Max-sparsity atomic autoencoders with application to inverse problems

Ali Joundi, Yann Traonmilin, Alasdair Newson

Max-sparsity atomic autoencoders with application to inverse problems

Ali Joundi[1*], Yann Traonmilin[1], and Alasdair Newson[2]

[1] Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
[2] ISIR, Sorbonne Université, Paris, France. *ali.joundi@u-bordeaux.fr

Abstract. An atomic autoencoder is a neural network architecture that decomposes an image as a sum of low dimensional atoms. While it is efficient for image datasets which are well represented by this summation model, it is more limited for the representation of more generic images. In this article, we propose a new atomic model, the max-sparsity model to better represent images. We study some theoretical properties of this model and implement the corresponding atomic autoencoder. We show experimentally that it leads to a sparse decomposition of input images with interpretable low-level visual features. With this new architecture, we solve a super resolution inverse problem via a projected gradient descent that uses the trained network as a projection operator. The resulting estimation shows improved robustness compared to previous architectures.

Keywords: Atomic autoencoders · Sparse models · Image super-resolution

1 Introduction

An autoencoder is a neural network that approximates the identity function on a given dataset. By reducing the size of hidden layers, autoencoders produce low dimensional latent representations of elements of the dataset, an important property in the context of image processing. Formally, an autoencoder is the composition $f = f_D \circ f_E : \mathbb{R}^n \to \mathbb{R}^n$ of an encoder $f_E : \mathbb{R}^n \to \mathbb{R}^d$ that projects inputs onto the latent space with a decoder $f_D : \mathbb{R}^d \to \mathbb{R}^n$ that recovers them from their latent representations. To train an autoencoder on a dataset X, we minimize the loss function $\mathcal{L}$ defined by:

$$\mathcal{L}_X(f) = \mathcal{L}_X(f_D \circ f_E) := \sum_{x \in X} \|f_D \circ f_E(x) - x\|_2^2. \tag{1}$$

The benefits of accessing an explicit latent space are multiple. Autoencoders can generate new elements of the induced low dimensional space by decoding new latent vectors. If enough structure of the latent space is guaranteed, it allows to manipulate, edit or interpolate input images efficiently. In addition, autoencoders enable, by providing a low dimensional prior, the solving of ill-posed imaging problems such as super-resolution or deconvolution problems.

However, training autoencoders with the sole constraint that it needs to recover the input can lead to a latent set of image representations with little structure, due to its non-convexity. This heavily limits the effectiveness of autoencoders for these tasks.

Adding more structure to the latent space has been an important subject of research in the field of generative models. In particular, atomic autoencoders, proposed in [10], are custom autoencoders designed to decompose images into sparse representations, i.e. a sum of atoms from a given continuous dictionary. Given an image x, an atomic autoencoder encodes x into a latent vector $f_E(x) = \theta \in \mathbb{R}^{kd_0}$. The latent vector θ is divided into k atomic latent codes $\theta_i \in \mathbb{R}^{d_0}$ such that $\theta = (\theta_i)_{i=1}^k$ that are merged with a decoder having the form $f_D(\theta) = \sum_{i=1}^k g(\theta_i)$, where g is the atomic decoder modelling the continuous dictionary. It was shown in [10] that these atoms represent decorrelated low-level features of images. This property of neural networks is called *atomic disentanglement* of the latent space (not to be confused with the classical disentanglement notion that aims to represent high level interpretable features). However, on some image datasets, while giving promising decomposition of shapes, atomic autoencoders show some limits for providing a precise representation of images. One observation is that the sum of localized atoms fail to represent accurately typical image sets. Indeed, instead of being sums of objects, natural images can be seen as a set of objects occluding one another, which was e.g. statistically described by the dead leaves model [5].

Contribution In this paper, we propose an improvement of atomic autoencoders for image processing through a new sparsity model that models images as the maximum pixel wise of atoms in a given dictionary. This way of merging atoms aims to better represent natural images where objects may occlude one another. The resulting atomic autoencoder encodes the input image into a latent vector divided into blocks that are decoded separately to elementary images – or atoms. We obtain the output by computing the maximum pixel wise over these atoms instead of summing them.

In Section 2, we define the max-sparsity model and describe some of its structural properties in relation to a classical sparsity model. We show in particular that it may be included within a classical sparsity model under certain assumptions, which indicates greater structure induced by this model.

In Section 2.3, we propose a modification of the atomic autoencoder from [10] (SUM-AAE) to perform the maximum pixel wise operation (MAX-AAE) over the obtained atoms instead of summing them. We compare the two atomic autoencoders and the decomposition they achieve over two different datasets: Section 3.1 shows experimentally that SUM-AAE and MAX-AAE achieve similar reconstruction performances. However, our proposed architecture MAX-AAE achieves a decomposition with sharper and more disjoint atoms compared to SUM-AAE. **Section 3.2** shows that the decomposition achieved by MAX-AAE yields sparser representations of the considered images.

In Section 4, we use atomic autoencoders to solve a super resolution inverse problem. In this setting, they are used as a projection step for the Projected

Gradient Descent algorithm (PGD). This is motivated by the fact that these networks, when trained properly, project the input onto a low dimensional space of features captured during training. Experiments in section 4 prove that even if they are more constrained, using MAX-AAE in the PGD can recover original images as good as when we use a simple autoencoder instead and with improved performances compared to SUM-AAE.

Related work Atomic autoencoders [10] are closely related to dictionary learning. Given a dictionary $\mathcal{D}$ and a weight vector w (often considered sparse), an image x is modelled as $x = \mathcal{D}w$. This provides a prior in optimization problems that forces the recovered solution to have a desired structure (see [9] for an overview). To estimate the underlying dictionary $\mathcal{D}$ of a dataset, many algorithms have been proposed using for instance Non Negative Matrix factorization (NMF) [8], Singular Value Decomposition – as k-SVD [1] or even deep neural networks for continuous versions [14, 7]. However each of them supposes a linear relation between the learned dictionary and the original images.

Other natural images models have been explored and do not directly rely on a linear dictionary. The dead leaves model [5] supposes that each image consists of the superposition of several independent and different objects that partly occlude one another. It is built over a random process that generates objects to recover the final image. However, it does not provide a fixed dictionary that can be used to solve optimization problems. The idea of using a maximum element-wise function with atomic autoencoders is directly inspired by the dead leaves model, with the goal of modelling the superposition of objects through obtaining a dictionary.

By decomposing images into atoms, low level features are obtained. Plethora of works aim to discover underlying features of images. This operation is referred as *disentanglement*. Previous autoencoders architectures have been built to disentangle latent variables by modifying the loss to decorrelate them [4]. Non deterministic approaches were proposed: β-VAE [6] generalizes variational autoencoders with a weighted Kullback-Leibler loss. It achieves a better disentanglement at the expense of the reconstruction quality. Apart from autoencoders, Generative Adversarial Networks (GANs) have proven to be efficient in providing disentangled latent variables [3]. In this work, we specifically focus on deterministic latent representations (i.e. with an explicit encoder).

In terms of applications, generative models are mainly used to generate new data, but they are also useful in solving inverse problems. [12] gives an overview of several techniques with some associated theoretical results. To solve the underlying optimization problem, it is possible, in particular, to use generative neural networks in a projected gradient descent. The generative function projects inputs onto the natural images set at each iteration. For example, a basic autoencoder is used in [11] whereas [13] uses GANs to perform this projection.

2 The max-sparsity model

In the classical dictionary approach, an image x is modelled as a sum of elementary images within a dictionary $\mathcal{D}$ weighted by a weight matrix w *ie.* $x =$

$\mathcal{D}w$. Instead, we propose to consider each image as the superposition of different independent objects that occlude one another. To model such occlusions with a dictionary approach, we propose to perform a maximum pixel wise operation over the weighted elements of $\mathcal{D}$ to form the output. We call this model a *max-sparsity model* as opposed to the classical *sum-sparsity models*.

We define in this section the *max-sparsity model* and overview some of its properties in relation to the *sum-sparsity model*.

2.1 Definition

We denote by $\mathcal{D}$ a finite or infinite dictionary of vectors in $\mathbb{R}^n$. In the finite case, we can write $\mathcal{D} = \{a_i \in \mathbb{R}^n, i \leq \#\mathcal{D}\}$. We call $a_{i,j}$ the j-th coordinate of the i-th atom in $\mathcal{D}$ and x_i the i-th component of vector $x \in \mathbb{R}^n$.

We define a generalized sparsity model Σ as

$$\Sigma = \{x = \sigma(\lambda_1 a_1, \ldots, \lambda_k a_k), a_i \in \mathcal{D}, \lambda_i \in \mathbb{R}\} \tag{2}$$

where σ is an arbitrary aggregation function : $\mathbb{R}^{n\times k} \longrightarrow \mathbb{R}^n$ and k is the sparsity of the image in $\mathcal{D}$. Through this generalized sparsity model we define:

- the classical sparsity model Σ_k [9] (which we call *sum-sparsity model*) when the aggregation function σ is a sum, i.e.:

$$\Sigma_k := \left\{ x = \sum_{i=1}^{k} \lambda_i a_i, a_i \in \mathcal{D}, \lambda_i \in \mathbb{R} \right\}; \tag{3}$$

- the *max-sparsity* model $\Sigma_k^{\max}$ when the aggregation function $\sigma(\cdot) = \text{maxel}(\cdot)$ is the maximum element wise operation, i.e.:

$$\Sigma_k^{\max} := \{x = \text{maxel}(\lambda_1 a_1, \ldots, \lambda_k a_k), a_i \in \mathcal{D}, \lambda_i \in \mathbb{R}\}; \tag{4}$$

  where $v = \text{maxel}(u_1, \ldots, u_k)$ is defined by $(v_j)_{j\leq n} = (\max(u_{1,j}, \ldots, u_{k,j}))_{j\leq n}$.

As the representation is generally supposed to be sparse, we suppose that $k < n$. While we focus on the structural properties of the *max-sparsity* model in the following, this general formulation opens the broader question of the choice of aggregation function for a given dataset.

2.2 Properties of $\Sigma_k^{\max}$

In inverse problems, the model structure is crucial to provide guarantees. For example, positively homogeneous or homogeneous models have different recovery guarantees [15].

Definition 1 (Homogeneity). *A set Σ is positively homogeneous (called a cone) if for all $\lambda \in \mathbb{R}^+$ and $x \in \Sigma$, $\lambda x \in \Sigma$. The set $\Sigma_k^{\max}$ is homogeneous (called a union of subspaces) if for all $\lambda \in \mathbb{R}$ and $x \in \Sigma$, $\lambda x \in \Sigma$.*

Proposition 1. *The max-sparsity model set $\Sigma_k^{\max}$ is a cone. The set $\Sigma_k^{\max}$ is not a homogeneous model in general.*

This shows that $\Sigma_k^{\max}$ is not a union of linear subspaces. Thus, it has in general less structure than the sum-sparsity model. However, with more constraints on the atoms, it is included in the sum-sparsity model.

Proposition 2. *If the dictionary $\mathcal{D}$ verifies the two following assumptions:*

1. *for all $i \neq j$, $\operatorname{supp}(a_i) \cap \operatorname{supp}(a_j) = \varnothing$*
2. *for all i and $j \leq n$, $k \leq n$ $a_{i,j} \cdot a_{i,k} \geq 0$*

then $\Sigma_k^{\max} \subsetneq \Sigma_k$. In general, $\Sigma_k^{\max} \not\subset \Sigma_k$.

For example, if $\mathcal{D}$ is a subset of the canonical basis of $\mathbb{R}^n$ then $\Sigma_k^{\max} \subsetneq \Sigma_k$. However, it is not ensured that for every dictionary $\mathcal{D}$ the inclusion is still valid.

Given an underdetermined linear measurement operator $\mathbf{A}$, the problem of recovering $x \in \Sigma$ from $\mathbf{A}x$ has a unique solution if and only if $\ker(\mathbf{A}) \cap (\Sigma - \Sigma) = 0$ where $\Sigma - \Sigma$ is the set $\{z \in \mathbb{R}^n : \exists\, x, y \in \Sigma, z = x - y\}$ [16].

If $\Sigma_k^{\max} \subset \Sigma_k$, then, if a matrix $\mathbf{A}$ verifies the property $\ker(\mathbf{A}) \cap (\Sigma_k - \Sigma_k) = \{0\}$, then $\ker(\mathbf{A}) \cap (\Sigma_k^{\max} - \Sigma_k^{\max}) = \{0\}$. This shows that, when atoms have separated supports, the recovery of elements of Σ_k implies the recovery of elements of $\Sigma_k^{\max}$, but the converse is not generally true. Hence, for a fixed dictionary, recovering $\Sigma_k^{\max}$ is "easier" than recovering Σ_k. Given this conclusion, we compare the recovery of images by atomic autoencoders that enforce them to be in Σ_k and in $\Sigma_k^{\max}$ in Section 4. We will indeed observe that max-sparse atomic autoencoders provide better solutions to the image super-resolution problem.

2.3 Atomic autoencoders with max-sparsity model (MAX-AAE)

In [10], the sum-sparsity model is used to build the atomic autoencoder $f = f_D \circ f_E$ such that: $\theta = f_E(x)$ and $\tilde{x} = f_D(\theta) = \sum_{i=1}^{k} g(\theta_i)$, where f_E and g (defining f_D) are implemented with deep Leaky ReLU networks – Figure 1. In this work, we propose to modify the decoder f_D to define a max-sparsity model.

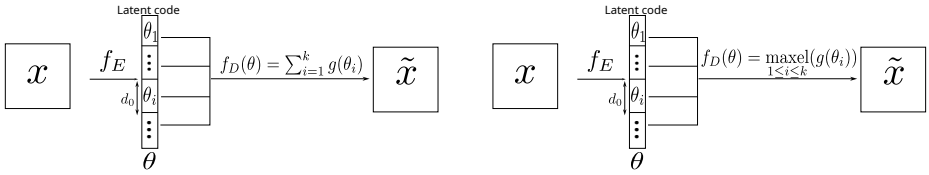
Definition 2 (Max-sparsity atomic autoencoder). *A max-sparsity atomic autoencoder is a function $f = f_D \circ f_E$ that is the composition of an encoder $f_E : \mathbb{R}^n \to \mathbb{R}^{kd_0}$ and a decoder $f_D : \mathbb{R}^d \to \mathbb{R}^n$ defined by*

$$f_D(\theta) = \operatorname{maxel}(g(\theta_1), \ldots, g(\theta_k)) \tag{5}$$

where $g : \mathbb{R}^{d_0} \to \mathbb{R}^n$ is the function decoding a single atomic latent code – see Figure 1.

Note that using a maximum pixel wise operation adds a non-linearity to the decoder acting as a ReLU function because $\max(a, b) = \texttt{ReLU}(a - b) + b$.

By constraining the decoder, the autoencoder output is falling in a potentially smaller space. This is related to the sparsity of the decomposition obtained by an atomic autoencoder. To assess this sparsity in the context of atomic autoencoders, we define the notion of *activated atoms*:



(a) SUM-AAE (b) MAX-AAE

Fig. 1: Architecture of sum-sparsity and max-sparsity atomic autoencoders.

Definition 3 (Activated atoms). *Given an image $x \in \mathbb{R}^n$ and a vector $\theta \in \mathbb{R}^{kd}$ such that $f_D(\theta) = x$, an atom $a_i = f_D(\theta_i)$ is activated with respect to a threshold s if $\frac{\|a_i\|_2}{\|x\|_2} \geq s$*

We will observe in Section 3.1 that max-sparsity atomic autoencoders (MAX-AAE) yield sparser representations than sum-sparsity atomic autoencoders (SUM-AAE) thus giving an interpretation of the improved performance of MAX-AAE in the context of inverse imaging problem.

3 Application to image decomposition

We compare the sum-sparsity atomic autoencoder (SUM-AAE) and the max-sparsity atomic autoencoder (MAX-AAE), defined in 2.3. We train both of them on two different datasets MNIST and Fashion-MNIST of size 30000. They have the same architecture: an encoder with two convolutional layers followed by two fully connected layers and a latent code composed of 20 blocks of size 10. Each block is decoded via the same decoder. It consists of four fully connected layers and three convolutional layers yielding 20 images (atoms) before applying the aggregation function. SUM-AAE sums the atoms whereas MAX-AAE applies the maximum pixel wise operation. We use Leaky ReLU as an activation function for all hidden layers. As a baseline, we train a simple autoencoder (AE) with a latent space of size 200 and having roughly the same number of parameters. Every comparison in this paper is achieved over subsets of size 600 of MNIST and Fashion MNIST test sets. We compare the recovered images through Peak signal-to-noise ratio (PSNR) metric. Furthermore, we assess the sparsity of autoencoded images through the mean number of *activated atoms* per image (defined in 3). The chosen threshold is 0.05. The code to reproduce experiments is available in [2].

3.1 Image decomposition with max-sparsity atomic autoencoder

We train two atomic autoencoders SUM-AAE and MAX-AAE over MNIST, a 28x28 digit images dataset and Fashion MNIST a 28x28 clothe images dataset. We keep the same structure for both of them but increased the number of filters and the size of the layers for Fashion MNIST.

Figure 2 compares the autoencoding of some images from the 2 datasets and their associated PSNRs. Table 1 gives the average PSNR over the test sets. According to these figures, MAX-AAE performs better than SUM-AAE and AE over MNIST. For Fashion MNIST, textured images are slightly degraded with MAX-AAE. However, we will observe in the context of image super-resolution that this slight degradation is a positive feature as MAX-AAE provides a better prior for image reconstruction and limits hallucinations (compared to AE AND SUM-AAE).

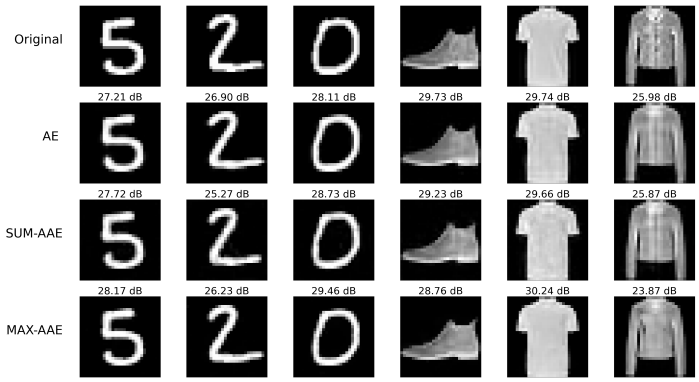


Fig. 2: Reconstruction of digits (MNIST) and clothes (Fashion MNIST) images through the autoencoders AE, SUM-AAE and MAX-AAE. The performances are similar for MNIST. For Fashion MNIST, MAX-AAE show a slight degradation for textured images.

Table 1: Average PSNR of the autoencoded images. We compare the autoencoders MAX-AAE, SUM-AAE and AE over two test sets of size 600.

	MNIST	Fashion MNIST
AE	26.72 ± 2.66 dB	$\mathbf{25.46 \pm 3.76}$ dB
SUM-AAE	27.35 ± 2.34 dB	25.28 ± 3.57 dB
MAX-AAE	$\mathbf{28.42 \pm 2.74}$ dB	24.25 ± 3.27 dB

Figures 3 shows the decompositions performed by SUM-AAE and MAX-AAE over MNIST and FashionMNIST. In both cases, MAX-AAE provides a decomposition with sharp distinct atoms whereas in SUM-AAE many atoms are blurry and scattered despite the mean PSNR of MAX-AAE being lower. We also observe that the support of atoms in the decomposition are mostly separated in MAX-AAE, indicating an inclusion in a sum-sparsity model (see Proposition 2).

In addition to these improved decompositions, the latent representations provided by MAX-AAE are sparser. We represent in Figure 4 the histogram of the number of activated atoms for a test set of size 600. To complement it, we show

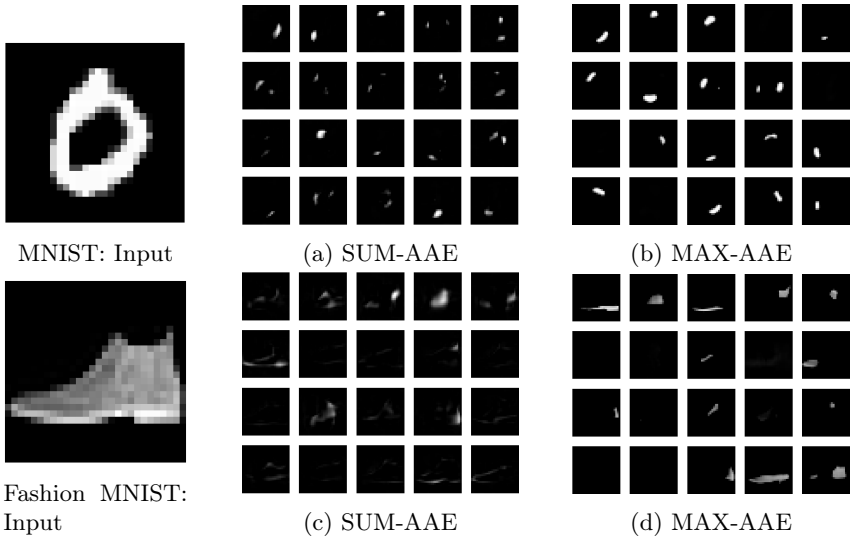
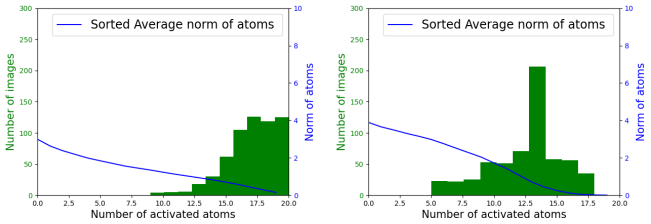


Fig. 3: Decomposition of an input from MNIST and Fashion MNIST through the two atomic autoencoders. SUM-AAE decomposes into scattered, blurry atoms while MAX-AAE produces sharp, disjoint atoms with a sparser decomposition.

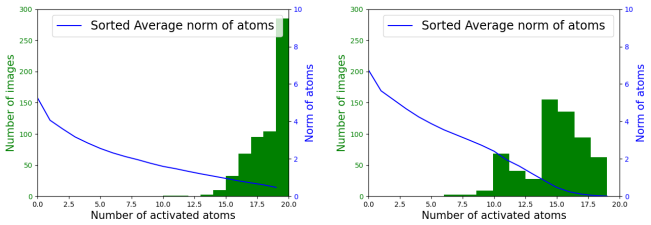
the average evolution of the norm of each atoms after ordering them in decreasing order. It shows that MAX-AAE activates less atoms than SUM-AAE as its related histogram is shifted to the left. The norm of the atoms decreases to 0 from the 16-th block for MAX-AAE when none of them is zero in SUM-AAE despite the faster decrease. Consequently, MAX-AAE produces a sparser representation. It must be noted that the training is performed without any regularization of the latent set in the loss. Hence, we blindly estimate the intrinsic dimensions of these two datasets with the atomic autoencoder architecture.

3.2 Atomic disentanglement of latent variables

As defined in [10], an atomic autoencoder achieves *atomic disentanglement* if it decomposes the input into atoms, each of them associated to a low-level feature. Modifying a latent block must act over the sole feature it represents. It is shown in [10] that SUM-AAE provides atomic disentanglement of its variables through two examples: MNIST Dataset and off-the-grid spikes. We focus here on Fashion MNIST. We show experimentally that even though it is a more complex dataset with more shape diversity, MAX-AAE achieves an atomic disentanglement of its latent set. Figure 3 shows that MAX-AAE produces sharp atoms. In particular, for Fashion MNIST, it is possible to assign each block to a particular low level feature. The shoe for instance has precise activated blocks that represent specific parts of the shoe. In Figure 3, we see that the lower part of the sole of the shoe is represented by the first block e.g. the first image in MAX-AAE decomposition.



(a) SUM-AAE for MNIST (b) MAX-AAE for MNIST



(c) SUM-AAE for Fashion MNIST (d) MAX-AAE for Fashion MNIST

Fig. 4: Histogram of activated atoms for 600 images of the test set (right y-axis) and the average norm of each atom in descending order (left y-axis). For MAX-AAE, the histogram is shifted to the left which shows that on average less atoms are activated and the norm of the atoms shatters to 0 earlier than SUM-AAE.

A way of showing atomic disentanglement is to modify values from the latent vector and see how it influences the output image. To do so, after choosing a dimension in a block, we compute the two extreme values along this dimension over all the test set. We, then, interpolate with a constant step between these two values. In other words, for a dataset X, if $\theta \in \mathbb{R}^{kd}$, $n \in \mathbb{N}$, $i \in [\![0, n]\!]$, $p \in [\![1, k]\!]$ and $q \in [\![1, d]\!]$, we modify θ and decode via f_D as follows:

$$\hat{x}_i = f_D(\theta^i) = f_D(\theta_{1,1} \dots | \dots \theta^i_{p,q} \dots | \dots \theta_{k,d}), \text{ where } \theta^i_{p,q} = \theta^{\min}_{p,q} + i\frac{\theta^{\max}_{p,q} - \theta^{\min}_{p,q}}{n} \tag{6}$$

$$\theta^{\max}_{p,q} = \max_{\theta \in f_E(X)} \theta_{p,q} \text{ and } \theta^{\min}_{p,q} = \min_{\theta \in f_E(X)} \theta_{p,q}, \text{ and } \hat{x}^i \text{ is the } i\text{-th output.}$$

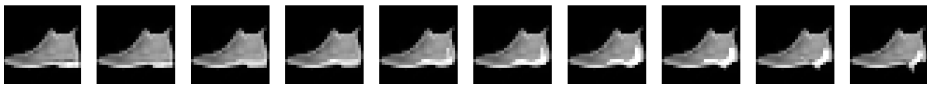


Fig. 5: Modification of the first variable from the first block of the latent vector associated to the shoe through MAX-AAE.This experiment shows an interpolation between a flat sole and a heel while leaving the other features unchanged.

Figure 5 shows that, by modifying one unique value of a dimension in a given block, only one feature in the image space is modified. In this specific example, from the first image (associated to the minimum value of dimension 0 from block 0) to the last image (associated to the maximum value) the only modified feature is the orientation lower part of the sole.

4 Application to image super resolution

Image super-resolution consists in estimating a high resolution image from a low resolution image. To solve this ill-posed problem, a low-dimensional prior is necessary. In this Section, we study the performance of our new MAX-AAE architecture in this context. We define

$$y = \mathbf{A}\hat{x}, \tag{7}$$

where $y \in \mathbb{R}^m$ is the measurement vector representing a subsampled image, $\hat{x} \in \mathbb{R}^n$ the original high resolution image. The linear operator $\mathbf{A} \in \mathbb{R}^{m \times n}$ represents a low pass filtering followed by a subsampling, i.e. $\mathbf{A} = \mathbf{SF}$ where $\mathbf{S}$ is the sub sampling matrix (by a factor 2 in the experiments). The matrix $\mathbf{F}$ models the convolution by a Gaussian blur.

Given a prior low dimensional model Σ on x, we want to solve:

$$\mathbf{x}^* = \underset{x \in \Sigma}{\operatorname{argmin}} \|\mathbf{A}x - y\|_2^2. \tag{8}$$

Here, Σ is induced by a trained autoencoder f and is, then, non-convex. Many methods can be used to solve this problem. To compare the different autoencoders capacities, we chose the projected gradient descent (PGD) described by the following iterations, given an initialization x_0:

$$\mathbf{x}_{k+1} = P_\Sigma(x_k - \gamma \mathbf{A}^T(\mathbf{A}x_k - \mathbf{y})) \tag{9}$$

where γ is the step size and the projection step is performed with the trained autoencoder (i.e. $P_\Sigma = f$, see e.g. [11]). For general low-dimensional models, it has been shown that this algorithm linearly converges to fixed points of the projection under a restricted isometry property of $\mathbf{A}$ and a *restricted* Lipschitz property of the projection [17].

We compare the results of Projected Gradient Descent with projection performed using the three autoencoders from the previous section over the two different datasets. We call PGD-AE, PGD-SUM-AAE and PGD-MAX-AAE the three variations of PGD. Table 2 compares them quantitatively over 600 images from the considered test set. On both MNIST and Fashion MNIST, we observe that PGD-MAX-AAE significantly outperforms PGD-SUM-AAE, and has comparable performance to PGD-AE.

Visually, we see crucial differences between the reconstructions of PGD-MAX-AAE and PGD-AE (Figure 6). While on simple data like MNIST, the visual results are similar, on more complex examples, there are a significant differences. For example in Figure 6, on the shirt – fifth column, the reconstruction

Table 2: PSNR of estimation. Even MAX-AAE is more constrained, PGD-MAX-AAE gives equal results than the classical PGD-AE. It also outperforms PGD-SUM-AAE.

	MNIST	Fashion MNIST
PGD-AE	26.90 ± 2.63 dB	21.95 ± 2.92 dB
PGD-SUM-AAE	25.25 ± 2.22 dB	20.91 ± 2.58 dB
PGD-MAX-AAE	$\mathbf{27.02 \pm 2.72}$ dB	$\mathbf{22.01 \pm 3.45}$ dB

is very different, with texture artefacts being hallucinated by both PGD-AE and PGD-SUM-AAE. We attribute this to the sparser representation induced by the max-sparsity model 3.1, which supports this model as a good image model. Note that for both MAX-AAE and SUM-AAE we can access the decomposition of the recovered image as in Figure 3, i.e. an "explanation" of the estimation is provided.

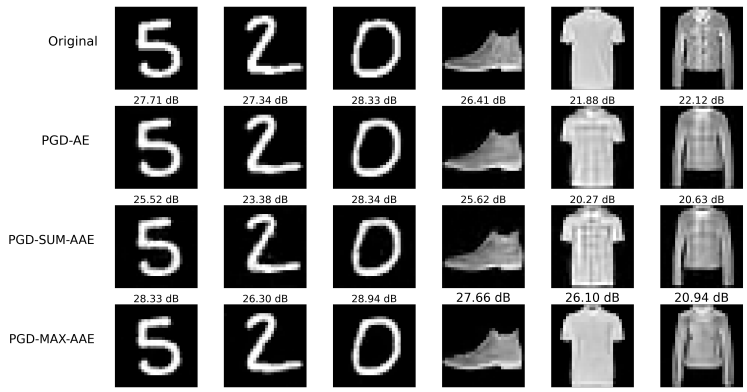


Fig. 6: Recovery of MNIST and Fashion MNIST images through PGD with different projection steps. Visually, the different PGD perform equally on MNIST. For Fashion MNIST, PGD-MAX-AAE is more robust as it creates less artifacts.

5 Conclusion

We have introduced a new atomic autoencoder based on a max-sparsity model. This autoencoder provides sparser and sharper atomic decompositions of the input. It also achieves an atomic disentanglement of its latent space.

We observed that using this new low dimensional representation for the super-resolution inverse problem provides robust reconstruction of images compared to the classical autoencoder and the original proposition of atomic autoencoder.

Many future directions arise from these results. On the theoretical side, it is still an open question to give a mathematical proof of the atomic disentanglement achieved by atomic autoencoders. In terms of further applications, the next step for atomic autoencoders is to be extended to photorealistic textured images.

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[2] Anonymized. Code to reproduce experiments, released at publication, 2024.

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[4] B. Cheung, J. Livezey, A. Bansal, and B. Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014.

[5] Y. Gousseau and F. Roueff. Modeling occlusion and scaling in natural images. *Multiscale Modeling & Simulation*, 6(1):105–134, 2007.

[6] I. Higgins, L. Matthey, A. Pal, C. P Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

[7] J. Hu and Y. Tan. Nonlinear dictionary learning with application to image classification. *Pattern Recognition*, 75:282–291, 2018.

[8] D. Lee and S. Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999.

[9] J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing, 2014.

[10] A. Newson and Y. Traonmilin. Disentangled latent representations of images with atomic autoencoders. In *Fourteenth International Conference on Sampling Theory and Applications*, 2023.

[11] P. Peng, S. Jalali, and X. Yuan. Solving inverse problems via auto-encoders. *IEEE Journal on Selected Areas in Information Theory*, 1(1):312–323, 2020.

[12] J. Scarlett, R. Heckel, M. Rodrigues, P. Hand, and Y. Eldar. Theoretical perspectives on deep learning methods in inverse problems. *IEEE journal on selected areas in information theory*, 3(3):433–453, 2022.

[13] V. Shah and C. Hegde. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *2018 IEEE ICASSP*, pages 4609–4613. IEEE, 2018.

[14] S. Tariyal, A. Majumdar, R. Singh, and M. Vatsa. Deep dictionary learning. *IEEE Access*, 4:10096–10109, 2016.

[15] Y. Traonmilin and R. Gribonval. Stable recovery of low-dimensional cones in hilbert spaces: One rip to rule them all. *Applied and Computational Harmonic Analysis*, 45(1):170–205, 2018.

[16] Y. Traonmilin, R. Gribonval, and S. Vaiter. A theory of optimal convex regularization for low-dimensional recovery. *Information and Inference: A Journal of the IMA*, 13(2):iaae013, 2024.

[17] Yann Traonmilin, Jean François Aujol, and Antoine Guennec. Towards optimal algorithms for the recovery of low-dimensional models with linear rates. *arXiv preprint arXiv:2410.06607*, 2024.

Proofs

Proof (of Proposition 1). Suppose $x \in \Sigma_k^{\max}$ and $\mu \in \mathbb{R}^+$. By definition of $\Sigma_k^{\max}$ (eq. (2)), we have $x = \sigma(\lambda_1 a_1, \dots, \lambda_k a_k)$. Let $1 \leq j_0 \leq n$. We have $x_{j_0} = \max(\lambda_1 a_{1,j_0}, \dots, \lambda_k a_{k,j_0})$ and there is $1 \leq i_0 \leq k$ such that $x_{j_0} = \lambda_{i_0} a_{i_0,j_0} \geq \lambda_i a_{i,j_0}$ for all $1 \leq i \leq n$.

This implies $\mu\lambda_{i_0} a_{i_0,j_0} \geq \mu\lambda_i a_{i,j_0}$ because $\mu \geq 0$. We conclude by

$$\begin{aligned} \mu x_{j_0} &= \mu \max(\lambda_1 a_{0,j_0}, \dots \lambda_k a_{k,j_0}) \\ &= \mu\lambda_{i_0} a_{i_0,j_0} = \max(\mu\lambda_1 a_{0,j_0}, \dots \mu\lambda_n a_{n,j_0}) \end{aligned} \tag{10}$$

and $\mu x = \sigma(\mu\lambda_1 a_1, \dots, \mu\lambda_k a_k) \in \Sigma_k^{\max}$.

Now consider a dictionary $\mathcal{D} = (a_i)_{i \leq k}$ where

$$a_1 = \begin{pmatrix} 1 \\ a_{1,2} \\ \vdots \\ a_{1,n} \end{pmatrix}; a_i = \begin{pmatrix} 0 \\ a_{i,2} \\ \vdots \\ a_{i,n} \end{pmatrix}$$

for all $i > 1$.

Let $x = \sigma(a_1, \dots, a_k)$, then $x_1 = \max(a_{1,1}, \dots, a_{k,1}) = \max(1, 0, \dots, 0) = 1$. On the other hand, if $-x \in \Sigma_k^{\max}$, then by definition of $\Sigma_k^{\max}$ (eq. (2)) there exists $(\lambda_1 \dots \lambda_k) \in \mathbb{R}^k$ such that $-x = \sigma(\lambda_1 a_1, \dots, \lambda_k a_k)$.

Therefore, we have $-x_1 = \max(\lambda_1, 0, \dots, 0) = -1$. First, $0 = \max(\lambda_1, 0, \dots, 0) = -1$ is impossible. Then $\lambda_1 = \max(\lambda_1, 0, \dots, 0) = -1$ which is impossible because $\max(\lambda_1, 0, \dots) \geq 0$. The two assertions are both impossible.

□

Proof (of Proposition 2). Suppose $x \in \Sigma_k^{\max}$, then by definition of $\Sigma_k^{\max}$ (equation (4)) $x = \sigma(\lambda_1 a_1, \dots, \lambda_k a_k)$. We want to show that we can find $(\mu_1, \dots, \mu_k) \in \mathbb{R}^k$, such that $x = \sum_{i=1}^k \mu_i a_i$.

The assumption (1) gives that for all $i \neq j$, $\operatorname{supp}(a_i) \cap \operatorname{supp}(a_j) = \varnothing$. Then, for a given coordinate r, there is k_0 such that $x_r = \max(\lambda_1 a_{1,r}, \dots, \lambda_k a_{k,r}) = \max(\lambda_{k_0} a_{k_0,r}, 0)$.

- If $\max(\lambda_{k_0} a_{k_0,r}, 0) = 0$, for all $\tilde{r} \in \operatorname{supp}(a_{k_0})$, we have $\lambda_{k_0} a_{k_0,\tilde{r}} = 0$ (with assumption (2)). Then we set $\mu_{k_0} = 0$.
- If $\lambda_{k_0} a_{k_0,r} = \lambda_{k_0} a_{k_0,r}$, or all $\tilde{r} \in \operatorname{supp}(a_{k_0})$, we have $\lambda_{k_0} a_{k_0,\tilde{r}} = \lambda_{k_0} a_{k_0,\tilde{r}}$ (with assumption (2)). Then we set $\mu_{k_0} = \lambda_{k_0}$.

If both cases, there is μ_{k_0} such that $x_{r \in \operatorname{sup}(a_{k_0})} = \mu_{k_0} a_{k_0}$ and

$$x = \sum_i^k x_{r \in \operatorname{sup}(a_i)} = \sum_i^k \mu_i a_i \in \Sigma_k. \tag{11}$$

□