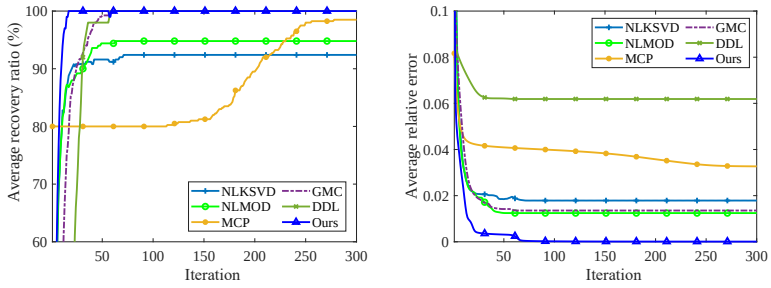



(a) (b)

(c) (d)

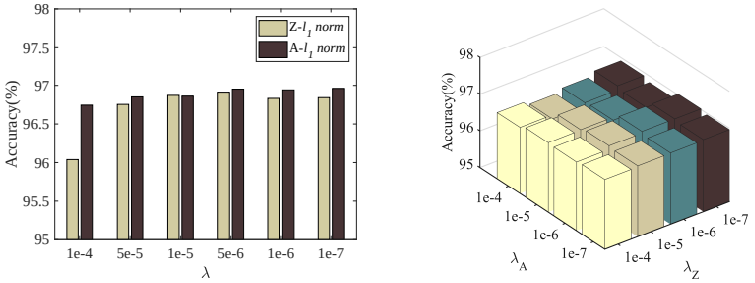
Fig. 3: Comparison of the accelerated version of the algorithm with the original DNLDL_$\ell_1$. (a) and (b) depict the dictionary recovery ratio and relative error when $\rho =$ Softplus, respectively. (c) and (d) are then $\rho =$Tanh.

encompasses single-layer structures, deep structures, linear models, and nonlinear models. NLKSVD and NLMOD are single-layer NLDL algorithms, MCP and GMC are advanced single-layer linear DL algorithms, and DDL is a deep linear dictionary learning model. To ensure the fairness of the experiment so that the single-layer model and multilayer structure construct the same $\boldsymbol{S}$ dimension, we set the dictionary to $\boldsymbol{A} \in \mathbb{R}^{50\times 80}$ and the coefficient to $\boldsymbol{Z} \in \mathbb{R}^{80\times 1500}$.




(a) (b)

Fig. 4: Convergence comparison of different dictionary learning algorithms. (a) and (b) are the dictionary recovery ratio and relative error, respectively.




(a) (b)

Fig. 5: Classification accuracy under different $\lambda$. (a) Applying $\ell_1$-norm on coefficients $\boldsymbol{Z}$ or the dictionaries $\boldsymbol{A}$ and (b) applying $\ell_1$-norm on both $\boldsymbol{Z}$ and $\boldsymbol{A}$.

We compare the performance of different algorithms for various DL frameworks, and it is noteworthy that in Fig. 4, the proposed algorithm achieves satisfactory results with a 100% dictionary recovery ratio and the fastest convergence speed. In terms of layer structure, multi-layer dictionaries can mine deeper and richer features than single-layer, and in terms of nonlinear frameworks, they can capture hidden representations better than linear models.

### 4.3 Results of application experiments

**Image classification** We investigated the influence of hyperparameters on the model. In the field of dictionary learning, the regularization parameter $\lambda$ plays a crucial role in balancing the sparse penalty term and the reconstruction error term. Since $\theta_i$ have relatively little impact on optimization performance in LPOM, we set them uniformly to 2. In Fig. 5, we compare the effect of the ReLU activation function on the experimental results at different $\lambda$ values. In subsequent experiments, we uniformly choose the optimal $\lambda$ as $1 \times 10^{-5}$.

In Tab. 1, we compare the classification effects of different activation functions, where ST is an irreversible soft-threshold function. To verify the effectiveness of adding $\ell_1$ regularization, we compared the performance under four

Table 1: Accuracy on MNIST test datasets with different nonlinear functions.

| $\ell_1$ norm | Sigmoid | Tanh | LeakyReLU | ReLU | ELU | ST |
|---|---|---|---|---|---|---|
| None | 90.74% | 96.51% | 97.24% | 97.97% | 97.47% | 81.87% |
| Z | 94.22% | 97.41% | 97.39% | 98.12% | 97.61% | 97.64% |
| A | 94.26% | 97.51% | 97.36% | 98.09% | 97.60% | 97.73% |
| A+Z | 94.18% | 97.48% | 97.29% | 98.10% | 97.70% | 97.83% |

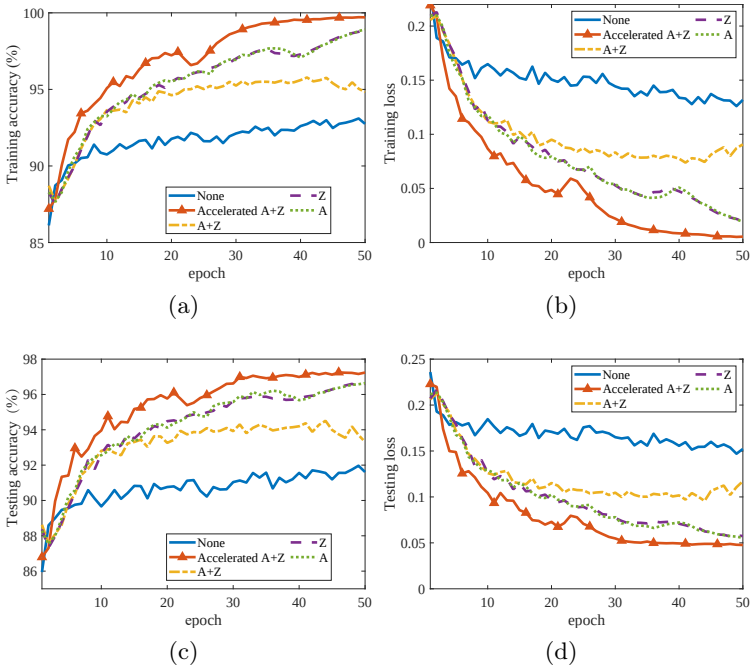



Fig. 6: Classification results of the acceleration and original algorithms ($\rho$=Sigmoid). (a) and (b) are training accuracy and loss, respectively, and (c) and (d) are test results

scenarios: without any sparse constraint (None), applying sparse constraint only to the coefficients $\boldsymbol{Z}$, applying sparse constraint only to the dictionaries $\boldsymbol{A}$, and applying sparse constraint to both coefficients $\boldsymbol{Z}$ and dictionaries $\boldsymbol{A}$. The experimental results indicate that the introduction of regularization improves the performance of the original algorithm. Under the Sigmoid nonlinear function, the test accuracy increased by 3.52%. This suggests that $\ell_1$ regularization avoids the interference of redundant features for a concise data representation. Fig. 6 shows that the Accelerated DNLDL_$\ell_1$ significantly improves the performance of the original algorithm in real tasks with faster convergence and higher accuracy.

**Image denoising** To generate a noisy image, we randomly add different levels $\sigma$ of white noise to the original image. Then the grayscale image of of size $512 \times 512$

Table 2: Denoising results of DNLDL with different nonlinear functions (Gaussian noise $\sigma = 25$).

| $\rho$ | House | | Peppers | | Boat | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Linear | 28.95 | 0.7166 | 28.84 | 0.7395 | 27.58 | 0.7017 |
| Softplus | 29.24 | 0.7338 | 29.95 | 0.7775 | 27.82 | 0.7191 |
| ReLU | 31.45 | 0.8377 | 30.95 | 0.8168 | 28.86 | 0.7591 |
| ELU | 31.35 | 0.8345 | 30.91 | 0.8166 | 28.89 | 0.7606 |
| LeakyReLU | 31.40 | 0.8385 | 30.79 | 0.8124 | 28.90 | 0.7556 |

is divided into $8 \times 8$ image blocks and flattened to $64 \times 1$ for denoising. Tab. 2 shows the performance of DNLDL_$\ell_1$ with different nonlinear functions, where ELU and ReLU have better denoising results. In Tab. 3 by comparing with other mainstream dictionary learning algorithms, our method has good image denoising capability with high PSNR and SSIM. Furthermore, applying sparse constraints to both dictionaries and coefficients has better denoising performance than acting on only one of them.

Table 3: Denoising results different algorithms (Gaussian noise $\sigma = 50$).

| Algorithm | House | | Peppers | | Boat | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| NLMOD[4] | 20.78 | 0.2951 | 19.22 | 0.2551 | 20.18 | 0.3227 |
| KSVD[23] | 27.64 | 0.7598 | 25.46 | **0.7607** | 25.40 | 0.6415 |
| GMC[23] | 27.60 | 0.7592 | 25.39 | 0.7570 | 25.45 | 0.6428 |
| Our(Z) | 28.14 | 0.7604 | 27.87 | 0.7435 | 25.92 | 0.6520 |
| Our(A) | 28.14 | 0.7648 | 27.85 | 0.7468 | 25.92 | 0.6530 |
| Our($\mathbf{A}$+$\mathbf{Z}$) | **28.68** | **0.7792** | **27.93** | 0.7446 | **27.93** | **0.6544** |

## 5 Conclusion

We propose a new deep nonlinear dictionary learning method, which simultaneously introduces a $\ell_1$-norm penalty to both the dictionaries and the coefficients, thus facilitating efficient extraction of key features. We advocate using the LPOM, approximating the model as a series of non-convex optimization subproblems. Through this process, we successfully circumvent the need for inverting nonlinear functions, relying solely on the inherent properties of these functions. In addition, we introduce Nesterov's acceleration techniques, the advantage of this approach whose strength lies in the increased computational efficiency. By integrating deep structures and norm penalties, it can more effectively learn crucial information from the data.

We need to specify that models of different depths should be designed according to the complexity of the task. In the future, we will focus on developing the architecture of DNDL models, specifically a lightweight, universal model.

**Acknowledgments.** This work is supported by the National Nature Science Foundation of China under Grant 62076077 and the Guangxi Science and Technology Major Project under Grant No. AA22068057.

## References


1. Baddoo, P.J., Herrmann, B., McKeon, B.J., Brunton, S.L.: Kernel learning for robust dynamic mode decomposition: linear and nonlinear disambiguation optimization. Proceedings of the Royal Society A **478**(2260), 20210830 (2022)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences **2**(1), 183–202 (2009)
3. Bruton, J., Wang, H.: Dictionary learning for clustering on hyperspectral images. Signal, Image and Video Processing **15**, 255–261 (2021)
4. Chen, X., Li, Y., Ding, S., Tan, B., Jiang, Y.: A novel nonlinear dictionary learning algorithm based on nonlinear-ksvd and nonlinear-mod. CAAI International Conference on Artificial Intelligence pp. 167–179 (2022)
5. Cheng, D., Zhang, X., Song, G.: A novel $\ell$p norm-based superposed sr recognition framework via adaptive weighted dictionary learning. Optik **224**, 165723 (2020)
6. Du, H., Zhang, Y., Ma, L., Zhang, F.: Structured discriminant analysis dictionary learning for pattern classification. Knowledge-Based Systems **216**, 106794 (2021)
7. Dumitrescu, B., Irofti, P.: Dictionary learning algorithms and applications. Springer (2018)
8. Hao, Y., Stuart, T., Kowalski, M.H., Choudhary, S., Hoffman, P., Hartman, A., Srivastava, A., Molla, G., Madad, S., Fernandez-Granda, C., et al.: Dictionary learning for integrative, multimodal and scalable single-cell analysis. Nature Biotechnology pp. 1–12 (2023)
9. Hu, J., Tan, Y.P.: Nonlinear dictionary learning with application to image classification. Pattern Recognition **75**, 282–291 (2018)
10. Huang, K., Wen, H., Ji, H., Cen, L., Chen, X., Yang, C.: Nonlinear process monitoring using kernel dictionary learning with application to aluminum electrolysis process. Control Engineering Practice **89**, 94–102 (2019)
11. Jiang, Y., Tan, B., Ding, S., Chen, X., Li, Y.: Device-free indoor localization based on kernel dictionary learning. IEEE Sensors Journal **23**(21), 26202–26214 (2023)
12. Li, J., Xiao, M., Fang, C., Dai, Y., Xu, C., Lin, Z.: Training neural networks by lifted proximal operator machines. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(6), 3334–3348 (2020)
13. Li, Z., Zhang, Z., Qin, J., Zhang, Z., Shao, L.: Discriminative fisher embedding dictionary learning algorithm for object recognition. IEEE transactions on neural networks and learning systems **31**(3), 786–800 (2019)
14. Li, Z., Xie, Y., Zeng, K., Xie, S., Kumara, B.T.: Adaptive sparsity-regularized deep dictionary learning based on lifted proximal operator machine. Knowledge-Based Systems **260**, 110123 (2023)
15. Li, Z., Yang, Z., Zhao, H., Xie, S.: Direct-optimization-based dc dictionary learning with the mcp regularizer. IEEE Transactions on Neural Networks and Learning Systems **34**(7), 3568–3579 (2023)
16. Lin, J., Li, Y., Tan, B., Ding, S.: Nonlinear dictionary learning algorithm with nonconvex regularizations. In: 2023 IEEE 9th International Conference on Cloud Computing and Intelligent Systems (CCIS). pp. 93–99 (2023)
17. Mahdizadehaghdam, S., Panahi, A., Krim, H., Dai, L.: Deep dictionary learning: A parametric network approach. IEEE Transactions on Image Processing **28**(10), 4790–4802 (2019)
18. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th annual international conference on machine learning. pp. 689–696 (2009)

19. Majumdar, A., Ward, R.: Robust greedy deep dictionary learning for ecg arrhythmia classification. 2017 International Joint Conference on Neural Networks (IJCNN) pp. 4400–4407 (2017)
20. Mukherjee, S., Basu, R., Seelamantula, C.S.: $\ell$1-k-svd: A robust dictionary learning algorithm with simultaneous update. Signal Processing **123**, 42–52 (2016)
21. Nhat, P.D., Le, H.M., Le Thi, H.A.: Accelerated difference of convex functions algorithm and its application to sparse binary logistic regression. In: IJCAI. pp. 1369–1375 (2018)
22. Schmitz, M.A., Heitz, M., Bonneel, N., Ngole, F., Coeurjolly, D., Cuturi, M., Peyré, G., Starck, J.L.: Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. SIAM Journal on Imaging Sciences **11**(1), 643–678 (2018)
23. Tan, B., Li, Y., Zhao, H., Li, X., Ding, S.: A novel dictionary learning method for sparse representation with nonconvex regularizations. Neurocomputing **417**, 128–141 (2020)
24. Tao, L., Jiang, X., Liu, X., Li, Z., Zhou, Z.: Multiscale supervised kernel dictionary learning for sar target recognition. IEEE Transactions on Geoscience and Remote Sensing **58**(9), 6281–6297 (2020)
25. Tariyal, S., Majumdar, A., Singh, R., Vatsa, M.: Deep dictionary learning. IEEE Access **4**, 10096–10109 (2016)
26. Van Nguyen, H., Patel, V.M., Nasrabadi, N.M., Chellappa, R.: Kernel dictionary learning. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2021–2024. IEEE (2012)
27. Wang, H., Ren, B., Song, L., Cui, L.: A novel weighted sparse representation classification strategy based on dictionary learning for rotating machinery. IEEE Transactions on Instrumentation and measurement **69**(3), 712–720 (2019)
28. Zheng, H., Yong, H., Zhang, L.: Deep convolutional dictionary learning for image denoising. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 630–641 (2021)