Disentangled latent representations of images with atomic autoencoders
Supplementary material

Alasdair Newson[1] and Yann Traonmilin[2]
[1]Telecom Paris, Paris, France. alasdair.newson@telecom-paris.fr
[2]CNRS,Univ. Bordeaux, Bordeaux INP, IMB, UMR 5251,F-33400 Talence, France.
yann.traonmilin@math.u-bordeaux.fr

I. Proof of Lemma II.1

First, let us recall the "filling latent set assumption".

Assumption I.1 (Filling latent set)**.** *We say an atomic autoencoder $\phi \circ \zeta$ of Σ yields a filling latent set $\Theta = \phi^{-1}(\Sigma)$ if there is a bijection $h \in \mathcal{C}^1([0,1]^{kd_0}, \Theta)$ with $\mathcal{C}^1$ inverse between $[0,1]^{kd_0}$ and Θ.*

Lemma I.1. *Suppose $\Sigma, \Theta, \phi \circ \zeta, h$ verify Assumption I.1 and $\text{int}(h^{-1}(\Theta)) \neq \emptyset$ (where int denotes the interior). Let $\theta \in \Theta$ such that $h^{-1}(\theta) \in \text{int}(h^{-1}(\Theta))$. Then there exists an open set O of $\mathbb{R}^{kd_0}$ such that $\theta + O \subset \Theta$.*

Proof. Let $u \in \text{int}(h^{-1}(\Theta))$ such that $h(u) \in \Theta$. With Assumption I.1, as $h \in C^1$ and is a bijection, by continuity, there is an open set $\tilde{O} \in \mathbb{R}^{kd_0}$ such that $h(u + \tilde{O}) \in \Theta$. The image of $u + \tilde{O}$ is an open set $Q \subset \Theta$, hence $O = Q - h(u)$ is an open set and $\theta + O = h(u) - h(u) + Q = Q \subset \Theta$. □

II. Elementary properties of ideal autoencoders

Note that injectivity of ϕ (discussion on injectivity "up to a permutation" is out of the scope of this paper and left for future work) is not necessarily required to provide atomic disentanglement. Having equivalent representations does not affect the ability to navigate the latent space in our definition. We now consider f_E the trained encoder, f_D the trained decoder (we drop the star exponent to keep notations light), and call the latent code $z = f_E(x)$ for $x \in \Sigma$. Note that z is generally not the same as the ideal latent code θ. We place ourselves in the case where an ideal atomic autoencoder $\zeta \circ \phi$ that achieves atomic disentanglement is induced by $\Sigma_{\psi,k}$. We also suppose that we are able to train an autoencoder $f_D \circ f_E$ up to an arbitrary precision: in the case of a perfectly trained autoencoder, we have $f_E : \mathbb{R}^n \to \mathbb{R}^{kd_0}$ and for any $z \in \mathbb{R}^{kd_0}$, $f_D(z) = \sum_{j=1}^{k} g(z_j)$ for some $g : \mathbb{R}^{d_0} \to \mathbb{R}^n$ and for any $x \in \Sigma$, $f_D \circ f_E(x) = x$.

If ϕ, ζ, f_D, f_E are smooth, $f_D \circ f_E$ achieves atomic disentanglement.

Proposition II.1. *Suppose $\phi, \zeta, f_D, f_E \in \mathcal{C}^1$ and $\phi \circ \zeta$ verifies Assumption I.1 on Σ. Then $f_E \circ f_D$ verifies Assumption I.1 on Σ.*

Proof. Define $T = f_D^{-1}(\Sigma)$. Also, since $\phi \circ \zeta$ verifies Assumption I.1, we have that there exists a bijection $h \in \mathcal{C}^1$ between $[0,1]^{kd_0}$ and Θ.

Let $\tilde{h} = f_E \circ \phi \circ h$, the function $\tilde{h}$ is $\mathcal{C}^1$ by composition of $\mathcal{C}^1$ functions and any element of T is the image of an element of $[0,1]^{kd_0}$. Reciprocally, by defining $\tilde{h}^{-1} = h^{-1} \circ \zeta \circ f_D$ that is also $\mathcal{C}^1$, as $f_D(T) = \Sigma$, any element of $[0,1]^{kd_0}$ is the image of an element of T. This proves Assumption I.1. □

What this simple proposition tells us is that most of the desirable disentanglement properties are guaranteed by the structure of the autoencoder itself, and having a latent set that "fills" the latent space is a byproduct of smoothness of the autoencoder given the dimensions are well chosen (i.e. small enough). We now show that $f_D \circ f_E$ can possibly mix coordinates of the ideal latent blocks if ψ can be broken into simpler functions. We prove this for the case of two blocks of size 2 and $\Theta = [0,1]^{2\times 2}$.

Proposition II.2. *Let $\phi \circ \zeta$ be an atomic autoencoder such that there exists $\tilde{\psi}$ such that $\psi(\theta_i) = \tilde{\psi}(\theta_{i,1}) + \tilde{\psi}(\theta_{i,2})$ (with $\Theta = [0,1]^{2\times 2}$). Then there exists an atomic autoencoder $f_D \circ f_E$ such that $f_D(z) = g(z_1) + g(z_2)$ and for any $\theta = (\theta_{1,1}, \theta_{1,2}, \theta_{2,1}, \theta_{2,2}) \in \Theta$, $f_E(\phi(\theta)) = (\theta_{1,1}, \theta_{2,1}, \theta_{1,2}, \theta_{2,2})$, i.e. the encoder f_E mixes the ideal latent coordinates into two different blocks.*

Proof. For the decoder, just consider $f_D = \phi$ and $g = \psi$. Now define the permutation p of $\{1,2,3,4\}$, such that $p(1,1) = (1,1)$, $p(1,2) = (2,1)$, $p(2,1) = (1,2)$, $p(2,2) = (2,2)$) and the function $\rho : [0,1]^{2\times 2} \to [0,1]^{2\times 2}$ such that $[\rho(\theta)]_{p(i,j)} = \theta_{i,j}$. Now define $f_E = \rho \circ \zeta$. We have

$$\begin{aligned} f_D \circ f_E(x) &= \phi(\rho(\zeta(x))) = \phi(\rho(\theta)) = \phi(\theta_{11}, \theta_{2,1}, \theta_{1,2}, \theta_{2,2}) \\ &= \tilde{\psi}(\theta_{1,1}) + \tilde{\psi}(\theta_{2,1}) + \tilde{\psi}(\theta_{1,2}) + \tilde{\psi}(\theta_{2,2}) \\ &= \phi(\theta) = x. \end{aligned} \tag{1}$$

□

In other words, if the function defining the dictionary is the sum of two elementary functions then single atoms could be

Conv. part of encoder - number of filters per layer

Filters			
32	16	8	8

MLP part of encoder and size of latent space

Neurons	Neurons	Neurons (kd_0)	d_0	k
512	512	96	6	16

MLP. part of block decoder

Neurons		
6 (d_0)	32	32

Conv part of block decoder - number of channels per layer

Filters					
8	8	8	16	32	1 (output)

TABLE I

ARCHITECTURE OF THE ATOMIC AUTOENCODER USED FOR THE EXPERIMENTS IN THIS PAPER, FOR THE CASE OF MNIST. IN THE CASE OF MNIST, THE ENCODER PROJECTS IMAGES TO A LATENT SPACE OF SIZE $kd_0 = 6 \times 16$. EACH BLOCK DECODER IS DEEPER THAN THE ENCODER: THIS IS NORMAL, SINCE THE BLOCK DECODER MUST GO FROM A LATENT CODE OF SIZE $d_0 = 6$ TO AN IMAGE OF SIZE 128×128.

coded on several blocks. This is often not desirable because each block should modify an individual "simple" feature. In terms of design of the autoencoder, this tells us that if ψ can be broken down into simpler functions, then latent blocks should be chosen to be smaller. In our two practical examples, we observe that ψ is no such function.

III. IMPLEMENTATION DETAILS

The DNN architecture is designed to be as simple as possible. The encoder performs iterated 3×3 convolutions, with a stride of 2×2 (subsampling by 2 at each layer), followed by a leaky ReLU non-linearity ($\alpha = 0.2$). There are four such layers.The number of convolutional channels at the output of the first layer is fixed to 32, and this is then divided by 2 until reaching 8. This reduction of channels is designed to gradually reach a compact representation of the image.

Finally, two fully connected layers, with leaky ReLU ($\alpha = 0.2$) are used to project the tensors to the latent space $\mathbb{R}^{kd_0}$, with k, d_0 chosen according to application.

The block decoder g takes a latent block (in $\mathbb{R}^{d_0}$) and outputs an image. The first layers of g are two fully connected layers with leaky ReLUs to reach a size of $2 \times 2 \times 8$. This is followed by a convolutional section, with 3×3 convolutions. The convolutional section is not symmetric with respect to the encoder, since we go from a small $2 \times 2 \times 8$ tensor to a full image. Each convolutional layer contains a 3×3 convolution,

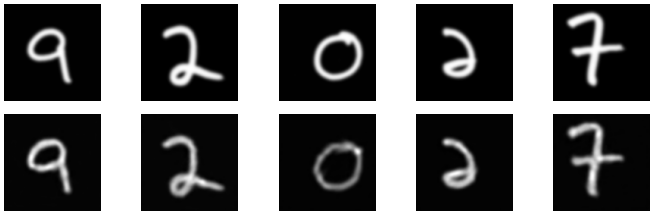

Fig. 1. **Autoencoding 128×128 mnist images on $\mathbb{R}^{30}$.**

followed by an upsampling by 2×2 and finally a leaky ReLU. The number of channels (filters) is chosen in a manner similar to the encoder: we keep 8 channels until we can increase them by multiplying by 2 until the final layer, which is 32. However, we stress that the encoder and the block decoder are *not* symmetric, as the block decoder goes from each latent block z_i rather than the total latent code.

In Table II, we show the specific architecture chosen for the mnist images. In this case, the encoder projects images to a latent space of size $kd_0 = 6 \times 16$. We considered that each small segment/penstroke present in mnist images can be parametrised by 6 parameters, and we let there be a maximum of 16 individual strokes in the image. This number of strokes was obtained by fixing the size of an image patch in which we considered a stroke would be carried out. We fixed this to 32 (recall that our high-resolution mnist images are of size 128×128), which gives $4 \times 4 = 16$ image patches. Note that this is a slight over-parametrisation of mnist data, since we considered that strokes could be present anywhere in the image (even though mnist images are quite well centred), however the total number of latent coordinates is still very low: 196). If we compare this with other autoencoder architectures (eg. variational autoencoders), the number of parameters is higher. This is quite normal, since each of our parameters is disentangled with respect to an image penstroke. In variational autoencoders, the latent codes may be disentangled with respect to the number of classes, which gives smaller latent spaces, but this is completely different from decomposing an image into visually distinct atomic components. Thus, comparisons with such approaches are not meaningful here.

We also note the asymmetry between the encoder and block decoder: each block decoder is deeper than the encoder. This is normal, since the block decoder must go from a latent code of size $d_0 = 6$ to an image of size 128×128, whereas the encoder goes from 128×128 to 96 (the full latent space size).

IV. VERIFICATION THAT THE ATOMIC AUTOENCODER SUCCEEDS IN AUTOENCODING

In Figure 1, we verify that the atomic autoencoder indeed succeeds in the autoencoding task. In other words, the output of the atomic autoencoder is indeed close to the input.

V. INTERPOLATION IN THE LATENT SPACE OF THE ATOMIC AUTOENCODER

In Figure 2, we show the result of interpolating between two latent points in the latent space of the atomic autoencoder. It

can be seen, that the interpolation modifies the line segments sequentially, to go from a number 6 to a number 2.

VI. Navigation in the latent space for off-the-grid spikes

Fig. 2. **Interpolation with mnist images.** We have interpolated linearly between two latent codes of images from the mnist dataset. We observe that the strokes are shortened or lengthened to go from a 6 to a 2.

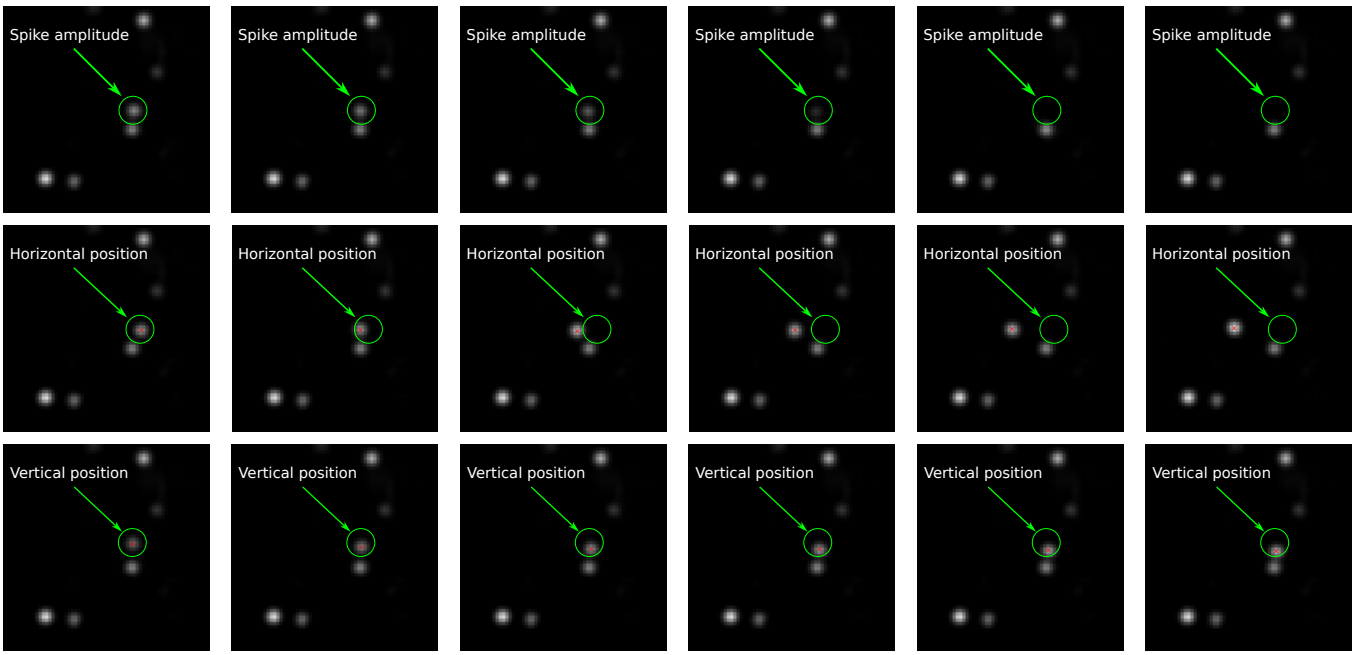

Fig. 3. **Navigation in the latent space for images of spikes.** Left to right: linearly modification of a latent coordinate in a block, keeping the others constant. Top to bottom: modification of the three different coordinates of the latent block. We observe that the first coordinate changes the amplitude, while the second and third modify the position (red mark). No supervision in the training was involved here.