SUPPLEMENTARY MATERIALS: Wasserstein Dictionary Learning: Optimal Transport-based unsupervised non-linear dictionary learning*

Morgan A. Schmitz[†], Matthieu Heitz[‡], Nicolas Bonneel[‡], Fred Ngolè[§], David Coeurjolly[‡], Marco Cuturi[¶], Gabriel Peyré[‖], and Jean-Luc Starck[†]

SM1. Detailed derivations.

Let us first introduce the notation:

$$\varphi : \begin{array}{c} \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N \\ b_s, d \mapsto K^\top \dfrac{d}{Kb_s} \end{array}.$$

SM1.1. Computation of $\partial_b \varphi$.

By definition:

$$\frac{\partial \varphi}{\partial b_s}(b_s, d) = -K^\top \Delta\left(\frac{d}{(Kb_s)^2}\right) K \tag{SM1}$$

In what follows, we will denote $\varphi_{NS}(b, D) = \left[\varphi(b_1, d_1)^\top, \dots, \varphi(b_S, d_S)^\top\right]^\top \in \mathbb{R}^{NS}$:

$$\partial_b \varphi_{NS}(b, D) = \begin{pmatrix} \frac{\partial \varphi(b_1,d_1)}{\partial b_1} & \mathbf{0}_{N\times N} & \dots & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \frac{\partial \varphi(b_2,d_2)}{\partial b_2} & \dots & \mathbf{0}_{N\times N} \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_{N\times N} & \dots & \mathbf{0}_{N\times N} & \frac{\partial \varphi(b_S,d_S)}{\partial b_S} \end{pmatrix}$$

SM1.2. Computation of Ψ_b.

Taking the logarithm of (16) yields:

$$\log(\Psi(b, D, \lambda)) = \sum_s \lambda_s \log(\varphi(b_s, d_s))$$

The differentiation of which gives us:

$$\Delta\left(\frac{\mathbb{1}_N}{\Psi(b, D, \lambda)}\right) \partial_b \Psi(b, D, \lambda) = \begin{pmatrix} \lambda_1 I_N & \dots & \lambda_S I_N \end{pmatrix} \Delta\left(\frac{\mathbb{1}_{NS}}{\varphi_{NS}(b, D)}\right) \partial_b \varphi_{NS}(b, D)$$

$$\implies \Psi_b = [\partial_b \varphi_{NS}(b, D)]^\top \Delta\left(\frac{\mathbb{1}_{NS}}{\varphi_{NS}(b, D)}\right) J_\lambda \Delta(\Psi(b, D, \lambda)) \tag{SM2}$$

*Part of this work was presented as a conference proceeding [SM1].

[†]Astrophysics Department, IRFU, CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France (morgan.schmitz@cea.fr)
Université Paris-Diderot, AIM, Sorbonne Paris Cité, CEA, CNRS, F-91191 Gif-sur-Yvette, France
[‡]Université de Lyon, CNRS/LIRIS, Lyon, France
[§]LIST, Data Analysis Tools Laboratory, CEA Saclay, France
[¶]Centre de Recherche en Economie et Statistique, Paris, France
[‖]DMA, ENS Ulm, Paris, France

Where $J_\lambda = \begin{pmatrix} \lambda_1 I_N \\ \vdots \\ \lambda_S I_N \end{pmatrix} \in \mathbb{R}^{NS\times N}$.

SM1.3. Computation of Ψ_D.

Let $i \in \{1,\dots,S\}$.

$$\Psi(b,D,\lambda) = \prod_{s\neq i} \Delta(\varphi_c(b_s,d_s))^{\lambda_s}.\left(K^\top \frac{d_i}{Kb_i}\right)^{\lambda_i}$$

And:

$$\frac{\partial\left(K^\top \frac{d_i}{Kb_i}\right)^{\lambda_i}}{\partial d_i} = \lambda_i \Delta\left(K^\top \frac{d_i}{Kb_i}\right)^{\lambda_i-1} K^\top \Delta\left(\frac{\mathbb{1}_N}{Kb_i}\right)$$

$$\Longrightarrow \quad \frac{\partial \Psi}{\partial d_i}(b,D,\lambda) = \lambda_i \frac{\Delta(\Psi(b,D,\lambda))}{\Delta\left(K^\top \frac{d_i}{Kb_i}\right)} K^\top \left(\frac{\mathbb{1}_N}{Kb_s}\right) \tag{SM3}$$

SM1.4. Computation of Φ_b.

$$\partial_b \Phi(b,D,\lambda) = \begin{pmatrix} \Delta\left(\frac{\mathbb{1}_N}{\varphi(b_1,d_1)}\right) \\ \vdots \\ \Delta\left(\frac{\mathbb{1}_N}{\varphi(b_S,d_S)}\right) \end{pmatrix} \partial_b \Psi(b,d)$$

$$- \begin{pmatrix} \Delta\left(\frac{\Psi(b,D,\lambda)}{\varphi(b_1,d_1)^2}\right)\frac{\partial\varphi(b_1,d_1)}{\partial b_1} & \mathbf{0}_{N\times N} & \dots & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \Delta\left(\frac{\Psi(b,D,\lambda)}{\varphi(b_2,d_2)^2}\right)\frac{\partial\varphi(b_2,d_2)}{\partial b_2} & \dots & \mathbf{0}_{N\times N} \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_{N\times N} & \dots & \mathbf{0}_{N\times N} & \Delta\left(\frac{\Psi(b,D,\lambda)}{\varphi(b_S,d_S)^2}\right)\frac{\partial\varphi(b_S,d_S)}{\partial b_S} \end{pmatrix}$$

$$= \Delta\left(\frac{\mathbb{1}_{NS}}{\varphi_{NS}(b,D)}\right) I_{N,S}^\top(\partial_b\Psi(b,D,\lambda)) - \Delta\left(\frac{\mathbb{1}_{NS}}{\varphi_{NS}(b,D)}\right)\Delta(\Phi(b,D,\lambda))\partial_b\varphi_{NS}(b,D)$$

$$= \Delta\left(\frac{\mathbb{1}_{NS}}{\varphi_{NS}(b,D)}\right)\left[I_{N,S}^\top(\partial_b\Psi(b,D,\lambda)) - \Delta(\Phi(b,D,\lambda))\partial_b\varphi_{NS}(b,D)\right]$$

$$\Longrightarrow \quad \Phi_b = \left[\Psi_b I_{N,S} - [\partial_b\varphi_{NS}(b,D)]^\top \Delta(\Phi(b,D,\lambda))\right]\Delta\left(\frac{\mathbb{1}_{NS}}{\varphi_{NS}(b,D)}\right)$$

$$\overset{\text{(SM2)}}{=} [[\partial_b\varphi_{NS}(b,D)]^\top \Delta\left(\frac{\mathbb{1}_{NS}}{\varphi(b,D)}\right) J_\lambda \Delta(\Psi(b,D,\lambda)) I_{N,S}$$

$$- [\partial_b\varphi_{NS}(b,D)]^\top \Delta(\Phi(b,D,\lambda))]\Delta\left(\frac{\mathbb{1}_{NS}}{\varphi_{NS}(b,D)}\right)$$

$$= [\partial_b\varphi_{NS}(b,D)]^\top\left[\Delta\left(\frac{\mathbb{1}_{NS}}{\varphi(b,D)}\right) J_\lambda \Delta(\Psi(b,D,\lambda)) I_{N,S} - \Delta(\Phi(b,D,\lambda))\right]\Delta\left(\frac{\mathbb{1}_N}{\varphi_{NS}(b,D)}\right) \tag{SM4}$$

Where $I_{N,S} = [I_N,\dots,I_N] \in \mathbb{R}^{N\times NS}$. Moreover, we have:

$$\Delta\left(\frac{\mathbb{1}_{NS}}{\varphi(b,D)}\right)J_\lambda\Delta(\Psi(b,D,\lambda)) = \begin{pmatrix}\Delta(1/\varphi(b_1,d_1)) & & \\ & \ddots & \\ & & \Delta(1/\varphi(b_S,d_S))\end{pmatrix}\begin{pmatrix}\lambda_1\Delta(\Psi(b,D,\lambda))\\ \vdots \\ \lambda_S\Delta(\Psi(b,D,\lambda))\end{pmatrix}$$
$$= \begin{pmatrix}\lambda_1\Delta\left(\frac{\Psi(b,D,\lambda)}{\varphi(b_1,d_1)}\right) & & \\ & \ddots & \\ & & \lambda_S\Delta\left(\frac{\Psi(b,D,\lambda)}{\varphi(b_S,d_S)}\right)\end{pmatrix}$$
$$= \Delta(\Phi(b,D,\lambda))\begin{pmatrix}\lambda_1 I_N\\ \vdots \\ \lambda_S I_N\end{pmatrix}$$
$$\Delta\left(\frac{\mathbb{1}_{NS}}{\varphi(b,D)}\right)J_\lambda\Delta(\Psi(b,D,\lambda)) = \Delta(\Phi(b,D,\lambda))J_\lambda$$

Hence, in (SM4):

$$\Phi_b = [\partial_b\varphi_{NS}(b,D)]^\top\Delta(\Phi(b,D,\lambda))[J_\lambda I_{N,S} - I_{NS}]\Delta\left(\frac{\mathbb{1}_N}{\varphi_{NS}(b,D)}\right)$$

SM1.5. Computation of Φ_D. Let $i \in \{1,\dots\}$. $\forall s \neq i$, the only dependency in d_i of $\Phi^s(b,D,\lambda)$ resides in Ψ (see (17)), hence:

$$\forall s \neq i, \frac{\partial\Phi^s}{\partial d_i} = \Delta\left(\frac{\mathbb{1}_N}{\varphi(b_s,d_s)}\right)\partial_{d_i}\Psi$$
$$\overset{\text{(SM3)}}{=} \lambda_i\frac{\Delta(\Psi(B,D,\lambda))}{\Delta(\varphi(b_s,d_s))\Delta(\varphi(b_i,d_i))}K^\top\Delta\left(\frac{\mathbb{1}_N}{Kb_i}\right)$$
$$\overset{(17)}{=} \lambda_i\frac{\Delta(\Phi^i(B,D,\lambda))}{\Delta(\varphi(b_s,d_s))}K^\top\Delta\left(\frac{\mathbb{1}_N}{Kb_i}\right)$$

As for $s = i$, we have:

$$\Phi^i(b,D,\lambda) = \frac{\Psi(b,D,\lambda)}{K^\top\frac{d_i}{Kb_i}}$$
$$\Longrightarrow \frac{\partial\Phi^i}{\partial d_i}(b,D,\lambda) = \Delta\left(\frac{\mathbb{1}_N}{\varphi(b_1,d_1)}\right)\partial_D\Psi(b,D,\lambda) - \frac{\Delta(\Psi(b,D,\lambda))}{\Delta(\varphi_i(b_i,d_i)^2)}\partial_{d_i}\varphi(b_i,d_i)$$
$$= \Delta\left(\frac{\mathbb{1}_N}{\varphi(b_1,d_1)}\right)\partial_D\Psi(b,D,\lambda) - \frac{\Delta(\Phi^i(b,D,\lambda))}{\Delta(\varphi(b_i,d_i))}K^\top\left(\frac{\mathbb{1}_N}{Kb_i}\right)$$
$$= (\lambda_i - 1)\frac{\Delta(\Phi^i(b,D,\lambda))}{\Delta(\varphi(b_i,d_i))}K^\top\Delta\left(\frac{\mathbb{1}_N}{Kb_i}\right)$$

Algorithm SM1 `HeavyballSinkhorn`: Computation of approximate Wasserstein barycenters with acceleration

Inputs: Data $x \in \Sigma_N$, atoms $d_1, \dots, d_S \in \Sigma_N$, weights $\lambda \in \Sigma_S$,
extrapolation parameter $\tau \leq 0$

$\forall s, b_s^{(0)} := \mathbf{1}_N$

for $l = 1$ to L step 1 do

$\quad \forall s, \tilde{a}_s^{(l)} := \frac{d_s}{K b_s^{(l-1)}}$

$\quad \forall s, a_s^{(l)} := \left(a_s^{(l-1)}\right)^{\tau} \left(\tilde{a}_s^{(l)}\right)^{1-\tau}$

$\quad p := \prod_s \left(K^\top a_s^{(l)}\right)^{\lambda_s}$

$\quad \forall s, \tilde{b}_s^{(l)} := \frac{p}{K^\top a_s^{(l)}}$

$\quad \forall s, b_s^{(l)} := \left(b_s^{(l-1)}\right)^{\tau} \left(\tilde{b}_s^{(l)}\right)^{1-\tau}$

od

Outputs: $P^{(L)}(D, \lambda) := p$

Algorithm SM2 `GeneralizedSinkhorn`: Computation of unbalanced barycenters with acceleration

Inputs: Data $x \in \Sigma_N$, atoms $d_1, \dots, d_S \in \Sigma_N$, weights $\lambda \in \Sigma_S$,
extrapolation parameter $\tau \leq 0$, KL parameter $\rho > 0$

$\forall s, b_s^{(0)} := \mathbf{1}_N$

for $l = 1$ to L step 1 do

$\quad \forall s, \tilde{a}_s^{(l)} := \left(\frac{d_s}{K b_s^{(l-1)}}\right)^{\frac{\rho}{\rho+\gamma}}$

$\quad \forall s, a_s^{(l)} := \left(a_s^{(l-1)}\right)^{\tau} \left(\tilde{a}_s^{(l)}\right)^{1-\tau}$

$\quad p := \left(\sum_{s=1}^S \lambda_s \left(K^\top a_s^{(l)}\right)^{\frac{\gamma}{\rho+\gamma}}\right)^{\frac{\rho+\gamma}{\gamma}}$

$\quad \forall s, \tilde{b}_s^{(l)} := \left(\frac{p}{K^\top a_s^{(l)}}\right)^{\frac{\rho}{\rho+\gamma}}$

$\quad \forall s, b_s^{(l)} := \left(b_s^{(l-1)}\right)^{\tau} \left(\tilde{b}_s^{(l)}\right)^{1-\tau}$

od

Outputs: $P^{(L)}(D, \lambda) := p$

SM2. Generalized barycenters.

SM3. Additional results.

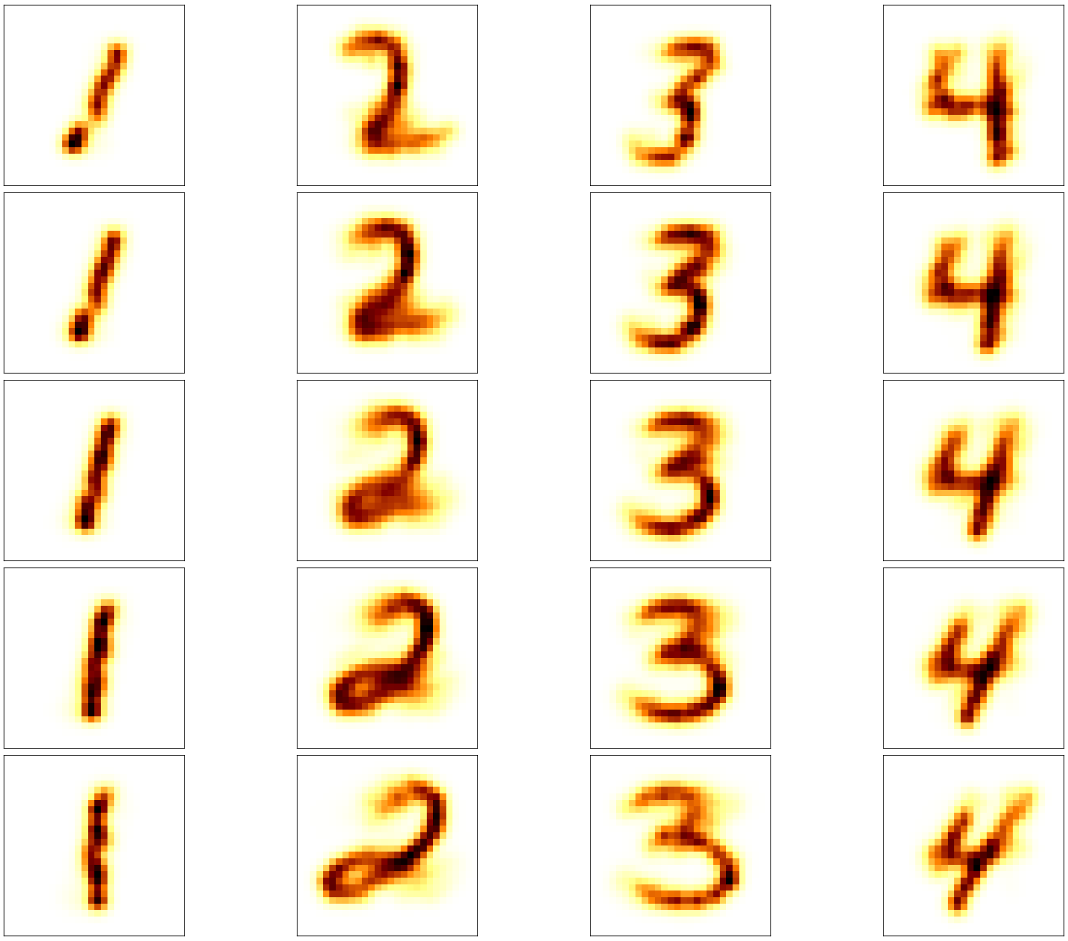

Figure SM1: Span of our 2-atoms dictionary for weights $(1-t,t), t \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ when trained on images of digits $1, 2, 3, 4$. See the first columns of Figure C.1 for comparison with first WPGs.

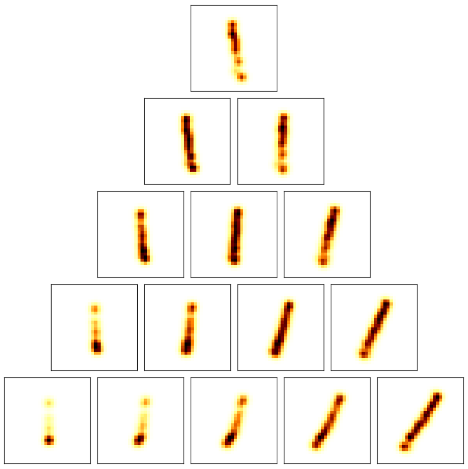

Figure SM2: Same as Figure 6 when training on images of the digit 1.

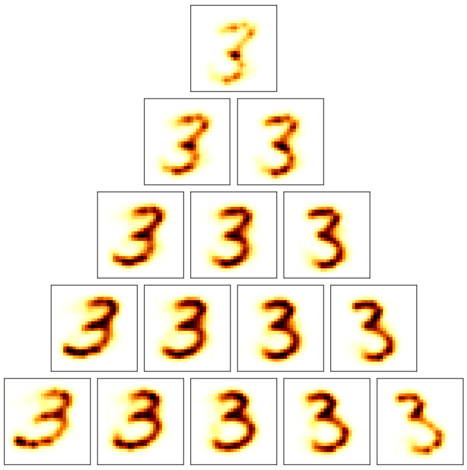

Figure SM3: Same as Figure 6 when training on images of the digit 3.

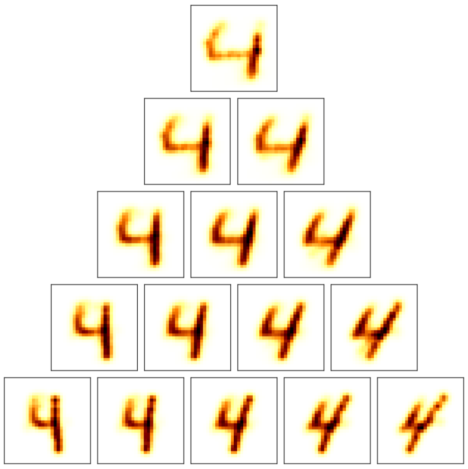

Figure SM4: Same as Figure 6 when training on images of the digit 4.

SM3.1. MNIST and Wasserstein Geodesics.

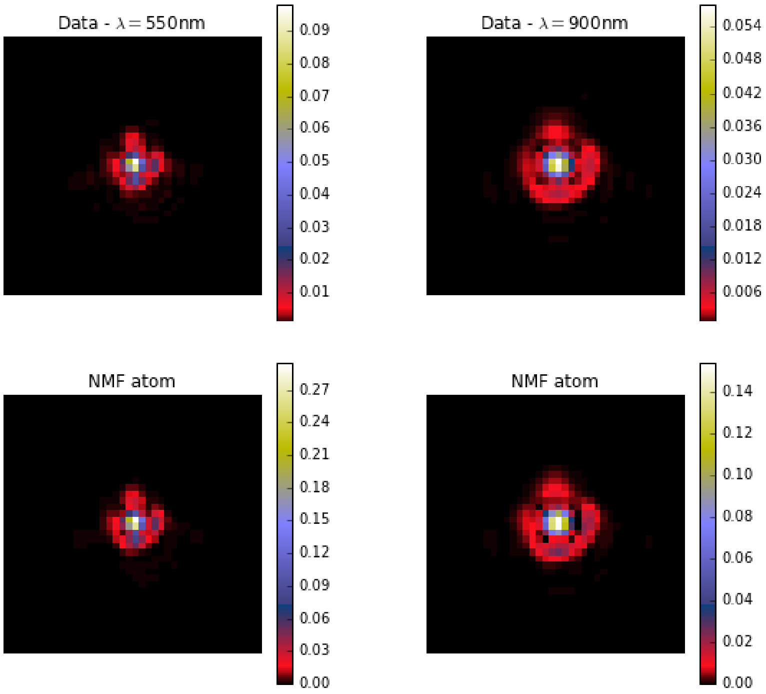

Figure SM5: Extreme wavelength PSFs in the dataset and atoms learned from NMF. See Figure 9 for those learned using our method.

SM3.2. Point Spread functions.

SM3.3. Wasserstein faces.

REFERENCES

[1] M. A. Schmitz, M. Heitz, N. Bonneel, F. Ngolè, D. Coeurjolly, M. Cuturi, G. Peyré, and J.-L. Starck, *Optimal transport-based dictionary learning and its application to euclid-like point spread function representation*, in SPIE Optical Engineering+ Applications, International Society for Optics and Photonics, 2017.

[2] M. Turk and A. Pentland, *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience, 3 (1991), pp. 71–86.

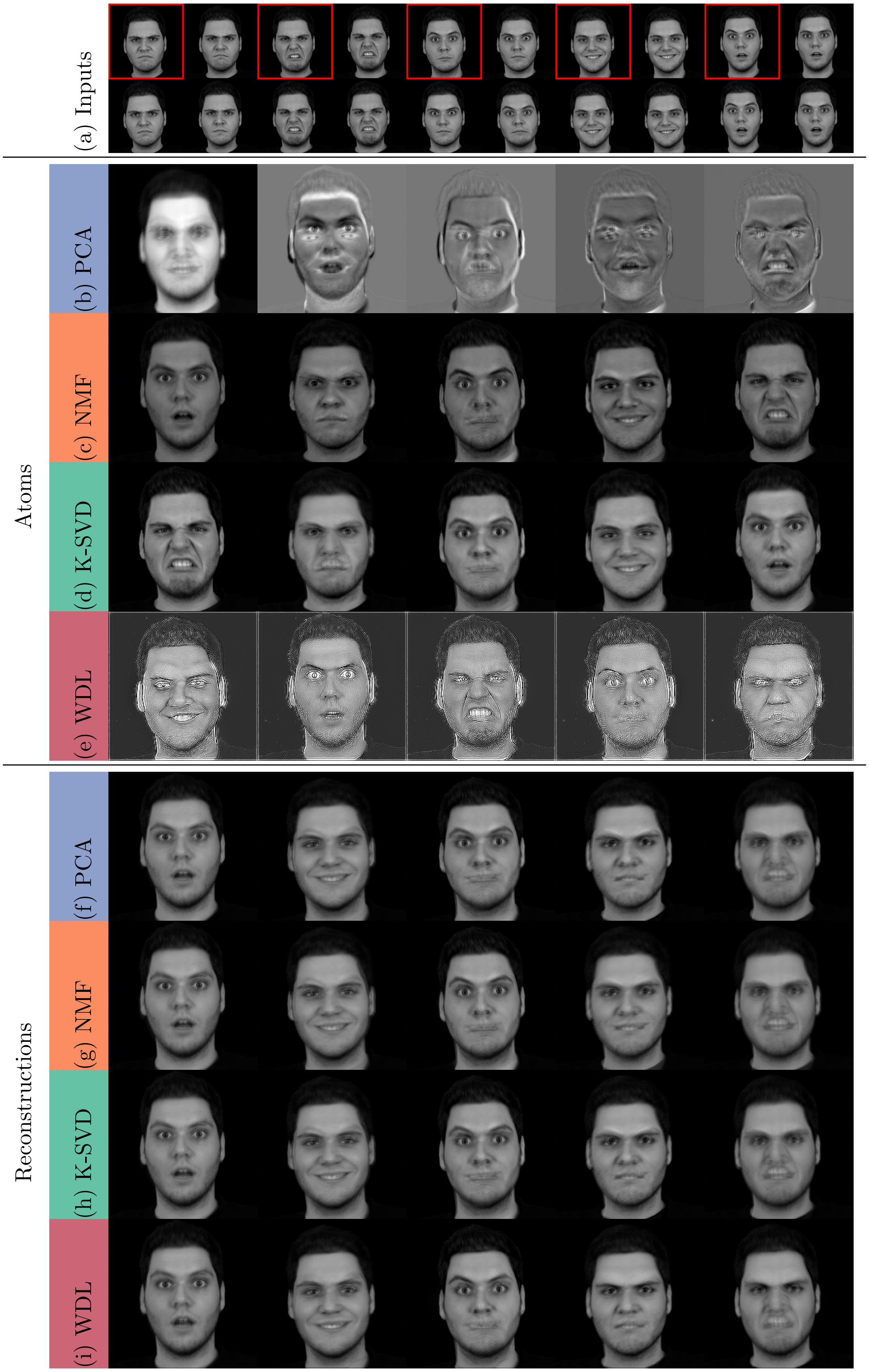

Figure SM6: Similarly to Figure 13, we compare our method to the Eigenfaces [SM2] approach, NMF and K-SVD as a tool to represent faces on a low dimensional space.

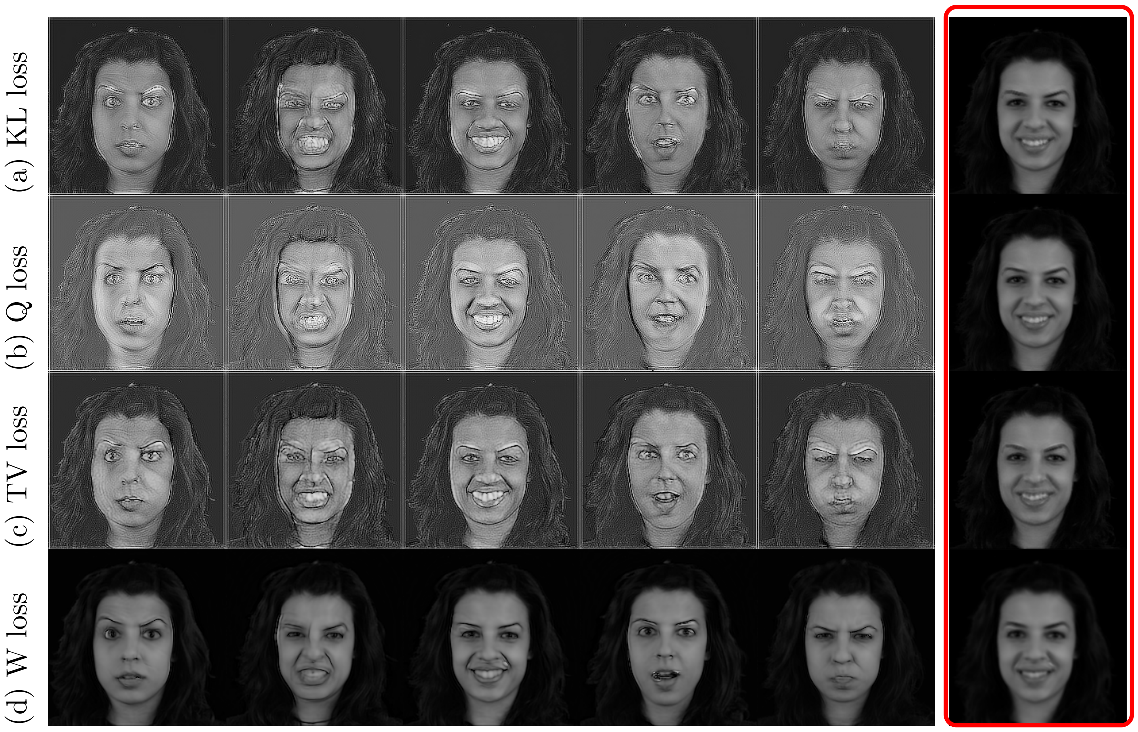

Figure SM7: Similarly to Figure 14, we compare the atoms obtained using different loss functions, ranking them by mean PSNR: (a) $\overline{PSNR} = 33.81$, (b) $\overline{PSNR} = 33.72$, (c) $\overline{PSNR} = 32.95$ and (d) $\overline{PSNR} = 32.34$