



HAL
open science

Compressive learning for patch-based image denoising

Hui Shi, Yann Traonmilin, Jean-François Aujol

► **To cite this version:**

Hui Shi, Yann Traonmilin, Jean-François Aujol. Compressive learning for patch-based image denoising. 2022. hal-03429102v2

HAL Id: hal-03429102

<https://hal.science/hal-03429102v2>

Preprint submitted on 14 Feb 2022 (v2), last revised 21 Apr 2022 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Compressive learning for patch-based image denoising*

Hui Shi[†], Yann Traonmilin[†], and Jean-François Aujol[†]

Abstract. The Expected Patch Log-Likelihood algorithm (EPLL) and its extensions have shown good performances for image denoising. The prior model used by EPLL is usually a Gaussian Mixture Model (GMM) estimated from a database of image patches. Classical mixture model estimation methods face computational issues as the high dimensionality of the problem requires training on large datasets. In this work, we adapt a compressive statistical learning framework to carry out the GMM estimation. With this method, called *sketching*, we estimate models from a compressive representation (the *sketch*) of the training patches. The cost of estimating the prior from the sketch no longer depends on the number of items in the original large database. To accelerate further the estimation, we add another dimension reduction technique (low-rank modeling of the covariance matrices) to the compressing learning framework. To demonstrate the advantages of our method, we test it on real large-scale data. We show that we can produce denoising performances similar to performances obtained with models estimated from the original training database using GMM priors learned from the sketch with improved execution times.

Key words. Image denoising, Compressive learning, Sketching, Optimization,

AMS subject classifications. 68U10, 94A08, 49N30

1. Introduction. We consider the classical noisy observation model of a clean natural image $u \in \mathbb{R}^N$ (composed of N pixels):

$$(1.1) \quad v = u + w$$

where v is the observed degraded version of u . The acquisition noise w is usually assumed to be an additive white Gaussian noise of variance σ , i.e. $w \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2 I_N)$. In the last two decades, non local patch-based methods have been proven successful for denoising. Methods such as Piecewise Linear Estimators [60, 1], BM3D [10, 29] or NL-Bayes [31, 30, 58, 28] are examples of non-local methods [3]. In patch-based image denoising, the noisy image v is divided into small patches $\{v_i\}_{i=1}^M$. Each patch $v_i \in \mathbb{R}^P$ (P is the patch size) can be seen as a vector in a high dimensional space. The denoising problem is considered on each patch:

$$(1.2) \quad v_i = u_i + w_i,$$

and a corresponding denoised version u_i^* of the true values u_i are estimated. To overcome the ill-posedness of this inverse problem, various denoising methods [32, 31, 30, 22] consider patch models within a Bayesian framework. According to the Bayes' theorem, the objective is to find u_i^* which maximizes the posterior probability distribution $f(u_i|v_i)$ under the prior

*Submitted to the editors DATE.

Funding: This work was funded by the French National Research Agency (ANR) under reference ANR-20-CE40-0001 (EFFIREG project).

[†]Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 TALENCE, FRANCE. ({hui.shi, yann.traonmilin, jean-francois.aujol}@math.u-bordeaux.fr).

34 $p(u_i)$. The Maximum A Posteriori (MAP) problem is formulated as

$$35 \quad (1.3) \quad u_i^* = \arg \max_{u_i \in \mathbb{R}^P} f(u_i|v_i) = \arg \max_{u_i \in \mathbb{R}^P} f(v_i|u_i)p(u_i) \propto \arg \max_{u_i \in \mathbb{R}^P} e^{-\frac{\|u_i - v_i\|^2}{2\sigma^2}} p(u_i)$$

36 where $\|\cdot\|$ denotes the ℓ^2 -norm. This yields

$$37 \quad (1.4) \quad u_i^* = \arg \min_{u_i \in \mathbb{R}^P} \frac{\|u_i - v_i\|^2}{2\sigma^2} - \log(p(u_i)).$$

38 Ideally, the choice of the prior distribution should be determined by the nature of the
39 image to be estimated. In practice, Gaussian Mixture Models (GMM) [60, 56, 22] have shown
40 their effectiveness. With the GMM prior, the solution of problem (1.4) can be approximated
41 by a Wiener filter solution (see subsection 2.2).

42 Among these various non-local denoising methods, the Expected Patch Log-Likelihood
43 algorithm (EPLL) [63] occupies a central position due to its efficient denoising performance.
44 A large number of works build on the original EPLL formulation to deal with more general
45 prior or go beyond the denoising problem [12, 37, 6, 33, 41, 52, 11, 44]. EPLL uses a GMM
46 prior learned from a very large set of patches extracted from clean images. The key to the
47 success of EPLL is to find a good prior distribution. Since in practice patch sizes are typically
48 greater than 5×5 , estimating prior distributions in such a high-dimensional space is a difficult
49 task. Moreover, to estimate the best possible model, we need to maximize the redundancy
50 of structural information and use training databases as large as possible. As the traditional
51 empirical minimization approaches require access to the whole training dataset, when the
52 collection size is large, the learning process can be extremely costly. For instance, in the case
53 of the classical learning method Expectation Maximization (EM), the memory consumption
54 and computation time depend on the size of the database (see section 3).

55 Leveraging ideas from compressive sensing [15] and streaming algorithms [9], R. Gribon-
56 val et al. propose a *sketching* method [25, 19, 20, 18, 17] to compress the training database.
57 This scalable technique compress the whole training collection into a fixed-size representation
58 (a vector): a *sketch* of the training dataset before learning. The sketch captures the nec-
59 essary information for the considered learning task. For certain mixture model estimation,
60 it is then possible to learn their parameters directly from the sketch, without access to the
61 original dataset. Hence the space and time complexity of the learning algorithm no longer
62 depends on the original database size, but only on the size of the sketch which is linked to the
63 dimensionality of the model. Sketching has been already used successfully in machine learn-
64 ing [45, 17, 27, 7, 5, 40], generative networks [46], source localization [13, 14], independent
65 component analysis [48] and depth imaging [49]. In [25], the sketching is implemented and
66 evaluated on synthetic data to estimate a GMM with diagonal covariances. It is shown that
67 on large synthetic data, for the estimation of GMM, the sketching produces precise results
68 while requiring fewer memory space and computations. In this work, we explore the sketching
69 method in the image patches context where GMM with full covariance must be estimated
70 from the compressed database.

71 Due to the curse of dimensionality, it is computationally expensive to manipulate the
72 GMMs' covariance matrices. In [42], the authors show that most natural images and videos

73 can be represented by a GMM with low-rank covariance matrices. The experiments have also
 74 shown the efficiency of low-rank covariance matrices applied to image denoising [38], image
 75 inpainting, high-speed video and hyperspectral imaging [59]. This motivates us to use such
 76 low-rank covariances in the GMM modeling of patches and extend the sketching framework
 77 accordingly to gain computational speedup and to manage the modeling of the image patches
 78 in the most possible flexible way.

79 **1.1. Contributions.** A preliminary and short version of this work has appeared in [51]. In
 80 this paper, we provide a more detailed version of this work with a final consolidated version
 81 of the proposed learning algorithm, validated by extended numerical experiments.

82 **Figure 1** summarizes the principle of our approach. We first construct a sketch by averaging
 83 random Fourier features computed over the whole image patch database. Then the model
 84 parameters are learned directly from the sketch by our Low-rank Continuous Orthogonal
 85 Matching Pursuit (LR-COMP) algorithm without access to the original database. Finally,
 the learned model is used with a Bayesian method (EPLL) for the denoising task. Our

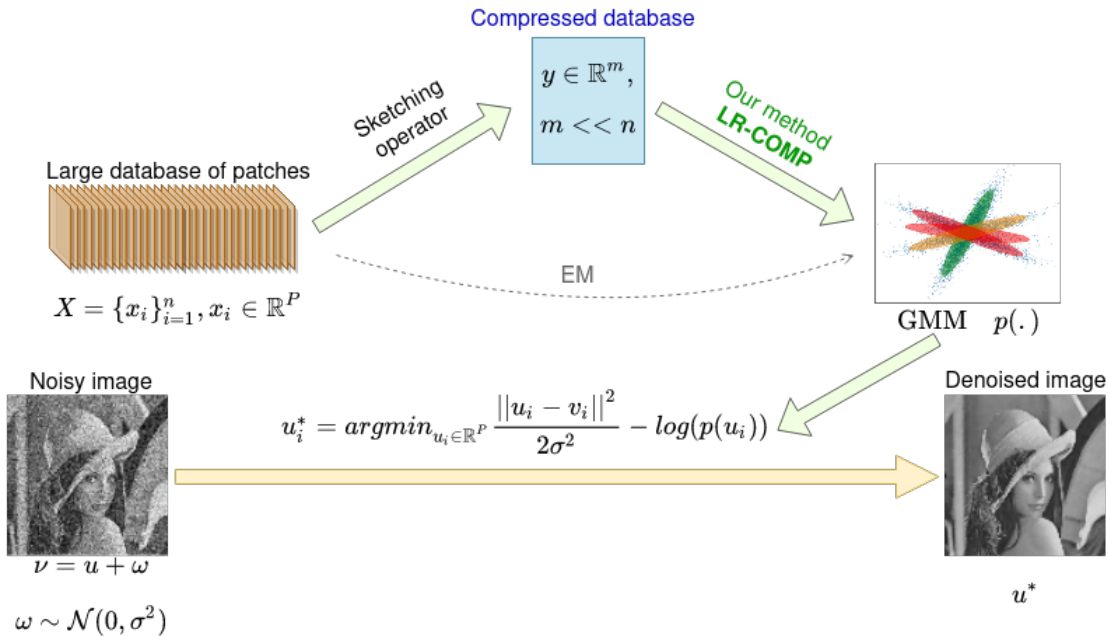


Figure 1. A summary of our method

86 contributions of this piece of work are the following:

- 88 • In this work, we propose an algorithm LR-COMP to estimate a GMM with non-
 89 diagonal and low-rank covariance matrices. Compared to previous work in [25], our
 90 extension to non-diagonal covariance matrices allows us to learn a GMM prior from
 91 a compressed database of patches in the context of image denoising. Moreover, with
 92 the low-rank approximation of the covariance matrices, we lighten the computation
 93 burden in the denoising process while keeping good denoising performances.

- We demonstrate the performance of our approach on real large-scale data (over 4 millions training samples of patch size of 7×7) for the task of patch-based image denoising. We show that using models trained with the compressed database, we can obtain similar denoising performances compared to the models obtained with the classical EM algorithm. To the best of our knowledge, this is also the first time that the sketching framework has been applied with such high dimensional GMMs.
- Computationally, we estimate the model from a compressed database which is about 1000 times smaller than the original patch database. It leads to running time approximately two times faster compared to the EM method.

The paper is organized as follows. [Section 2](#) is a reminder of the EPLL framework. Then we review the EM algorithm in [section 3](#). In [section 4](#), we explain the compressive learning method. In [section 5](#), we focus on explaining how to adapt the sketching framework to learn a GMM in the image patch context. We also interpret the extension to low-rank covariances and the implementation details of the adapted learning algorithm LR-COMP. In [section 6](#), we provide numerical experiments that demonstrate the performance of our approach. Some conclusions and tracks for further works follow in [section 7](#).

2. Image denoising with EPLL. We review in this section the Expected Patch Log-Likelihood (EPLL) framework for image denoising. EPLL is a patch-based image restoration algorithm introduced by Zoran and Weiss [63]. The EPLL framework restores an image u by performing the following maximum a posteriori (MAP) estimation over all N patches:

$$(2.1) \quad u^* = \arg \min_{u \in \mathbb{R}^N} \frac{P}{2\sigma^2} \|u - v\|^2 - \sum_{i=1}^N \log(p(\mathcal{P}_i u))$$

where $\mathcal{P}_i : \mathbb{R}^N \rightarrow \mathbb{R}^P$ is a linear operator that extracts a patch of P pixels centered at the position i , typically $P = 7 \times 7$. The function $p(\cdot)$ is the density of the prior probability distribution of the patches. Note that in practice, we assume that patches are distributed independently.

2.1. Optimization. Due to the non-convexity of $p(\cdot)$, direct optimization of the problem may be difficult. The authors of EPLL propose to perform the optimization with “half-quadratic splitting” [16]. By introducing N auxiliary unknown vectors $z_i \in \mathbb{R}^P$ and a denoising parameter $\beta > 0$, the problem is then considered as:

$$(2.2) \quad u^* = \arg \min_{\substack{u \in \mathbb{R}^N \\ z_1, \dots, z_N \in \mathbb{R}^P}} \frac{P}{2\sigma^2} \|u - v\|^2 + \frac{\beta}{2} \sum_{i=1}^N \|\mathcal{P}_i u - z_i\|^2 - \sum_{i=1}^N \log(p(z_i)).$$

The optimization (2.2) is accomplished by alternating the minimization of u and z_i .

- **Solving u for fixed z_i** — Problem (2.2) turns into a linear inverse problem with the

126 Tikhonov regularization. It has a closed form solution:

$$\begin{aligned}
 \hat{u} &= \arg \min_{u \in \mathbb{R}^N} \frac{P}{2\sigma^2} \|u - v\|^2 + \frac{\beta}{2} \sum_{i=1}^N \|\mathcal{P}_i u - z_i\|^2 \\
 (2.3) \quad &= (I + \frac{\beta\sigma^2}{P} \sum_{i=1}^N \mathcal{P}_i^T \mathcal{P}_i)^{-1} (v + \frac{\beta\sigma^2}{P} \sum_{i=1}^N \mathcal{P}_i^T z_i)
 \end{aligned}$$

128 with $\sum_{i=1}^N \mathcal{P}_i^T \mathcal{P}_i = PI$, where P is the number of patches overlapping each pixel.
 129 Hence we have

$$(2.4) \quad \hat{u} = (I + \sigma^2 \beta I)^{-1} (v + \sigma^2 \beta \bar{z}_i)$$

131 where $\bar{z}_i := (\sum_{i=1}^N \mathcal{P}_i^T \mathcal{P}_i)^{-1} \sum_{i=1}^N \mathcal{P}_i^T z_i = \frac{1}{P} \sum_{i=1}^N \mathcal{P}_i^T z_i$ is the image after averaging
 132 all overlapping patches z_i .

133 • **Solving z_i for fixed u** — The minimization problem (2.2) is separable with respect
 134 to the latent variable z_i . It means that for each z_i we solve a patch MAP estimation
 135 under the patch prior $p(z_i)$, i.e. for all i ,

$$(2.5) \quad \hat{z}_i = \arg \min_{z_i \in \mathbb{R}^P} \frac{\beta}{2} \|\mathcal{P}_i \hat{u} - z_i\|^2 - \log(p(z_i)).$$

137 The solution of this problem depends on the choice of patch prior $p(\cdot)$.

138 **2.2. Denoising with a GMM prior.** EPLL assumes that the prior is a finite Gaussian
 139 mixture model (GMM) with zero-mean on centered patches: the empirical mean estimated
 140 from noisy patches are removed before the denoising process (2.5) and added back in the end.
 141 We consider that a zero-mean patch $x \in \mathbb{R}^P$ is a random vector generated from a distribution
 142 with density $p(x)$ defined as

$$(2.6) \quad p(x) = \sum_{k=1}^K \alpha_k \mathcal{N}_P(x; 0, \Sigma_k)$$

144 where K is the number of Gaussian components and $\alpha_k \geq 0$ are weights of each component
 145 such that $\sum_{k=1}^K \alpha_k = 1$. The function $\mathcal{N}_P(x; 0, \Sigma_k)$ denotes the density of a Gaussian dis-
 146 tribution with zero-mean with covariance $\Sigma_k \in \mathbb{R}^{P \times P}$. Recall that the zero-mean Gaussian
 147 distribution density is:

$$(2.7) \quad \mathcal{N}_P(x; 0, \Sigma_k) = \frac{1}{(2\pi)^{P/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2} x^T \Sigma_k^{-1} x}.$$

149 Hence, under the GMM prior, Problem (2.5) turns to:

$$(2.8) \quad \hat{z}_i = \arg \min_{z_i \in \mathbb{R}^P} \frac{\beta}{2} \|\mathcal{P}_i \hat{u} - z_i\|^2 - \log\left(\sum_{k=1}^K \alpha_k \mathcal{N}_P(z_i; m_i, \Sigma_k)\right)$$

151 where we supposed that the mean m_i are correctly estimated by the empirical mean of noisy
152 patches.

153 This problem cannot be solved in closed form as the second term is the logarithm of a sum
154 of exponential. In [63], the authors proposed to solve this problem by keeping only one Gauss-
155 ian component. For a given centered patch $\tilde{z}_i = \mathcal{P}_i \hat{u} - m_i$, we chose the component k_i^* that
156 maximizes the posterior probability $p(k_i|\tilde{z}_i)$. This leads to computationally efficient imple-
157 mentations. [54] also justified that only one component is required for good reconstructions.
158 The index k_i^* is chosen by

$$\begin{aligned} k_i^* &= \arg \max_{1 \leq k_i \leq K} p(k_i|\tilde{z}_i) = \arg \max_{1 \leq k_i \leq K} p(k_i)p(\tilde{z}_i|k_i) \\ (2.9) \quad &= \arg \min_{1 \leq k_i \leq K} -2 \log \alpha_{k_i} + \log \left| \Sigma_{k_i} + \frac{1}{\beta} I_P \right| + \tilde{z}_i^\top (\Sigma_{k_i} + \frac{1}{\beta} I_P)^{-1} \tilde{z}_i \end{aligned}$$

160 where α_{k_i} and Σ_{k_i} are the weights and the covariance matrices of the k th Gaussian component
161 for the given patch \tilde{z}_i . With k_i^* (instead of a sum of K components), the solution of (2.8) is
162 then a Wiener filtering solution:

$$(2.10) \quad \hat{z}_i = (\Sigma_{k_i^*} + \frac{1}{\beta} I_P)^{-1} \Sigma_{k_i^*} \tilde{z}_i + m_i.$$

164 **2.3. Eigenspace implementation of EPLL.** The matrix inversions in (2.9) and (2.10) can
165 be done efficiently by using the singular value decomposition over the covariance matrices.
166 We denote $\Sigma_k = U_k \Lambda_k U_k^T$, with $U_k \in \mathbb{R}^{P \times P}$ an unitary matrix and $\Lambda_k = \text{diag}(\lambda_1^{(k)}, \dots, \lambda_P^{(k)})$ a
167 diagonal matrix. The diagonal entries $\lambda_j^{(k)}$ of Λ_k are the singular values of Σ_k . Then we can
168 compute (2.9) by:

$$(2.11) \quad k_i^* = \arg \min_{1 \leq k \leq K} -2 \log \alpha_k + \sum_{j=1}^P \left(\log(\lambda_j^{(k)} + \frac{1}{\beta}) + \frac{[\tilde{v}_i^{(k)}]_j^2}{\lambda_j^{(k)} + \frac{1}{\beta}} \right)$$

170 where

$$(2.12) \quad \tilde{v}_i^{(k)} = U_k^T \tilde{z}_i.$$

172 Then (2.10) leads to

$$(2.13) \quad \hat{z}_i = U_{k_i^*} S_{k_i^*} U_{k_i^*}^T \tilde{z}_i + m_i = U_{k_i^*} S_{k_i^*} \tilde{v}_i^{(k_i^*)} + m_i$$

174 with

$$(2.14) \quad S_{k_i^*} = \text{diag} \left(\frac{\lambda_j^{(k_i^*)}}{\lambda_j^{(k_i^*)} + \frac{1}{\beta}} \right)_{j=1, \dots, P}.$$

176 **3. Learning a GMM with EM.** The Expectation-Maximization (EM) algorithm is a clas-
 177 sical mixture estimation approach. This algorithm starts with some initial estimates of model
 178 parameters and then iteratively updates the estimate until the the estimates are not changing
 179 much. See [Appendix B](#) for the details of the EM algorithm. In each iteration, it carries out
 180 two steps: the E-Step (expectation step) and the M-Step (maximization step). In E-Step,
 181 using the current estimate of the parameters, we evaluate the posterior probabilities. In the
 182 M-Step we compute parameters that maximize the probabilities found on the E-Step. These
 183 estimated parameters are then used to determine the distribution of the latent variables in
 184 the next E-Step.

185 As for the time complexity of one iteration of this algorithm, it is linear in the number of
 186 model components K and the number of elements in the database n . However it is cubic with
 187 respect to the dimensions P due to the fact that we need to inverse the covariance matrix when
 188 calculating the density in E-Step. Thus, when estimating a K -components GMM on a data-
 189 base of n elements of dimension P , the computational complexity of one iteration of the EM
 190 algorithm is $\mathcal{O}(nKP^2 + KP^3)$. The major criticism of the EM algorithm is that when dealing
 191 with a large dataset, it often converges slowly. To address this problem, researchers have
 192 developed various variations of the traditional EM algorithm [36, 36]. Learning parameters
 193 using EM technique face computational issues linked to the size of the dataset and the number
 194 of parameters to estimate, which would make the use of (very) large image patches databases
 195 impractical. In the next section we will see an alternative manner to learn parameters using
 196 compressive learning.

197 **4. Sketching.** Sketching is a dimensionality reduction method. The principle is to com-
 198 press the whole dataset massively before learning. First, the dataset $\chi = \{x_i\}_{i=1}^n$ is summa-
 199 rized into a vector $y \in \mathbb{C}^m$ ($m \ll n$) called the *sketch*:

$$200 \quad (4.1) \quad y := \text{Sketch}(\chi).$$

201 Note that the computation of the sketch can be performed in a distributed manner. Then
 202 we apply a learning procedure Υ that allows us to learn an estimate Ψ^* of some statistical
 203 parameters Ψ of the dataset directly from the sketch \hat{y} , namely

$$204 \quad (4.2) \quad \Psi^* = \Upsilon(\hat{y}) = \Upsilon(\text{Sketch}(\chi))$$

205 More specifically, learning from the sketch corresponds to a minimization problem

$$206 \quad (4.3) \quad \Psi^* \in \arg \min_{\Psi} E(\hat{y}, \Psi)$$

207 where the energy of the model $E(\cdot, \cdot)$ quantifies the fit between the sketch \hat{y} and the parameter
 208 Ψ . In the context of statistical learning, the energy E can be seen as a proxy of the empirical
 209 risk. The principle of sketching is summarized in [Fig. 2](#).

210 **4.1. Compressive mixture estimation.** In machine learning, the data $x_i \in \mathbb{R}^d$ (e.g. in our
 211 case, the patches with $d = P$) are often modeled as i.i.d. random samples generated from a
 212 probability distribution parameterized by Θ with a density $f_{\Theta} \in \mathcal{D}$ (\mathcal{D} is the set of probability
 213 measures over \mathbb{R}^d). The idea of sketching is to project the measure f_{Θ} on a low-dimensional

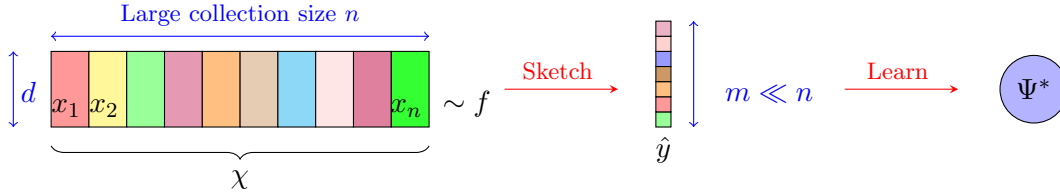


Figure 2. Schema of sketching

214 vector space while keeping all the necessary information of the dataset. Mathematically, given
 215 a linear sketching operator \mathcal{S} :

$$216 \quad (4.4) \quad \begin{aligned} \mathcal{S} : \mathcal{D} &\longrightarrow \mathbb{C}^m \\ y &= \mathcal{S}f \end{aligned}$$

217 and for some finite $K \in \mathbb{N}^*$, we define a K -sparse model $f_{\Theta, \alpha} \in \mathcal{D}$:

$$218 \quad (4.5) \quad f_{\Theta, \alpha} = \sum_{k=1}^K \alpha_k f_{\theta_k}$$

219 where $f_{\theta_k} \in \mathcal{D}$ are elementary measures parametrized by θ_k , $\alpha_k \geq 0$ for all components and
 220 $\sum_{k=1}^K \alpha_k = 1$. We can express the vector z as

$$221 \quad (4.6) \quad y = \mathcal{S}f_{\Theta, \alpha} = \sum_{k=1}^K \alpha_k \mathcal{S}f_{\theta_k}.$$

222 In practice we only have access to the empirical probability distribution $\hat{f} = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$
 223 where δ_{x_i} is a unit mass at x_i . So we can define the empirical sketch as $\hat{y} = \frac{1}{n} \mathcal{S} \sum_{i=1}^n \delta_{x_i}$.
 224 The goal of the sketching framework is to recover $f_{\Theta, \alpha}$ from y , hence we do the following
 225 minimization to estimate the parameters

$$226 \quad (4.7) \quad (\Theta^*, \alpha^*) \in \underset{\substack{\Theta \in \mathbb{R}^K \\ \alpha \in \mathbb{R}^K, \alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1}}{\arg \min} \quad \|\mathcal{S}f_{\Theta, \alpha} - \hat{y}\|_2^2.$$

227 The objective of sketched learning algorithms is to minimize a datafit functional between the
 228 compressed database and the sketch of the estimation. In other words, our aim is to find
 229 parameters α, Θ such that the sketch of the probability distribution parameterized by α, Θ is
 230 the closest to the empirical sketch \hat{y} .

231 **4.2. Recovery guarantees.** It was shown in [18] that we can guarantee theoretically the
 232 success of this estimation with a condition on the sketch size. These guarantees necessitate
 233 a ‘‘Lower Restricted Isometry Property’’ (LRIP) of the sketching operator. This property, is
 234 verified with high probability, for GMM with sufficiently separated mean and random Fourier
 235 sketching as long as the sketch size $m \geq O(K^2 d \text{polylog}(K, d))$, i.e. when the size of the sketch
 236 essentially depends on the parameters K (the number of components) and d (the model

237 dimension). Empirical results seem to indicate that for d_{tot} the total number of parameters, a
 238 database size of the order of d_{tot} is sufficient: in the case of estimating a GMM with diagonal
 239 covariance matrices [25], the authors observe that the quality of the reconstruction exhibits a
 240 sharp phase-transition with respect to the sketch size m . This phase transition happens for
 241 m proportional to d_{tot} . In our model, $d_{\text{tot}} = K(Pr + 1)$. The excess risk of the GMM learning
 242 task is then controlled by the sum of an empirical error term and a modeling error term. This
 243 guarantees that the estimated GMM approximates well the distribution of the data [19].

244 Note that EPLL uses a zero-mean GMM as the patch prior, therefore, during the learning
 245 process, the patches are centered before sketching and we do not estimate the mean of Gaus-
 246 sians. In our case, the sketched GMM learning problem reduces to the estimation of the sum
 247 of k zero-mean Gaussians with covariances $\Theta = (\Sigma_k)_{k=1}^K$, i.e. $f_{\Theta, \alpha} = \sum_{k=1}^K \alpha_k g_{\Sigma_k}$ where g_{Σ} is
 248 the zero-mean Gaussian measure with covariance Σ . In this context, the notion of separation
 249 used to prove guarantees in [18] does not hold. We still show empirically that the sketching
 250 process is successful without this separation assumption.

251 **4.3. Design of sketching operator: randomly sampling the characteristic function.** In
 252 [25], the sketch is a sampling of the characteristic function (i.e. the Fourier transform of the
 253 probability distribution f). Recall that the characteristic function ψ_f of a measure f is defined
 254 as:

$$255 \quad (4.8) \quad \psi_f(\omega) = \int_{\mathbb{R}^d} e^{-i\omega^T x} df(x) \quad \forall \omega \in \mathbb{R}^d.$$

256 The sketching operator is therefore expressed as:

$$257 \quad (4.9) \quad \mathcal{S}f = [\psi(\omega_1), \dots, \psi(\omega_m)]^T$$

258 where $\{\omega_1, \dots, \omega_m\}$ is a set of well chosen frequencies. In the spirit of Random Fourier Sam-
 259 pling, the authors of [25] propose to draw the frequencies from a probability distribution, i.e.
 260 $(\omega_1, \dots, \omega_m) \stackrel{i.i.d.}{\sim} \Delta$. The choice of frequencies is essential to the success of sketching, and we
 261 will discuss it in details in [subsection 5.1](#).

262 **5. Sketching image patches.** In this section, we adapt the sketching framework to the
 263 context of image patches. As when using the classical EM algorithm, the GMM learning from
 264 sketch is performed under the assumption that the training patches are i.i.d. Given a training
 265 set of n centered patches $\chi = \{x_1, \dots, x_n\} \subset \mathbb{R}^P$, we define the empirical characteristic function
 266 with

$$267 \quad (5.1) \quad \hat{\psi}(w) = \frac{1}{n} \sum_{j=1}^n e^{-i\omega^T x_j} \quad \text{with} \quad \omega \in \mathbb{R}^P.$$

268 Thus the empirical sketch \hat{y} is expressed as

$$269 \quad (5.2) \quad \hat{y} = [\hat{\psi}(\omega_1), \dots, \hat{\psi}(\omega_m)]^T = \frac{1}{n} \left[\sum_{j=1}^n e^{-i\omega_1^T x_j}, \dots, \sum_{j=1}^n e^{-i\omega_m^T x_j} \right]^T.$$

270 In other words, a sample of the sketched database is a P -dimensional frequency component
 271 calculated by averaging over patches (not to be mixed with usual 2D Fourier components of
 272 images). Thanks to the properties of the Fourier transform of Gaussians, the sketch of a single
 273 zero-mean Gaussian component g_{Σ_k} at frequency ω_l is

$$274 \quad (5.3) \quad (\mathcal{S}(g_{\Sigma_k}))_l = \psi_{g_{\Sigma_k}}(\omega_l) = e^{-\frac{1}{2}\omega_l^T \Sigma_k \omega_l}.$$

275 Thus, given the weights $\alpha = (\alpha_k)_{k=1}^K$ and the covariance matrices $\Sigma = (\Sigma_k)_{k=1}^K$, the sketch of
 276 a zero-mean GMM $f_{\Sigma, \alpha} = \sum_{k=1}^K \alpha_k g_{\Sigma_k}$ is

$$277 \quad (5.4) \quad z = [\mathcal{S}(f_{\Sigma, \alpha})_l]_{l=1, \dots, m} = \left[\sum_{k=1}^K \alpha_k e^{-\frac{1}{2}\omega_l^T \Sigma_k \omega_l} \right]_{l=1, \dots, m}.$$

278 As a consequence, denoting PSD_P the set of $P \times P$ positive symmetric definite matrices, the
 279 problem (4.7) of estimating GMM parameters becomes

$$280 \quad (5.5) \quad (\Sigma^*, \alpha^*) \in \underset{\substack{\Sigma = (\Sigma_k)_{k=1}^K, \Sigma_k \in PSD_P \\ \alpha \in \mathbb{R}^K, \alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1}}{\arg \min} \|\hat{y} - \mathcal{S}f_{\Sigma, \alpha}\|_2^2,$$

281 i.e.

$$282 \quad (5.6) \quad (\Sigma^*, \alpha^*) \in \underset{\substack{\Sigma_k \in PSD_P, \forall k \\ \alpha \in \mathbb{R}^K, \alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1}}{\arg \min} \left| \frac{1}{n} \sum_{j=1}^n e^{-i\omega_l^T x_j} - \sum_{k=1}^K \alpha_k e^{-\frac{1}{2}\omega_l^T \Sigma_k \omega_l} \right|^2.$$

283 In practice, the positive definite constraint in the optimization problem is hard to enforce
 284 directly on the space of $P \times P$ matrices (as previous work only considered diagonal covariances,
 285 it was not an issue). Our method, based on the Burer-Monteiro, permits us to respect the
 286 PSD constraint by recasting the covariance estimation problem as an optimization over $\mathbb{R}^{P \times P}$
 287 without constraint (see subsection 5.2 for more details).

288 **5.1. Frequency sampling.** The design of the probability distribution Δ for sampling the
 289 frequencies $\{\omega_1, \dots, \omega_m\}$ is essential to the success of sketching. In our work, we draw fre-
 290 quencies from the *adapted radius* frequency distribution proposed in [25]. The adapted radius
 291 heuristic proposes to sample ω as

$$292 \quad (5.7) \quad \omega = R\varphi$$

293 where $R \in \mathbb{R}_+$ is the norm of ω and $\varphi \in \mathbb{R}^P$ is the random direction. The radius R is
 294 chosen with a radius distribution $R \sim p_R(R; \eta) = ((\eta R)^2 + \frac{1}{4}(\eta R)^4)^{\frac{1}{2}} e^{-\frac{1}{2}(\eta R)^2}$ where η is
 295 a scale parameter that should be adjusted to the current dataset to ensure that most of
 296 the spectral content of the GMM is sampled. By combining this radius distribution with the
 297 decomposition (5.7), we have a frequency distribution Δ_k referred as *adapted radius* frequency
 298 distribution. See Appendix C for details. With this distribution, we avoid sampling very low
 299 frequencies. Figure 3 illustrates the curve of $p(R)$ with different values of η .

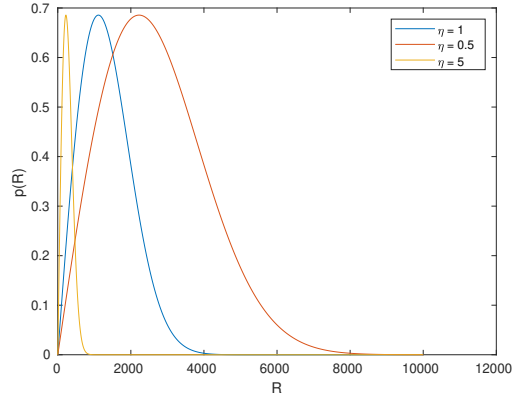


Figure 3. Curve of the radius distribution density

300 **5.2. Extension to low-rank covariances.** Bayesian MAP theory permits to use a GMM
 301 with degenerate covariance matrices as a denoising prior. In this case, the prior is given only
 302 in the union of subspaces spanned by the r leading eigenvectors of the K covariance matrices
 303 of the GMM. The experiments [38, 42] have shown that we can use low-rank covariance
 304 matrices for denoising while keeping good performance. This motivates us to approximate the
 305 covariance matrices in the GMM prior by low-rank matrices.

306 Following classical Burer-Monteiro method [4, 8] in low-rank matrix estimation, we pa-
 307 rameterize Σ_k by its factors X_k : $\Sigma_k = X_k X_k^T$. We define $f_{X,\alpha}$ the density function of a
 308 zero-mean GMM with $X = (X_k)_{k=1}^K$, where X_k is a factor of a covariance matrix.

309 Supposing that $\|\hat{y} - \mathcal{S}f_{X,\alpha}\|_2^2$ has a minimizer, we approximate the minimization (5.5) by

$$310 \quad (5.8) \quad (\hat{X}, \hat{\alpha}) \in \underset{\substack{X=(X_k)_{k=1}^K, X_k \in \mathbb{R}^{P \times r} \\ \alpha \in \mathbb{R}^K, \alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1}}{\arg \min} \|\hat{y} - \mathcal{S}f_{X,\alpha}\|_2^2,$$

311 i.e.

$$312 \quad (5.9) \quad (\hat{X}, \hat{\alpha}) \in \underset{\substack{X_k \in \mathbb{R}^{P \times r}, \forall k \\ \alpha \in \mathbb{R}^K, \alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1}}{\arg \min} \sum_{l=1}^m \left| \hat{y}_l - \sum_{k=1}^K \alpha_k e^{-\frac{1}{2} \omega_l^T X_k X_k^T \omega_l} \right|^2$$

313 where $\hat{X} = \{\hat{X}_1, \dots, \hat{X}_K\}$ is the collection of factorized rank reduced covariances. With the
 314 following proposition, we show that the difference between the costs minimized in (5.5) and
 315 (5.9) (the full rank and the low-rank cases, respectively) is associated with the smallest eigen-
 316 values of the covariance matrices. We qualitatively validate this approximation since these
 317 eigenvalues are typically small.

318 **Proposition 5.1.** Let $\Phi^* = \{\Sigma_1^*, \dots, \Sigma_K^*, \alpha_1^*, \dots, \alpha_K^*\}$ be a minimizer of (5.5). Suppose
 319 that there exists a minimizer $\hat{\Phi} = \{\hat{X}_1, \dots, \hat{X}_K, \hat{\alpha}_1, \dots, \hat{\alpha}_K\}$ for the problem (5.8). Let $C =$

320 $\frac{1}{2}\sqrt{\sum_{l=1}^m \|\omega_l\|_2^4}$. Then we have:

$$321 \quad \|\mathcal{S}f_{\hat{\Phi}} - \hat{y}\|_2 - \|\mathcal{S}f_{\Phi^*} - \hat{y}\|_2 \leq C \max_k (\sigma_{r+1}(\Sigma_k^*))$$

322 where the $\sigma_{r+1}(\Sigma_k^*)$ is the $(r+1)$ -th singular value of Σ_k^* sorted by decreasing order.

323 The proof is detailed in [Appendix D](#).

324 Ideally, we would like to obtain a similar bound for $\|\Sigma_k^* - \hat{X}_k \hat{X}_k^T\|_F$. We conjecture that
 325 a RIP (Restricted Isometry Property) would be needed for such a result. As the verification
 326 of RIP remains an open theoretical question in the zero-mean GMM case, we leave this
 327 theoretical question for further work.

328 **5.2.1. Related work.** There exist other methods to learn GMMs by incorporating a di-
 329 mensionality reduction technique. In the mixture models of probabilistic PCAs (MPPCA) [53],
 330 the covariance matrix Σ_k is parameterized as

$$331 \quad (5.10) \quad \Sigma_k = X_k X_k^T + \gamma_k^2 I.$$

332 The authors use the EM algorithm to optimize the parameters γ_k and X_k . In the high
 333 dimensional data clustering (HDDC) model [2], the authors generalized the MPPCA by setting
 334 the covariance matrix as

$$335 \quad (5.11) \quad \Sigma_k = X_k \text{diag}(\lambda_k) X_k^T + \gamma_k^2 I, \quad \lambda_k > 0.$$

336 As for the MPPCA, the parameters are learned by the EM algorithm. In the HDMI model [22],
 337 a model selection algorithm for the intrinsic dimension of each mixture component is proposed.
 338 In this model, the noise variance γ_k is a priori fixed and the other parameters are optimized
 339 by the EM algorithm. Finally, in the PCA-GMM model [21], the covariance matrix Σ_k is
 340 expressed as:

$$341 \quad (5.12) \quad \Sigma_k = \left(\frac{1}{\gamma_k^2} (I - X_k X_k^T) + X_k \tilde{S}^{-1} X_k^T \right)^{-1}, \quad \tilde{S} \in \text{SPD}(P).$$

342 This model is a more general one than HDDC. The parameter γ_k can either be fixed a priori or
 343 optimized simultaneously with the other parameters by the EM algorithm. Unlike the above
 344 models, our model doesn't estimate γ_k , we rather set a similar user-defined parameter (called
 345 μ in our case) at the denoising step. In our model, we also assume that the intrinsic dimension
 346 of each Gaussian component is the same and a priori fixed. Using automatically estimated
 347 ranks for covariances is a possible future work.

348 **5.3. An algorithm for learning patch prior from a sketch : LR-COMP (Low-Rank Con-**
 349 **tinuous Orthogonal Matching Pursuit).** Problem (4.7) can be solved approximately using the
 350 greedy Compressive Learning OMP called CL-OMP and a variation of CL-OMP called CL-
 351 OMP with Replacement (CL-OMPR) [25, 26]. These algorithms are based on the Matching
 352 Pursuit [34], Orthogonal Matching Pursuit [39] and Orthogonal Matching Pursuit with Re-
 353 placement [23] for classical compressive sensing, which handle sparse approximation problems.
 354 It starts from an empty support and it expands the support by greedily adding new atoms to

355 the current support. Each new atom θ' is found by maximizing the correlation $\langle \mathcal{S}f_{\theta'}, r \rangle$ where
 356 r is the current residual. Then it updates the weights and reduces the cost function with a
 357 descent algorithm initialized with the current parameters. For better feasible recovery, the
 358 algorithm using the replacement method was proposed. The approach increases the number
 359 of iterations of CL-OMP and extends the size of support more than the desired sparsity. Then
 360 it deletes the extra atoms by using the hard thresholding operator. We adapt these algorithms
 361 in the GMMs context with our low-rank approximation. Several modifications are detailed
 362 below:

- 363 • **No Replacement.** Although the algorithms using the replacement method show
 364 better results on synthetic data, our results tested on image patches show that the
 365 replacement step has a negligible effect. Therefore, we run our algorithm without this
 366 hard thresholding operator to decrease the computation time.
- 367 • **Estimation of the factors of covariance instead of the covariance matrices.**
 368 As we approximate the covariance matrices with their factors, in each step of the
 369 algorithm, we do operations directly on the factorized rank reduced covariance X
 370 instead of the covariance matrix Σ to lighten the computations.
- 371 • **Non-negativity and the normalization of the weights.** In Step 1 of the algo-
 372 rithm, we compute the real part of the correlation between the normalized atom and
 373 the residual as done in CL-OMP(R). This avoids a negative correlation and negative
 374 weights in practice. No matter how Step 3 was computed, using the projected gra-
 375 dient descent or the gradient descent, or with a direct calculation, there's negligible
 376 difference in the result and the running time. The weights are not forced to be sum-
 377 to-one at each iteration. However, after transforming the negative weights to zero, an
 378 l_1 -normalization of the weights is performed at the end of the algorithm.

Algorithm 5.1 LR-COMP: Compressive GMM estimation with low-rank covariances [50].

Input Empirical sketch \hat{y} , sketching operator \mathcal{S} , sparsity K , rank r

$\hat{r} \leftarrow y; X \leftarrow \emptyset$

for $t = 1$ **to** K **do**

Step 1: Perform a descent initialized with a $P \times r$ matrix of normally distributed random numbers:

$$X_k^* \leftarrow \arg \max_{X_k} \operatorname{Re} \left\langle \frac{\mathcal{S}f_{X_k}}{\|\mathcal{S}f_{X_k}\|_2}, \hat{r} \right\rangle_2, \text{init} = \text{rand}$$

Step 2: Extend the support: $X \leftarrow X \cup \{X_k^*\}$

Step 3: Find weights: $\alpha \leftarrow \arg \min_{\alpha} \left\| \hat{y} - \sum_{k=1}^{|X|} \alpha_k \mathcal{S}f_{X_k} \right\|_2^2$

Step 4: Perform a descent initialized with current parameters:

$$X, \alpha \leftarrow \arg \min_{X, \alpha} \left\| \hat{y} - \sum_{k=1}^{|X|} \alpha_k \mathcal{S}f_{X_k} \right\|_2^2, \text{init} = (X, \alpha)$$

Step 5: Update residual: $\hat{r} \leftarrow \hat{y} - \sum_{k=1}^{|X|} \alpha_k \mathcal{S}f_{X_k}$;

end for

Normalize the weights α_k such that $\sum_k \alpha_k = 1$

return Support X , weights α

379 The proposed algorithm is summarized in [Algorithm 5.1](#). In practice, we perform Step 4 with

380 a descent algorithm (L-BFGS). We use more iteration in the ultimate Step 4 (for $t = K$) as a
 381 "final adjustment". With this "final adjustment" step, we could reduce the running time by
 382 using fewer iterations in Step 4 for $t < K$. Our algorithm was implemented by extending the
 383 MATLAB toolbox [24]. The Matlab implementation of our approach is available at [50]. The
 384 main tool for the implementation of [Algorithm 5.1](#) is to compute the necessary gradients for
 385 the optimization problems in Steps 1, 3, and 4.

386 For the following section, denote the vector $v(X) = \mathcal{S}f_X \in \mathbb{R}^m$.

387 **5.3.1. Expression of the gradient for Step 1.** In step 1, we have the optimization problem
 388

$$389 \quad (5.13) \quad X_k^* \in \arg \max_{X_k \in \mathbb{R}^{P \times r}} \operatorname{Re} \left\langle \frac{\mathcal{S}f_{X_k}}{\|\mathcal{S}f_{X_k}\|_2}, \hat{r} \right\rangle_2, \quad \hat{r} \in \mathbb{C}^m.$$

390 Let $F(X_k) = -\operatorname{Re} \left\langle \frac{\mathcal{S}f_{X_k}}{\|\mathcal{S}f_{X_k}\|_2}, \hat{r} \right\rangle_2 = -\frac{v(X_k)^T \operatorname{Re}(\hat{r})}{\|v(X_k)\|_2}$, then problem (5.13) turns to

$$391 \quad (5.14) \quad X_k^* \in \arg \min_{X_k \in \mathbb{R}^{P \times r}} F(X_k).$$

392 In practice, with $W = [\omega_1, \dots, \omega_m] \in \mathbb{R}^{P \times m}$ the frequency matrix, we compute the gradient of
 393 $F(X_k)$ with the following operation:

$$394 \quad (5.15) \quad G = -\frac{1}{\|v(X_k)\|_2} W \left(W^T X_k \ast \left(v(X_k) \ast \left(\frac{F(X_k) v(X_k)}{\|v(X_k)\|_2} - \operatorname{Re}(\hat{r}) \right) \right) \right).$$

395 Here the symbol \ast represents the multiplication element by element in MATLAB. A matrix of
 396 size $m \times r$ multiplied using *dot** with a $m \times 1$ vector leads to a matrix of size $m \times r$ (multiplying
 397 all columns of the left side by the same column vector of the right side). The result G is a
 398 matrix of size $P \times r$. We need to reshape all the elements of the matrix G into a single column
 399 vector, whose result is the gradient $\nabla_{X_k} F(X_k)$. The detailed computation is in [Appendix E](#).

400 **5.3.2. The solution of Step 3.** The problem is

$$401 \quad (5.16) \quad \alpha^* = \arg \min_{\alpha \in \mathbb{R}^{|X|}} \left\| y - \sum_{k=1}^{|X|} \alpha_k \mathcal{S}f_{X_k} \right\|_2^2, \quad y \in \mathbb{C}^m.$$

402 Denote $V(X) = [v(X_1), \dots, v(X_{|X|})] \in \mathbb{R}^{m \times |X|}$, $\alpha = [\alpha_1, \dots, \alpha_{|X|}]^T \in \mathbb{R}^{|X|}$, then the problem can
 403 be expressed as a least-squares minimization

$$404 \quad (5.17) \quad \alpha^* = \arg \min_{\alpha \in \mathbb{R}^{|X|}} g(\alpha) = \arg \min_{\alpha \in \mathbb{R}^{|X|}} \|y - V\alpha\|_2^2.$$

405 We thus have

$$406 \quad (5.18) \quad \alpha^* = (V^T V)^{-1} V^T \hat{y}.$$

407 **5.3.3. Expression of the gradient for Step 4.** The problem is

$$408 \quad (5.19) \quad (X^*, \alpha) \in \arg \min_{\substack{X \in \mathbb{R}^{|X|}, X_k \in \mathbb{R}^{P+P \times r} \\ \alpha \in \mathbb{R}^{|X|}}} \left\| \hat{y} - \sum_{k=1}^{|X|} \alpha_k \mathcal{S}f_{X_k} \right\|_2^2.$$

409 Denote $V = [v(X_1), \dots, v(X_{|X|})]$, $\alpha = [\alpha_1, \dots, \alpha_K]^T$, we express

$$410 \quad (5.20) \quad h(X, \alpha) = \|\hat{y} - V\alpha\|_2^2,$$

411 so we have the gradients

$$412 \quad (5.21) \quad \nabla_{\alpha} h(X, \alpha) = 2V^T(V\alpha - \hat{y})$$

413 and

$$414 \quad (5.22) \quad \nabla_{X_k} h(X, \alpha) = 2\alpha_k \nabla_{X_k} v(X_k)^T (V\alpha - \hat{y}).$$

415 In practice, as in Step 1, we compute the second gradient by calculating the matrix

$$416 \quad (5.23) \quad G_2 = -2\alpha_k W(W^T X_k \dot{*} v(X_k) \dot{*} (V\alpha - y)).$$

417 The gradient $\nabla_{X_k} h(X, \alpha)$ corresponds the vector after reshaping G_2 .

418 As the function minimized here is smooth; the descent will be guaranteed to converge to
419 a local minimum. Recent works suggest that if all the OMP steps fall close enough to the
420 Gaussian of the global optimum [55], this step will converge to the global optimum under a
421 restricted isometry condition.

422 **5.4. Complexity of LR-OMP.** When estimating a K-components GMM, the proposed
423 algorithm LR-OMP has a computational cost of the order of $O(mP^2rK^2)$. In each iteration,
424 the computational cost is dominated by the matrix-vector product $W(W^T X)$ where W is a
425 matrix of size $P \times m$ and $W^T X$ is a matrix of size $m \times r$. As $m \ll n$, the computational cost of
426 our algorithm is lower than that of the EM. Moreover, it is possible to exploit the advantages
427 of GPU computing, the matrix multiplication can be performed by using multiple GPUs in
428 parallel [62]. This could result in a speed-up, especially for the "final adjustment" step.

429 **5.5. Denoising with low-rank covariance matrices.** In this section, we describe some
430 modifications required in EPLL to use our estimated model. The estimated parameters
431 are $\hat{\Phi} = \{\hat{X}_1, \dots, \hat{X}_K, \hat{\alpha}_1, \dots, \hat{\alpha}_K\}$ with $\hat{X}_k \in \mathbb{R}^{P \times r}$ and $\alpha_k \in \mathbb{R}_+$. A singular value decom-
432 position of \hat{X}_k is given by $\hat{X}_k = \hat{U}_k \hat{S}_k \hat{V}_k^T$. $\hat{U}_k, \hat{V}_k \in \mathbb{R}^{P \times P}$ are orthogonal matrices and
433 $\hat{S}_k = \text{diag}(\hat{s}_{k_1}, \dots, \hat{s}_{k_r}, 0, \dots, 0) \in \mathbb{R}^{P \times P}$ is a diagonal matrix. The r -rank covariance matrix
434 can be expressed with $\hat{\Sigma}_{kr} = \hat{X}_k \hat{X}_k^T = \hat{U}_k \hat{S}_k^2 \hat{U}_k^T$. We approximate the covariance matrix Σ_k
435 with $\Sigma_k \simeq \hat{\Sigma}_k = \hat{U}_k \hat{\Lambda}_k \hat{U}_k^T$ where $\hat{\Lambda}_k$ is formed as:

$$436 \quad (5.24) \quad \hat{\Lambda}_k = \begin{pmatrix} \hat{s}_{k_1}^2 & & & & \\ & \ddots & & & \\ & & \hat{s}_{k_r}^2 & & \\ & & & \mu & \\ & 0 & & & \ddots \\ & & & & & \mu \end{pmatrix}$$

437 where μ is a user parameter. Denoting $\hat{U}_k^r \in \mathbb{R}^{P \times r}$ the matrix formed by the first r columns
438 of \hat{U}_k and $\hat{\Lambda}_k^r$ the matrix formed with the first r rows and r columns of $\hat{\Lambda}_k$, we have:

$$439 \quad (5.25) \quad \left(\Sigma_k + \frac{1}{\beta} I_P \right)^{-1} = \hat{U}_k^r (\hat{\Lambda}_k^r + \frac{1}{\beta} I_r)^{-1} \hat{U}_k^{rT} + \frac{\beta}{\beta\mu + 1} (I_P - \hat{U}_k^r \hat{U}_k^{rT})$$

440 and

$$441 \quad (5.26) \quad \left(\Sigma_k + \frac{1}{\beta} I_P \right)^{-1} \Sigma_k = \hat{U}_k^r (\hat{\Lambda}_k^r + \frac{1}{\beta} I_r)^{-1} \hat{\Lambda}_k^r \hat{U}_k^{rT} + \frac{\beta\mu}{\beta\mu + 1} (I_P - \hat{U}_k^r \hat{U}_k^{rT}).$$

442 The detailed computation of (5.25) is in [Appendix F](#). Then the Gaussian selection step of
443 EPLL (2.11) becomes

$$444 \quad (5.27) \quad k_i^* = \arg \min_{1 \leq k \leq K} -2 \log \alpha_k + \sum_{j=1}^r \left(\log(\hat{s}_{k_j}^2 + \frac{1}{\beta}) + \frac{[\hat{v}_i^{(k)}]_j^2}{\hat{s}_{k_j}^2 + \frac{1}{\beta}} - \frac{\beta}{\beta\mu + 1} [\hat{v}_i^{(k)}]_j^2 \right)$$

445 where

$$446 \quad (5.28) \quad \hat{v}_i^{(k)} = \hat{U}_k^{rT} \tilde{z}_i.$$

447 With the optimal component k_i^* , the estimated patch (2.10) becomes (recall that \tilde{z}_i are cen-
448 tered patches and that m_i are the estimated mean of patches

$$\begin{aligned} \hat{z}_i &= (\Sigma_{k_i^*} + \frac{1}{\beta} I_P)^{-1} \Sigma_{k_i^*} \tilde{z}_i \\ &= \hat{U}_{k_i^*}^r (\hat{\Lambda}_{k_i^*}^r + \frac{1}{\beta} I_r)^{-1} \hat{\Lambda}_{k_i^*}^r \hat{U}_{k_i^*}^{rT} \tilde{z}_i + \frac{\beta\mu}{\beta\mu + 1} (I_P - \hat{U}_{k_i^*}^r \hat{U}_{k_i^*}^{rT}) \tilde{z}_i \\ &= \hat{U}_{k_i^*}^r \hat{\Lambda}'_{k_i^*} \hat{v}_i^{(k_i^*)} + \frac{\beta\mu}{\beta\mu + 1} (\tilde{z}_i - \hat{U}_{k_i^*}^r \hat{v}_i^{(k_i^*)}) + m_i \end{aligned}$$

450 with

$$451 \quad (5.30) \quad \hat{\Lambda}'_{k_i^*} = (\hat{\Lambda}_{k_i^*}^r + \frac{1}{\beta} I_r)^{-1} \hat{\Lambda}_{k_i^*}^r = \text{diag} \left(\frac{\hat{s}_{k_{ij}^*}^2}{\hat{s}_{k_{ij}^*}^2 + \frac{1}{\beta}} \right)_{j=1, \dots, r}.$$

452 **6. Experimental Results.** In this section we present several numerical experiments to
453 illustrate the benefits of our approach.

454 We randomly extract $n = 4 \times 10^6$ patches of size $P = 7 \times 7$ from the training images of the
455 Berkeley Segmentation Database (BSDS) [35]. Then the patches are compressed into a sketch.
456 Based on observations from numerical simulations, the scale parameter η must be adjusted
457 for each task and dataset [47]. In [25], the authors propose to estimate this parameter with a
458 small sketch on a small subset from the dataset. In our work, we choose the optimal parameter
459 η by hand. We then learn a mixture model of $K = 20$ Gaussian components with low-rank
460 covariance matrices. We compare the denoised results with the results obtained with a GMM
461 (full-rank) prior model learned by the EM algorithm. For the comparison, we train the prior

462 from the same image patches dataset. The denoising is performed with EPLL¹. To evaluate
 463 the quality of denoised images, we use two measures: PSNR (Peak Signal to Noise Ratio) and
 464 SSIM (Structural Similarity) [57]. For the test images, we use two datasets: Set12 [61] and
 465 BSD68 [43] for a thorough evaluation. The code is available at [50] to reproduce the results
 466 below.

467 **Figure 4** shows the denoising performance on 6 images of the Set12 dataset. The noisy
 468 images are obtained by adding zero-mean Gaussian noise with standard deviations $\sigma = 20$ to
 469 the test images. The covariance matrices of the model learned by the sketching have the rank
 470 $r = 20$. We observe that for most of images, we obtain similar or better values of PSNR and
 471 SSIM.

472 Another evaluation was carried out on the images from the BSD68 dataset. The test
 473 images have been corrupted by adding white Gaussian noise with standard deviations $\sigma =$
 474 $15, 60$. **Table 1** shows the average PSNR and SSIM values on the dataset. On average,
 475 our approach results are 0.2dB below the results with EM in terms of PSNR. However, our
 476 approach is about 2 times faster than the EM. Moreover, a loss of 0.2dB does not affect the
 477 visual quality in most natural images.

Table 1

The average PSNR and SSIM on the BSD68 dataset with 2 different levels of noise.

σ	Sketching	EM
15	31.8 / .876	32.0 / .879
50	24.4 / .637	24.6 / .646

478 We also evaluate the similarity of the models learned via EM and LR-OMP. In **Figure 5** we
 479 visualize the leading eigenvectors of the learned covariance whose weight is the largest. The
 480 represented eigenvectors are ordered decreasingly with respect to the eigenvalues. The figure
 481 shows that the learned components have rich and similar structures except for the smallest
 482 eigenvalues where we observe differences. As we also observe that the eigenvalues decay much
 483 faster with our method than with EM, it is hard to interpret further the difference with the
 484 result of EM. We can still say that these different "dictionaries" lead to similar denoising
 485 results.

486 **6.1. Influence of realization of sketching operator.** Our approach performs stable per-
 487 formances with different initialization. **Table 2** shows the variability of the PSNR/SSIM over
 488 different random sketch realizations. The evaluation is carried out on the classical images:
 489 cameraman, house, etc. The noisy images are obtained by adding white Gaussian noise with
 490 standard deviations $\sigma = 20$ to the test images.

491 **6.2. Influence of sketch size and the compression rate.** Theoretically, we can success-
 492 fully estimate a GMM with sufficiently separated mean and random Fourier sketching with
 493 high probability as long as the sketch size $m \geq O(K^2 P \text{polylog}(K, P))$. In our case, we learn
 494 zero-mean Gaussians. From [25], empirical results indicate that a sketch size of the order of
 495 the number of parameters is sufficient (i.e. it is conjectured that $K^2 P$ could be reduced to

¹Matlab implementation based on the code of [38].



Figure 4. From left to right: Original images, noisy images with noise $\sigma = 20$, results with EM model, results with LR-COMP model. The denoising results are evaluated with PSNR/SSIM. Similar denoising performances are obtained with LR-COMP with a 1000 times smaller compressed database. To estimate the prior model, our method is 2 times faster than the EM algorithm.

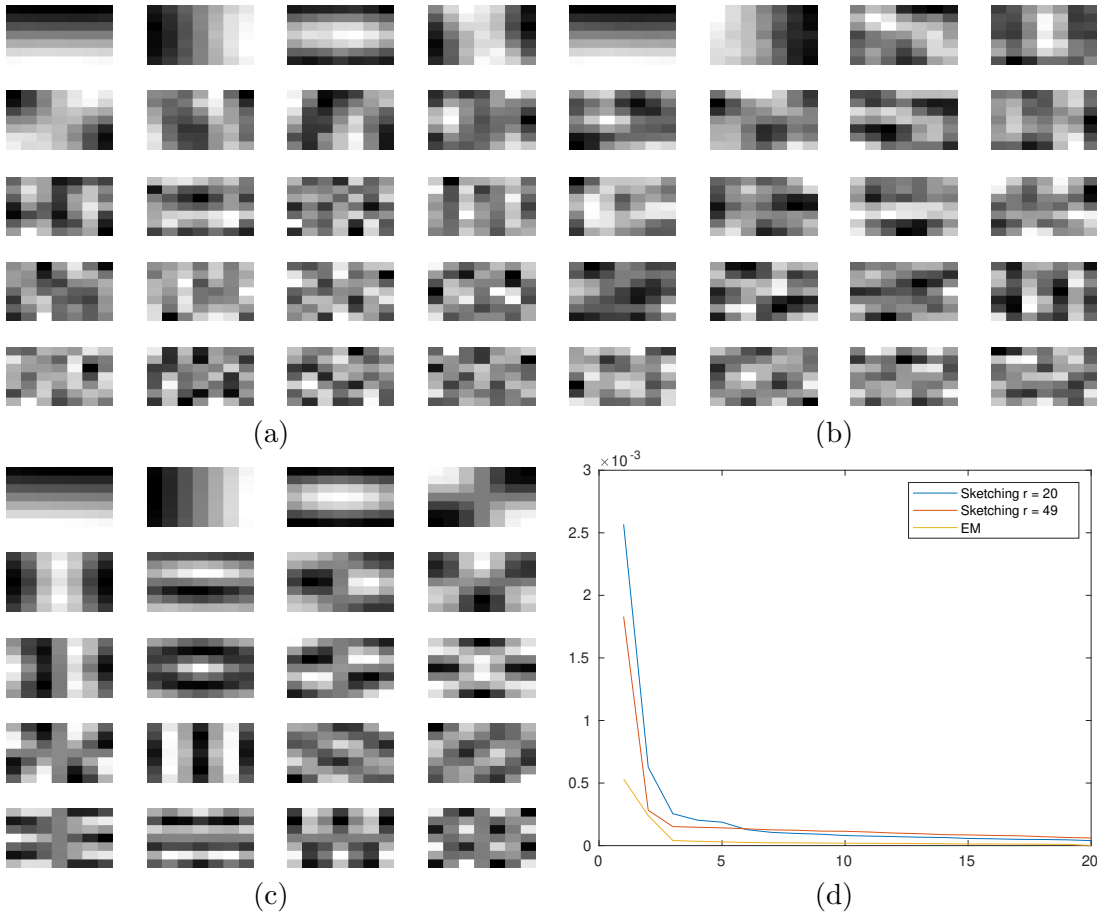


Figure 5. The first 20 eigenvectors of the covariance matrices (for the heaviest weight) learned by LR-OMP with rank $r = 20$ (a), $r = 49$ (b) and EM (c). The decay of the corresponding eigenvalues (d).

Table 2

Image denoising performance comparison of models estimated over different random sketch realization.

	Realization 1	Realization 2	Realization 3	Realization 4
<i>cameraman</i>	33.1 / .930	33.3 / .930	33.4 / .930	33.4 / .930
<i>house</i>	35.4 / .926	35.4 / .925	35.4 / .926	35.4 / .925
<i>jetplane</i>	32.1 / .936	32.3 / .937	32.4 / .937	32.4 / .937
<i>lena</i>	32.0 / .931	32.2 / .931	32.2 / .931	32.2 / .931
<i>pirate</i>	29.8 / .907	30.0 / .908	30.0 / .908	30.0 / .908

496 KP). In our experiments, we set $m = cK(P \times r + 1) = 10K(P \times r + 1) \approx 2 \times 10^5$, i.e the com-
 497 pressed database is approximately 1000 times smaller than the original patch database. The
 498 gains in terms of memory is approximately $\frac{n}{m}$ times compared to the EM approach. **Figure 6**
 499 shows the denoising performance and estimation time with models learned by using different
 500 sketch sizes ($c = 1, 5, 10, 20$). Our experiments show that a larger sketch size doesn't improve
 501 the denoising performance necessarily, and indeed it causes more learning time.

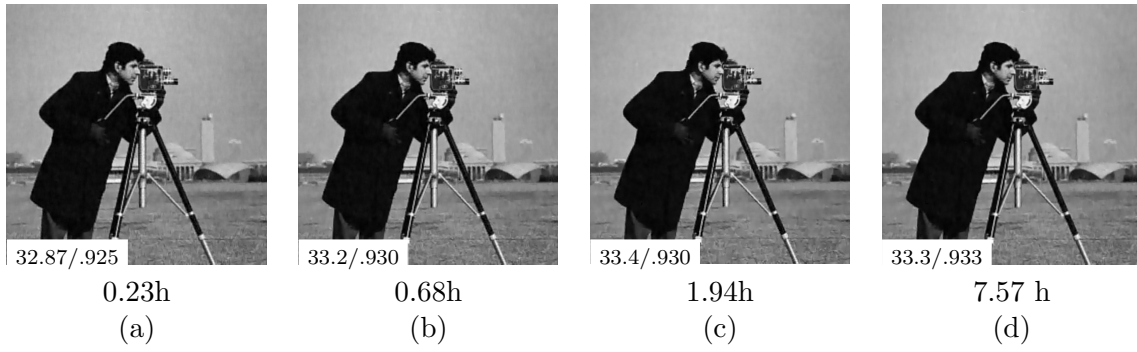


Figure 6. Denoising performance (PSNR/SSIM) and estimation time (hours) of models learned with different sketch size. $c=1$ (a), 5 (b), 10 (c), 20 (d).

502 **6.3. Influence of the rank r .** Figure 7 shows the denoising performance of models esti-
 503 mated with different ranks. Our experiments show that the model with reduced rank results
 504 in a minor PSNR/SSIM drop compared to the full-rank model. However, the learning time is
 505 much faster. According to the experiments, we cannot reduce the rank further (less than 20)
 to keep good denoising performance.



Figure 7. Denoising performance (PSNR/SSIM) of models learned with different intrinsic dimensions. $r = 10$ (left), 20 (middle), 49(right).

506

507 **6.4. Learning time.** In terms of time complexity, the running time depends on the number
 508 of components K and the complexity of the descent algorithm. In our approach, we use the
 509 Limited-memory BFGS algorithm to handle the optimization problems in Step 1 and 4. The
 510 latter is the most time-consuming part of the algorithm. To get the model ($c = 10, r = 20$)
 511 that achieves the denoising performance of our experiments (Figure 4), it takes less than
 512 2 hours on a computer with 2 * 32 cores AMD EPYC 7452 @ 2,35 GHz. With the same
 513 environment, our learning algorithm is about 2 times faster than the EM algorithm (3.68h)².

²Mo Chen (2021). EM Algorithm for Gaussian Mixture Model (EM GMM) (<https://www.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model-em-gmm>), MATLAB Central File Exchange. Retrieved October 11, 2021.

514 **7. Conclusions.** In this work, we adapt the sketching framework in the context of image
 515 patches. We propose an algorithm LR-COMP to estimate a GMM with low-rank approxi-
 516 mation and provide an implementation of the algorithm. Experiments illustrate that a high-
 517 dimensional GMM can be learned from a compressed database and then used for patch-based
 518 denoising. We achieve denoising performances close to state-of-the art model based methods
 519 while the learning procedure uses less memory and time than the classical EM algorithm.

520 In future works, we can generalize our approach to other models such as GGMM (General-
 521 ized Gaussian Mixture Model) for a better denoising performance [11]. We also aim to adapt
 522 the sketching to more inverse problems such as image super-resolution, image deblurring, etc.
 523 Another perspective is to extend our model to the study of video denoising method as the
 524 potential of the technique for video restoration remains unexplored. As mentioned earlier,
 525 the scale parameter η must be adjusted for each task and dataset. In our work, we choose
 526 this parameter by hand. Moreover, in our model, the intrinsic dimension of each Gaussian
 527 component is assumed the same and a priori fixed. It could be useful to design a procedure to
 528 estimate the hyper-parameters automatically in future work. In our work, we estimate a GMM
 529 with zero-mean. In this context, the notion of separation used to prove the restricted isometry
 530 property which in turn proves identifiability of the GMM in [18] and convergence of gradient
 531 descent in [55]. Proving a RIP on zero-mean would require a new notion of separation. We
 532 conjecture that an angular separation between Gaussian might enable us to prove such RIP.
 533 Such separation could e.g. compare the angle between eigenvectors of covariances by decreas-
 534 ing eigenvalue amplitude and weight the separation accordingly. The low-rank model should
 535 add even more separation as the Gaussian are supported on different sub-spaces. We still
 536 show empirically that the sketching process is successful without this separation assumption.
 537 This opens interesting new theoretical questions for the study of the success of compressive
 538 learning in patch-based image processing.

539 Appendix A. Definitions and theorems.

540 **Definition A.1. Singular values** For $A \in \mathbb{C}^{m \times n}$ and $i = 1, \dots, \min(m, n)$, the singular
 541 values $\sigma_i(A)$ (that we suppose sorted by decreasing order) of the matrix A are the absolute
 542 values of the eigenvalues of the matrix AA^T :

$$543 \quad (\text{A.1}) \quad \sigma_i^2(A) = \lambda_i(AA^T).$$

544 **Definition A.2. Frobenius norm.** For a matrix $A \in \mathbb{C}^{m \times n}$, the Frobenius norm of A is
 545 defined as

$$546 \quad (\text{A.2}) \quad \|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2} = \sqrt{\text{trace}(A^T A)} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2(A)}$$

547 where $\sigma_i(A)$ are the singular values of A .

548 **Definition A.3. Operator norm.** For a continuous linear operator $A : V \rightarrow W$, the
 549 operator norm of A is defined as

$$550 \quad (\text{A.3}) \quad \begin{aligned} \|A\|_{\text{op}} &= \inf\{c \geq 0 : \|Av\| \leq c\|v\| \quad \forall v \in V\} \\ &= \sup\left\{\frac{\|Av\|}{\|v\|} : v \neq 0 \quad \text{and} \quad v \in V\right\}. \end{aligned}$$

551 **Theorem A.4. Eckart-Young-Mirsky theorem.** Let $D = U\Sigma V^\top \in \mathbb{R}^{m \times n}$, $m \geq n$ be
 552 the singular value decomposition of D with $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_m)$. Let U_r (resp. V_r) be the
 553 matrix formed by the first r columns of U (resp. V) and $\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r)$. Then the
 554 r -rank matrix, obtained from the truncated singular value decomposition: $D^* = U_r \Sigma_r V_r^\top$ is
 555 the minimizer of the low-rank approximation:

$$556 \quad (\text{A.4}) \quad \|D - D^*\|_F = \min_{\text{rank}(D') \leq r} \|D - D'\|_F = \sqrt{\sum_{j \geq r+1} \sigma_j^2(D)}.$$

557 The minimizer D^* is unique if and only if $\sigma_{r+1} < \sigma_r$.

558 **Appendix B. EM algorithm.** Given a data set of n clean training patches $\chi =$
 559 $\{x_1, \dots, x_n\} \subset \mathbb{R}^{P \times n}$, the EM algorithm for estimating a GMM can be summarized as fol-
 560 lows:

561 1. Define the number of components K . For each component k , we initialize the param-
 562 eters $\Theta_k = (\mu_k, \Sigma_k, \alpha_k)$ randomly, and we compute the log likelihood

$$563 \quad (\text{B.1}) \quad \log \mathcal{L}(\Theta_k; x_1, \dots, x_n) = \sum_{i=1}^n \log \left(\sum_{k=1}^K \alpha_k \mathcal{N}_P(x_i; \mu_k, \Sigma_k) \right)$$

564 2. E-Step

565 Compute the posterior function $\Gamma_{i,k}$ with the current parameters Θ_k :

$$566 \quad (\text{B.2}) \quad \Gamma_{i,k} = \frac{\alpha_k \mathcal{N}_P(x_i; \mu_k, \Sigma_k)}{\sum_{j=1}^K \alpha_j \mathcal{N}_P(x_i; \mu_j, \Sigma_j)}$$

567 3. M-Step

568 Re-estimate the parameters Θ_k^{new} with the $\Gamma_{i,k}$ obtained in the E-Step:

$$569 \quad (\text{B.3}) \quad \mu_k^{new} = \frac{1}{N_k} \sum_{i=1}^n \Gamma_{i,k} x_i$$

570

$$571 \quad (\text{B.4}) \quad \Sigma_k^{new} = \frac{1}{N_k} \sum_{i=1}^n \Gamma_{i,k} (x_i - \mu_k)^T (x_i - \mu_k)$$

572

$$573 \quad (\text{B.5}) \quad \alpha_k^{new} = \frac{N_k}{\sum_{k=1}^K N_k}$$

574 where $N_k = \sum_{i=1}^n \Gamma_{i,k}$.

575 4. Re-evaluate the log likelihood. Iterate E-Step and M-Step until the log likelihood or
 576 the parameters are not changing much.

577 **Appendix C. Design of the adapted radius distribution.** The choice of the probability
 578 distribution Δ to draw the frequencies is a key ingredient in designing the sketching opera-
 579 tor. In our work, we choose the frequency distribution called the adapted radius [27] with a
 580 heuristic. In this appendix, we describe the design of the adapted radius distribution.

581 Assuming that we want to estimate a P -dimensional Gaussian $g = \mathcal{N}(0, I_P)$, we can
 582 compute the characteristic function $\psi_g(\omega)$ associated with g :

$$583 \quad (\text{C.1}) \quad \psi_g(\omega) = e^{-\frac{1}{2}\omega^T\omega}.$$

584 The adapted radius heuristic proposes not to sample ω directly but rather to sample the radius
 585 of the P -dimensional Gaussian $R = \sqrt{\omega^T\omega}$. Thus, we draw the frequency $\omega \in \mathbb{R}^P$ as

$$586 \quad (\text{C.2}) \quad \omega = R\varphi$$

587 where the radius $R \in \mathbb{R}_+$ is chosen with a radius distribution $R \sim p_R(R; \eta)$, and the direction
 588 $\varphi \in \mathbb{R}^P$ is uniformly generated on the l_2 unit sphere S_{P-1} , i.e. $\varphi \sim \mathcal{U}(S_{P-1})$. Then, the
 589 characteristic function $\psi_g(\omega)$ reduces to

$$590 \quad (\text{C.3}) \quad \psi_g(\omega) = \psi_g(R\varphi) = e^{-\frac{1}{2}R^2} = \psi(R).$$

591 We obtain a one-dimensional Gaussian function for R . To design the radius distribution, we
 592 consider the estimation of a Gaussian $g = \mathcal{N}(0, 1)$. We aim at sampling the radius R leading
 593 to large variations of the characteristic function when the parameters are closed to the true
 594 parameters. In other words, when parameters (μ, σ^2) are closed to $(0, 1)$, we want have a large
 595 $|\psi_{(\mu, \sigma^2)}(R) - \psi_{(0, 1)}(R)|$. This can be accomplished by promoting the radius R which makes
 596 the norm of the gradient $\|\nabla\psi_{(\mu, \sigma^2)}(R)\|_2$ large. Recall that $\psi_{(\mu, \sigma^2)}(R) = e^{-i\mu R}e^{-\frac{1}{2}\sigma^2 R^2}$ and
 597 the norm of the gradient is:

$$598 \quad (\text{C.4}) \quad \|\nabla\psi_{(\mu, \sigma^2)}(R)\|_2^2 = |-iR\psi_{(\mu, \sigma^2)}(R)|^2 + \left|-\frac{1}{2}R^2\psi_{(\mu, \sigma^2)}(R)\right|^2 = (R^2 + \frac{1}{4}R^4)e^{-\sigma^2 R^2}.$$

599 Therefore, $\|\nabla\psi_{(0, 1)}(R)\|_2 = (R^2 + \frac{1}{4}R^4)^{\frac{1}{2}}e^{-\frac{1}{2}R^2}$. It yields the density of a radius distribution :

$$600 \quad (\text{C.5}) \quad p_R(R; \eta) = ((\eta R)^2 + \frac{1}{4}(\eta R)^4)^{\frac{1}{2}}e^{-\frac{1}{2}(\eta R)^2}.$$

601 Here the scale parameter η should be chosen to probe the spectral content of the true GMM
 602 model.

603 **Appendix D. Proof of Proposition 5.1.**

604 *Proof.* Let $\Phi_k^* = (\Sigma_k^*, \alpha_k^*)$ be the minimizer of the problem (5.5), i.e.

$$605 \quad (\text{D.1}) \quad \Phi_k^* \in \arg \min_{\substack{\Sigma_k \in \mathbb{R}^{P \times P} \\ \alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1}} \left| \sum_{l=1}^m \left| \sum_{k=1}^K \alpha_k e^{-\frac{1}{2}\omega_l^T \Sigma_k \omega_l} - \hat{y}_l \right| \right|^2$$

606 and suppose that there exists a minimizer $\hat{\Phi}_k = (\hat{X}_k, \hat{\alpha}_k)$ for the problem (5.8):

$$607 \quad (\text{D.2}) \quad \hat{\Phi}_k \in \arg \min_{\substack{X_k \in \mathbb{R}^{P \times r} \\ \alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1}} \sum_{l=1}^m \left| \sum_{k=1}^K \alpha_k e^{-\frac{1}{2} \omega_l^T X_k X_k^T \omega_l} - \hat{y}_l \right|^2.$$

608 Let $\tilde{\Sigma}_k$ be the best rank- r approximation of Σ_k^* with the rank r i.e.

$$609 \quad (\text{D.3}) \quad \tilde{\Sigma}_k \in \arg \min_{\Sigma, \text{rank}(\Sigma)=r} \|\Sigma_k^* - \Sigma\|_F^2.$$

610 Define $\tilde{\Phi} = (\tilde{\Sigma}_k, \alpha_k^*)$. According to the definition (D.2) and the triangle inequality, we have

$$611 \quad (\text{D.4}) \quad \begin{aligned} \|\mathcal{S}f_{\hat{\Phi}} - y\|_2 &\leq \|\mathcal{S}f_{\tilde{\Phi}} - \hat{y}\|_2 \\ &= \|\mathcal{S}f_{\tilde{\Phi}} - \mathcal{S}f_{\Phi^*} + \mathcal{S}f_{\Phi^*} - \hat{y}\|_2 \\ &\leq \|\mathcal{S}f_{\tilde{\Phi}} - \mathcal{S}f_{\Phi^*}\|_2 + \|\mathcal{S}f_{\Phi^*} - y\|_2 \end{aligned}$$

612 The first term is

$$613 \quad (\text{D.5}) \quad \begin{aligned} \|\mathcal{S}f_{\tilde{\Phi}} - \mathcal{S}f_{\Phi^*}\|_2^2 &= \left\| \sum_{k=1}^K \alpha_k^* \mathcal{S}(f_{\tilde{\Sigma}_k} - f_{\Sigma_k^*}) \right\|_2^2 = \sum_{l=1}^m \left| \sum_{k=1}^K \alpha_k^* \left(e^{-\frac{1}{2} \omega_l^T \tilde{\Sigma}_k \omega_l} - e^{-\frac{1}{2} \omega_l^T \Sigma_k^* \omega_l} \right) \right|^2 \\ &= \sum_{l=1}^m \left| \sum_{k=1}^K \alpha_k^* e^{-\frac{1}{2} \omega_l^T \tilde{\Sigma}_k \omega_l} \left(1 - e^{-\frac{1}{2} \omega_l^T (\Sigma_k^* - \tilde{\Sigma}_k) \omega_l} \right) \right|^2. \end{aligned}$$

614 Using the convexity inequality $|1 - e^{-x}| \leq |x|$ and Cauchy–Schwarz inequality, we have

$$615 \quad (\text{D.6}) \quad \begin{aligned} \left| e^{-\frac{1}{2} \omega_l^T \tilde{\Sigma}_k \omega_l} \left(1 - e^{-\frac{1}{2} \omega_l^T (\Sigma_k^* - \tilde{\Sigma}_k) \omega_l} \right) \right| &\leq \left| 1 - e^{-\frac{1}{2} \omega_l^T (\Sigma_k^* - \tilde{\Sigma}_k) \omega_l} \right| \\ &\leq \frac{1}{2} \left| \omega_l^T (\Sigma_k^* - \tilde{\Sigma}_k) \omega_l \right| = \frac{1}{2} \left| \langle \omega_l, (\Sigma_k^* - \tilde{\Sigma}_k) \omega_l \rangle \right|. \end{aligned}$$

616 By the Eckart-Young-Mirsky theorem, we have that the largest singular value of $\Sigma_k^* - \tilde{\Sigma}_k$ is
617 $\sigma_{r+1}(\Sigma_k^*)$ and

$$618 \quad (\text{D.7}) \quad \left| \langle \omega_l, (\Sigma_k^* - \tilde{\Sigma}_k) \omega_l \rangle \right| \leq \|\omega_l\|_2^2 \|(\Sigma_k^* - \tilde{\Sigma}_k)\|_{\text{op}} = \|\omega_l\|_2^2 \sigma_{r+1}(\Sigma_k^*).$$

619 Therefore, using $\sum_{k=1}^K \alpha_k = \|\alpha\|_1 = 1$ and Hölder's inequality, and upper bound on (D.5)
620 reads

$$621 \quad (\text{D.8}) \quad \begin{aligned} \|\mathcal{S}f_{\tilde{\Phi}} - \mathcal{S}f_{\Phi^*}\|_2^2 &\leq \frac{1}{4} \sum_{l=1}^m \left| \|\alpha\|_1 \max_k (\|\omega_l\|_2^2 \sigma_{r+1}(\Sigma_k^*)) \right|^2 \\ &= \left(\frac{1}{4} \sum_{l=1}^m \|\omega_l\|_2^4 \right) \max_k (\sigma_{r+1}(\Sigma_k^*))^2. \end{aligned}$$

622

623 Denoting $C = \frac{1}{2} \sqrt{\sum_{l=1}^m \|\omega_l\|_2^4}$, we have from (D.4) that:

624 (E.9)
$$\|\mathcal{S}f_{\hat{\Phi}} - \hat{y}\|_2 \leq C \max_k(\sigma_{r+1}(\Sigma_k^*)) + \|\mathcal{S}f_{\Phi^*} - \hat{y}\|_2.$$

625

626 **Appendix E. Calculation of the gradient.** Denote $F(X_k) = -\frac{v(X_k)^T \hat{r}}{\|v(X_k)\|_2}$, where $r \in \mathbb{R}^m$ is
627 the real part of \hat{r} . We compute the gradient of F as follows:

628 (E.1)
$$\begin{aligned} \nabla_{X_k} F(X_k) &= -\frac{1}{\|v(X_k)\|_2^2} \left((\nabla_{X_k} v(X_k))^T r \|v(X_k)\|_2 - \frac{v(X_k)^T r (\nabla_{X_k} v(X_k))^T v(X_k)}{\|v(X_k)\|_2} \right) \\ &= -\frac{(\nabla_{X_k} v(X_k))^T}{\|v(X_k)\|_2} \left(r + \frac{v(X_k)^T r v(X_k)}{\|v(X_k)\|_2^2} \right) \\ &= \frac{(\nabla_{X_k} v(X_k))^T}{\|v(X_k)\|_2} \left(\frac{F(X_k) v(X_k)}{\|v(X_k)\|_2} - r \right). \end{aligned}$$

629 For each component $v_l(X_k) = e^{-\frac{1}{2} \omega_l^T X_k X_k^T \omega_l}$, we have

630 (E.2)
$$\frac{\partial v_l(X_k)}{\partial X_k} = -v_l(X_k) X_k^T \omega_l \omega_l^T.$$

631 Then for a given vector $z \in \mathbb{R}^m$

632 (E.3)
$$\langle \nabla_{X_k} v(X_k), z \rangle = -\sum_{l=1}^m z_l v_l(X_k) X_k^T \omega_l \omega_l^T.$$

633 In practice, we compute the scalar product with

634 (E.4)
$$\langle \nabla_{X_k} v(X_k), z \rangle = -W(W^T X_k \star (v(X_k) \star z))$$

635 where $W = [\omega_1, \dots, \omega_m] \in M_{P,m}(\mathbb{R})$ the frequency matrix and \star the multiplication element by
636 element in MATLAB. We compute the matrix G as

637 (E.5)
$$G = -\frac{1}{\|v(X_k)\|_2} W \left(W^T X_k \star \left(v(X_k) \star \left(\frac{F(X_k) v(X_k)}{\|v(X_k)\|_2} - r \right) \right) \right).$$

638 As a consequence, we reshape the matrix G to a vector which corresponds the gradient
639 $\nabla_{X_k} F(X_k)$.640 **Appendix F. The expression of (5.25).** Denoting $\hat{U}_k^r \in \mathbb{R}^{P \times r}$ (resp. $\hat{U}_k^c \in \mathbb{R}^{P \times (P-r)}$) the
641 matrix formed by the first r (resp. the last $P-r$) columns of \hat{U}_k and $\hat{\Lambda}_k^r$ the matrix formed
642 with the first r rows and r columns of $\hat{\Lambda}_k$. We use the bloc matrix multiplication:

643 (F.1)
$$\begin{aligned} \left(\Sigma_k + \frac{1}{\beta} I_P \right)^{-1} &= \hat{U}_k \left(\hat{\Lambda}_k + \frac{1}{\beta} I \right)^{-1} \hat{U}_k^T \\ &= \hat{U}_k^r \left(\hat{\Lambda}_k^r + \frac{1}{\beta} I_r \right)^{-1} \hat{U}_k^{rT} + \left(\mu + \frac{1}{\beta} I_{P-r} \right)^{-1} \hat{U}_k^c \hat{U}_k^{cT}. \end{aligned}$$

644 We have $\hat{U}_k^c \hat{U}_k^{cT} = (I_p - \hat{U}_k^r \hat{U}_k^{rT})$, thus

$$645 \quad (\text{F.2}) \quad \left(\Sigma_k + \frac{1}{\beta} I_P \right)^{-1} = \hat{U}_k^r (\hat{\Lambda}_k^r + \frac{1}{\beta} I_r)^{-1} \hat{U}_k^{rT} + \frac{\beta}{\beta\mu + 1} (I_p - \hat{U}_k^r \hat{U}_k^{rT}).$$

646 **Acknowledgments.** The authors acknowledge the support of the French National Re-
 647 search Agency (ANR) under reference ANR-20-CE40-0001 (EFFIREG project). Experiments
 648 presented in this paper were carried out using the PlaFRIM experimental testbed, supported
 649 by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil
 650 Régional d'Aquitaine (see <https://www.plafrim.fr/>). The authors are indebted to anonymous
 651 reviewers for providing insightful comments which have resulted in this paper.

652

REFERENCES

- 653 [1] C. AGUERREBERE, A. ALMANSA, Y. GOUSSEAU, J. DELON, AND P. MUSÉ, *Single shot high dynamic range*
 654 *imaging using piecewise linear estimators*, in 2014 IEEE International Conference on Computational
 655 Photography (ICCP), IEEE, 2014, pp. 1–10.
- 656 [2] C. BOUYEYRON, S. GIRARD, AND C. SCHMID, *High-dimensional data clustering*, *Computational statistics*
 657 *& data analysis*, 52 (2007), pp. 502–519.
- 658 [3] A. BUADES, B. COLL, AND J.-M. MOREL, *A non-local algorithm for image denoising*, in 2005 IEEE
 659 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, IEEE,
 660 2005, pp. 60–65.
- 661 [4] S. BURER AND R. D. MONTEIRO, *Local minima and convergence in low-rank semidefinite programming*,
 662 *Mathematical Programming*, 103 (2005), pp. 427–444.
- 663 [5] E. BYRNE, A. CHATALIC, R. GRIBONVAL, AND P. SCHNITER, *Sketched clustering via hybrid approximate*
 664 *message passing*, *IEEE Transactions on Signal Processing*, 67 (2019), pp. 4556–4569.
- 665 [6] N. CAI, Y. ZHOU, S. WANG, B. W.-K. LING, AND S. WENG, *Image denoising via patch-based adaptive*
 666 *gaussian mixture prior method*, *Signal, Image and Video Processing*, 10 (2016), pp. 993–999.
- 667 [7] A. CHATALIC, R. GRIBONVAL, AND N. KERIVEN, *Large-scale high-dimensional clustering with fast sketch-*
 668 *ing*, in 2018 International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE,
 669 2018, pp. 4714–4718.
- 670 [8] Y. CHI, Y. M. LU, AND Y. CHEN, *Nonconvex optimization meets low-rank matrix factorization: An*
 671 *overview*, *IEEE Transactions on Signal Processing*, 67 (2019), pp. 5239–5269.
- 672 [9] G. CORMODE AND S. MUTHUKRISHNAN, *An improved data stream summary: the count-min sketch and*
 673 *its applications*, *Journal of Algorithms*, 55 (2005), pp. 58–75.
- 674 [10] K. DABOV, A. FOI, V. KATKOVNIK, AND K. EGIAZARIAN, *Image denoising by sparse 3-d transform-*
 675 *domain collaborative filtering*, *IEEE Transactions on image processing*, 16 (2007), pp. 2080–2095.
- 676 [11] C.-A. DELEDALLE, S. PARAMESWARAN, AND T. Q. NGUYEN, *Image denoising with generalized gaussian*
 677 *mixture model patch priors*, *SIAM Journal on Imaging Sciences*, 11 (2018), pp. 2568–2609.
- 678 [12] J. FENG, L. SONG, X. HUO, X. YANG, AND W. ZHANG, *Image restoration via efficient gaussian mixture*
 679 *model learning*, in 2013 IEEE International Conference on Image Processing, IEEE, 2013, pp. 1056–
 680 1060.
- 681 [13] M. FONTAINE, C. VANWYNSBERGHE, A. LIUTKUS, AND R. BADEAU, *Scalable source localization with*
 682 *multichannel α -stable distributions*, in 2017 25th European Signal Processing Conference (EUSIPCO),
 683 IEEE, 2017, pp. 11–15.
- 684 [14] M. FONTAINE, C. VANWYNSBERGHE, A. LIUTKUS, AND R. BADEAU, *Sketching for nearfield acoustic*
 685 *imaging of heavy-tailed sources*, in International Conference on Latent Variable Analysis and Signal
 686 Separation, Springer, 2017, pp. 80–88.
- 687 [15] S. FOUCAIT AND H. RAUHUT, *A mathematical introduction to compressive sensing*, Springer, 2013.
- 688 [16] D. GEMAN AND C. YANG, *Nonlinear image recovery with half-quadratic regularization*, *IEEE transactions*
 689 *on Image Processing*, 4 (1995), pp. 932–946.

- 690 [17] R. GRIBONVAL, G. BLANCHARD, N. KERIVEN, AND Y. TRAONMILIN, *Compressive statistical learning*
691 *with random feature moments*, Mathematical Statistics and Learning, 3 (2021), pp. 113–164.
- 692 [18] R. GRIBONVAL, G. BLANCHARD, N. KERIVEN, AND Y. TRAONMILIN, *Statistical learning guarantees for*
693 *compressive clustering and compressive mixture modeling*, Mathematical Statistics and Learning, 3
694 (2021), pp. 165–257.
- 695 [19] R. GRIBONVAL, A. CHATALIC, N. KERIVEN, V. SCHELLEKENS, L. JACQUES, AND P. SCHNITER, *Sketching*
696 *datasets for large-scale learning (long version)*, arXiv preprint arXiv:2008.01839, (2020).
- 697 [20] R. GRIBONVAL, A. CHATALIC, N. KERIVEN, V. SCHELLEKENS, L. JACQUES, AND P. SCHNITER, *Sketching*
698 *data sets for large-scale learning: Keeping only what you need*, IEEE Signal Processing Magazine, 38
699 (2021), pp. 12–36.
- 700 [21] J. HERTRICH, D. P. L. NGUYEN, J.-F. AUJOL, D. BERNARD, Y. BERTHOUMIEU, A. SAADALDIN, AND
701 G. STEIDL, *PCA reduced gaussian mixture models with applications in superresolution*, arXiv preprint
702 arXiv:2009.07520, (2020).
- 703 [22] A. HOUDARD, C. BOUYEYRON, AND J. DELON, *High-dimensional mixture models for unsupervised image*
704 *denoising (HDMI)*, SIAM Journal on Imaging Sciences, 11 (2018), pp. 2815–2846.
- 705 [23] P. JAIN, A. TEWARI, AND I. S. DHILLON, *Orthogonal matching pursuit with replacement*, arXiv preprint
706 arXiv:1106.2774, (2011).
- 707 [24] N. KERIVEN, *SketchMLbox – A MATLAB toolbox for large-scale mixture learning*, Mar. 2018, <https://hal.inria.fr/hal-02960718>.
- 708 [25] N. KERIVEN, A. BOURRIER, R. GRIBONVAL, AND P. PÉREZ, *Sketching for large-scale learning of mixture*
709 *models*, Information and Inference: A Journal of the IMA, 7 (2018), pp. 447–508.
- 710 [26] N. KERIVEN, A. DELEFORGE, AND A. LIUTKUS, *Blind source separation using mixtures of alpha-stable*
711 *distributions*, in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing
712 (ICASSP), IEEE, 2018, pp. 771–775.
- 713 [27] N. KERIVEN, N. TREMBLAY, Y. TRAONMILIN, AND R. GRIBONVAL, *Compressive k-means*, in 2017
714 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017,
715 pp. 6369–6373.
- 716 [28] F. LAUS, M. NIKOLOVA, J. PERSCH, AND G. STEIDL, *A nonlocal denoising algorithm for manifold-valued*
717 *images using second order statistics*, 2016.
- 718 [29] M. LEBRUN, *An analysis and implementation of the bm3d image denoising method*, Image Processing On
719 Line, 2012 (2012), pp. 175–213.
- 720 [30] M. LEBRUN, A. BUADES, AND J.-M. MOREL, *Implementation of the “non-local bayes” (nl-bayes) image*
721 *denoising algorithm*, Image Processing On Line, 2013 (2013), pp. 1–42.
- 722 [31] M. LEBRUN, A. BUADES, AND J.-M. MOREL, *A nonlocal bayesian image denoising algorithm*, SIAM
723 Journal on Imaging Sciences, 6 (2013), pp. 1665–1688.
- 724 [32] A. LEVIN AND B. NADLER, *Natural image denoising: Optimality and inherent bounds*, in CVPR 2011,
725 IEEE, 2011, pp. 2833–2840.
- 726 [33] E. LUO, S. H. CHAN, AND T. Q. NGUYEN, *Adaptive image denoising by mixture adaptation*, IEEE
727 transactions on image processing, 25 (2016), pp. 4489–4503.
- 728 [34] S. G. MALLAT AND Z. ZHANG, *Matching pursuits with time-frequency dictionaries*, IEEE Transactions
729 on signal processing, 41 (1993), pp. 3397–3415.
- 730 [35] D. MARTIN, C. FOWLKES, D. TAL, AND J. MALIK, *A database of human segmented natural images and its*
731 *application to evaluating segmentation algorithms and measuring ecological statistics*, in Proceedings
732 8th International Conference on Computer Vision. ICCV 2001, vol. 2, IEEE, 2001, pp. 416–423.
- 733 [36] G. J. MCLACHLAN AND T. KRISHNAN, *The EM algorithm and extensions*, vol. 382, John Wiley & Sons,
734 2007.
- 735 [37] V. POPYAN AND M. ELAD, *Multi-scale patch-based image restoration*, IEEE Transactions on image pro-
736 cessing, 25 (2015), pp. 249–261.
- 737 [38] S. PARAMESWARAN, C.-A. DELEDALLE, L. DENIS, AND T. Q. NGUYEN, *Accelerating gmm-based patch*
738 *priors for image restoration: Three ingredients for a 100× speed-up*, IEEE Transactions on Image
739 Processing, 28 (2018), pp. 687–698.
- 740 [39] Y. C. PATI, R. REZAIIFAR, AND P. S. KRISHNAPRASAD, *Orthogonal matching pursuit: Recursive function*
741 *approximation with applications to wavelet decomposition*, in Proceedings of 27th Asilomar conference
742 on signals, systems and computers, IEEE, 1993, pp. 40–44.
- 743

- 744 [40] O. PERMIAKOVA AND T. BURGER, *Sketched stochastic dictionary learning for large-scale data and appli-*
745 *cation to high-throughput mass spectrometry*, Statistical Analysis and Data Mining: The ASA Data
746 Science Journal, (2021).
- 747 [41] Y. REN, Y. ROMANO, AND M. ELAD, *Example-based image synthesis via randomized patch-matching*,
748 IEEE Transactions On Image Processing, 27 (2017), pp. 220–235.
- 749 [42] F. RENNA, R. CALDERBANK, L. CARIN, AND M. R. RODRIGUES, *Reconstruction of signals drawn from*
750 *a gaussian mixture via noisy compressive measurements*, IEEE Transactions on Signal Processing, 62
751 (2014), pp. 2265–2277.
- 752 [43] S. ROTH AND M. J. BLACK, *Fields of experts: A framework for learning image priors*, in 2005 IEEE
753 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 2, IEEE,
754 2005, pp. 860–867.
- 755 [44] A. SAINT-DIZIER, J. DELON, AND C. BOUYEYRON, *A unified view on patch aggregation*, Journal of
756 Mathematical Imaging and Vision, 62 (2020), pp. 149–168.
- 757 [45] V. SCHELLEKENS AND L. JACQUES, *Compressive classification (machine learning without learning)*, arXiv
758 preprint arXiv:1812.01410, (2018).
- 759 [46] V. SCHELLEKENS AND L. JACQUES, *Compressive learning of generative networks*, arXiv preprint
760 arXiv:2002.05095, (2020).
- 761 [47] V. SCHELLEKENS AND L. JACQUES, *When compressive learning fails: blame the decoder or the sketch?*,
762 arXiv preprint arXiv:2009.08273, (2020).
- 763 [48] M. P. SHEEHAN, M. S. KOTZAGIANNIDIS, AND M. E. DAVIES, *Compressive independent component*
764 *analysis*, in 2019 27th European Signal Processing Conference (EUSIPCO), IEEE, 2019, pp. 1–5.
- 765 [49] M. P. SHEEHAN, J. TACHELLA, AND M. E. DAVIES, *A sketching framework for reduced data transfer in*
766 *photon counting lidar*, arXiv preprint arXiv:2102.08732, (2021).
- 767 [50] H. SHI, <https://github.com/shihui1224/sketching-for-denoising>.
- 768 [51] H. SHI, Y. TRAONMILIN, AND J.-F. AUJOL, *Sketched learning for image denoising*, in The Eighth Inter-
769 national Conference on Scale Space and Variational Methods in Computer Vision (SSVM), Cabourg,
770 France, May 2021.
- 771 [52] J. SULAM AND M. ELAD, *Expected patch log likelihood with a sparse prior*, in Energy Minimization
772 Methods in Computer Vision and Pattern Recognition, X.-C. Tai, E. Bae, T. F. Chan, and M. Lysaker,
773 eds., Cham, 2015, Springer International Publishing, pp. 99–111.
- 774 [53] M. E. TIPPING AND C. M. BISHOP, *Mixtures of probabilistic principal component analyzers*, Neural
775 computation, 11 (1999), pp. 443–482.
- 776 [54] D.-V. TRAN, S. LI-THIAO-TÉ, M. LUONG, T. LE-TIEN, AND F. DIBOS, *Number of useful components in*
777 *gaussian mixture models for patch-based image denoising*, in Image and Signal Processing, A. Man-
778 souri, A. El Moataz, F. Nouboud, and D. Mammass, eds., Cham, 2018, Springer International Pub-
779 lishing, pp. 108–116.
- 780 [55] Y. TRAONMILIN, J. AUJOL, AND A. LECLAIRE, *The basins of attraction of the global minimizers of non-*
781 *convex inverse problems with low-dimensional models in infinite dimension*, Preprint, abs/2009.08670
782 (2020).
- 783 [56] Y.-Q. WANG AND J.-M. MOREL, *Sure guided gaussian mixture image denoising*, SIAM Journal on Imag-
784 ing Sciences, 6 (2013), pp. 999–1034.
- 785 [57] Z. WANG, A. BOVIK, H. SHEIKH, AND E. SIMONCELLI, *Image quality assessment: from error visibility*
786 *to structural similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612.
- 787 [58] J. XU, L. ZHANG, W. ZUO, D. ZHANG, AND X. FENG, *Patch group based nonlocal self-similarity prior*
788 *learning for image denoising*, in Proceedings of the IEEE international conference on computer vision,
789 2015, pp. 244–252.
- 790 [59] J. YANG, X. YUAN, X. LIAO, P. LLULL, D. J. BRADY, G. SAPIRO, AND L. CARIN, *Video compressive*
791 *sensing using gaussian mixture models*, IEEE Transactions on Image Processing, 23 (2014), pp. 4863–
792 4878.
- 793 [60] G. YU, G. SAPIRO, AND S. MALLAT, *Solving inverse problems with piecewise linear estimators: From*
794 *gaussian mixture models to structured sparsity*, IEEE Transactions on Image Processing, 21 (2011),
795 pp. 2481–2499.
- 796 [61] K. ZHANG, W. ZUO, Y. CHEN, D. MENG, AND L. ZHANG, *Beyond a gaussian denoiser: Residual learning*
797 *of deep cnn for image denoising*, IEEE transactions on image processing, 26 (2017), pp. 3142–3155.

- 798 [62] P. ZHANG AND Y. GAO, *Matrix multiplication on high-density multi-gpu architectures: Theoretical and ex-*
799 *perimental investigations*, in IEEE International Conference on High Performance Computing, Data,
800 and Analytics, 2015.
- 801 [63] D. ZORAN AND Y. WEISS, *From learning models of natural image patches to whole image restoration*, in
802 2011 International Conference on Computer Vision, IEEE, 2011, pp. 479–486.