A unified framework of non-local parametric methods for image denoising

Sébastien Herbreteau, Charles Kervrann

A UNIFIED FRAMEWORK OF NON-LOCAL PARAMETRIC METHODS FOR IMAGE DENOISING

Sébastien Herbreteau, Charles Kervrann
Centre Inria de l'Université de Rennes, France
{sebastien.herbreteau, charles.kervrann}@inria.fr

ABSTRACT

We propose a unified view of non-local methods for single-image denoising, for which BM3D is the most popular representative, that operate by gathering noisy patches together according to their similarities in order to process them collaboratively. Our general estimation framework is based on the minimization of the quadratic risk, which is approximated in two steps, and adapts to photon and electronic noises. Relying on unbiased risk estimation (URE) for the first step and on "internal adaptation", a concept borrowed from deep learning theory, for the second, we show that our approach enables to reinterpret and reconcile previous state-of-the-art non-local methods. Within this framework, we propose a novel denoiser called NL-Ridge that exploits linear combinations of patches. While conceptually simpler, we show that NL-Ridge can outperform well-established state-of-the-art single-image denoisers.

1 Introduction

Over the years, a diverse range of strategies, tools, and theories have been proposed to solve the image denoising problem, at the crossroads of statistics, signal processing, optimization, and functional analysis. However, the recent immense influence on this field stems from the advancements in machine learning techniques, particularly deep neural networks. By framing denoising as a regression problem, the task essentially involves training a network to map the corrupted image to its source. This paradigm shift has revolutionized denoising and various other inverse problems in computer vision. Since then, numerous supervised neural networks have been introduced for image denoising [59, 60, 52, 58, 56, 57, 32, 40, 7, 35, 33, 46], achieving state-of-the-art performances.

However, these supervised approaches, apart from their demanding optimization phase, face challenges due to their heightened sensitivity to the quality of the training data. These methods require a training dataset that not only contains a diverse range of examples but also must be abundant and representative of various image scenarios. Otherwise, subpar or entirely inaccurate outcomes may result. Consequently, their application becomes impractical in certain scenarios, particularly when noise-free images are unavailable (although training on datasets composed of noisy/noisy image pairs was studied in [29, 42, 45, 23, 24, 1]). In an attempt to overcome this hurdle, single-image learning—an approach in which only the input noisy image is utilized for training—was explored with deep neural networks [48, 30, 1, 39, 20, 31] as an alternative strategy. However, their performance is still limited compared to their conventional counterparts [8, 26, 13, 10, 9, 12, 21, 19, 55, 34]. Within dataset-free image denoising, BM3D [8] continues to serve as a benchmark method and maintains competitiveness even though it was introduced approximately fifteen years ago. Relying on the non-local strategy, the mechanism of BM3D involves collaboratively processing groups of similar noisy patches distributed throughout the image. Subsequently, numerous methods rooted in patch grouping have emerged, such as [26, 13, 9, 10, 55].

In this paper, we present a unified view, based on quadratic risk minimization, to reinterpret and reconcile two-step non-local methods [26, 8] from families of parameterized functions. We focused on BM3D [8] and NL-Bayes [26] as they are considered as the best performing and fastest unsupervised methods in image denoising so far. In our estimation framework, no prior model for the distribution on patches is required. Second, derived from this framework, we propose a novel denoiser called NL-Ridge that exploits linear combinations of patches. We show that the resulting

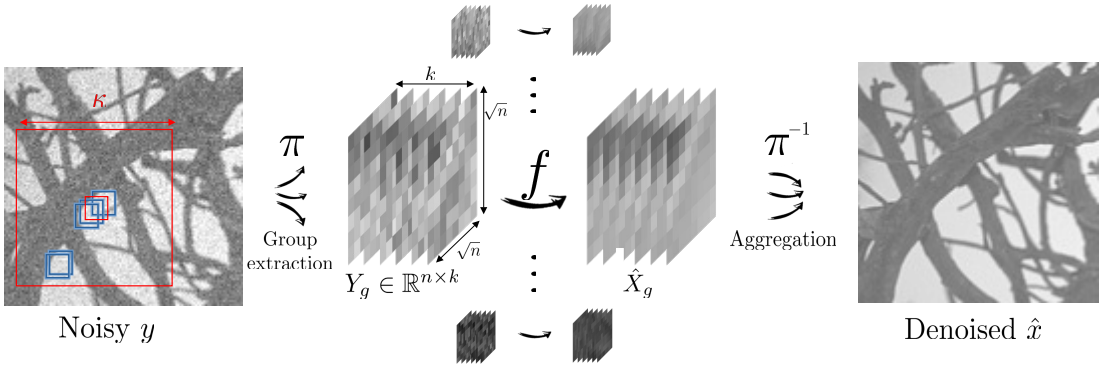


Figure 1: Illustration of the grouping technique for image denoising.

algorithm may outperform BM3D [8] and NL-Bayes [26], as well as several single-image deep-learning methods [48, 30, 1, 39, 20, 31], while being simpler conceptually.

The remainder of this paper is organized as follows. In Section 2, we construct NL-Ridge algorithm from the family of parameterized functions that processes patches by linear combinations, whether constrained or not. In Section 3, we show that when considering two specific families of functions, NL-Bayes [26] and BM3D [8] can be fully reinterpreted within our statistical framework. Finally, in Section 4, we demonstrate on artificially noisy but also real-noisy images that NL-Ridge compares favorably with its well-established counterparts [8, 26], despite relying only on linear combinations of patches.

2 NL-Ridge for image denoising

Popularized by BM3D [8], the grouping technique, sometimes referred to as block-matching, has proven to be a key element in achieving state-of-the-art performances in single-image denoising [8, 26, 13, 9, 38, 10, 17, 28]. This technique consists in gathering noisy patches together according to their similarities in order to denoise them collaboratively. First, groups of k similar noisy square patches $\sqrt{n} \times \sqrt{n}$ are extracted from the noisy image y. Specifically, for each overlapping patch y_g taken as reference, the similarity (*e.g.*, in the ℓ_2 sense) with its surrounding overlapping patches (*e.g.*, within a local window $\kappa \times \kappa$) is computed and the k-nearest neighbors, including the reference patch, are then selected to form a so-called similarity matrix $Y_g \in \mathbb{R}^{n \times k}$, where each column represents a flattened patch. Subsequently, all groups are processed in parallel by applying a local denoising function f that produces an estimate for each noise-free similarity matrix: $\hat{X}_g = f(Y_g) \in \mathbb{R}^{n \times k}$. Finally, the denoised patches are repositioned to their initial locations in the image and aggregated, or reprojected [49], as pixels may have several estimates, to build the final estimated image $\hat{\mathcal{I}}$. Generally, arithmetic (sometimes weighed) averaging of all estimates for a same underlying pixel is used to that end. Figure 1 displays the whole process, summarized in Algorithm 1 for a given local denoising function f.

Algorithm 1 Non-local methods for image denoising

Input: Noisy image y, patch size $\sqrt{n}$, group size k
for each $\sqrt{n} \times \sqrt{n}$ overlapping noisy patch in y **do**
  Extract its k most similar patches to form similarity matrix Y_g
  Perform collaborative denoising $\hat{X}_g = f(Y_g)$
end for
Aggregate all the denoised patches contained in the groups $\hat{X}_g$ to form the image $\hat{\mathcal{I}}$
return $\hat{\mathcal{I}}$

Within this framework, the choice of the local denoising function f remains an open question. A majority of state-of-the-art methods leverage the inherent sparsity of natural images for the design of f [8, 38, 13, 9, 10, 17]. For example, BM3D [8] and LSSC [38] assume a locally sparse representation of the similarity matrices in a predetermined basis or dictionary (wavelets or DCT) while others rather adopt a low-rank approach [13, 9, 10, 17]. As for NL-Bayes [26], a Bayesian framework is exploited at the patch level, in which f produces a maximum a posteriori probability (MAP) estimate.

2.1 Parametric linear patch combinations

Focusing on functions that perform locally linear combinations of patches, f is chosen among the set of the following parameterized functions for each input similarity matrix Y_g:

$$f_\Theta : Y \in \mathbb{R}^{n\times k} \mapsto Y\Theta \tag{1}$$

where $\Theta \in \mathbb{R}^{k\times k}$. Essentially, the k columns of matrix Θ encode the weights of the k different linear combinations aimed to be applied for patch group denoising. Note that parameters Θ may change from one similarity matrix Y_g to another, so that as many different matrices Θ as there are similarity matrices Y_g may be chosen. According to the constraints imposed on the combination weights, the search space for parameters Θ is restricted to subsets of $\mathbb{R}^{k\times k}$ as follows ($\mathbf{1}_k$ denotes the k-dimensional all-ones vector):

- **linear combinations** of patches: $\Theta \in \mathbb{R}^{k\times k}$.
- **affine combinations** of patches: $\Theta \in \mathbb{R}^{k\times k}$ s.t. $\Theta^\top \mathbf{1}_k = \mathbf{1}_k$.
- **conical combinations** of patches: $\Theta \in \mathbb{R}^{k\times k}$ s.t. $\Theta \succeq 0$.
- **convex combinations** of patches: $\Theta \in \mathbb{R}^{k\times k}$ s.t. $\Theta^\top \mathbf{1}_k = \mathbf{1}_k$ and $\Theta \succeq 0$.

Aggregating similar patches via a linear (in general convex) combination has already been exploited in the past [5, 21, 19, 11]. However, the originality of our approach lies in the way of computing the weights Θ, which significantly boosts performance.

2.2 Parameter optimization

In what follows, for the sake of notation simplicity, the index g from Y_g designating the group patch is removed. Thus, $Y \in \mathbb{R}^{n\times k}$ denotes any similarity matrix resulting of the corruption of X, its associated clean similarity matrix, by the underlying noise model (*e.g.*, Gaussian noise, Poisson noise, ...).

For each patch group Y, f_Θ is found by minimizing the local quadratic risk defined as:

$$\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2 . \tag{2}$$

In other words, we look for the Minimum-Mean-Squared-Error (MMSE) estimator f_{Θ^*} among the family of functions $(f_\Theta)_{\Theta\in\mathbb{R}^{k\times k}}$ defined in (1). The optimal estimator f_{Θ^*} minimizes the risk, *i.e.*

$$\Theta^* = \arg\min_\Theta \ \mathcal{R}_\Theta(X). \tag{3}$$

Unfortunately, Θ^* requires the knowledge of X which is unknown. The good news is that the risk $\mathcal{R}_\Theta(X)$ can be approximated through the following two-step algorithm:

1. In Step 1, an approximation of Θ^* is computed for each group of similar patches, through the use of an unbiased risk estimate (URE) of $\mathcal{R}_\Theta(X)$. After reprojection [49] of all denoised patches, a first denoised image $\hat{\mathcal{I}}^{(1)}$ is obtained.
2. In Step 2, $\hat{\mathcal{I}}^{(1)}$ is improved with a second estimation of Θ^* which is found thanks to the technique of "internal adaptation" described in [52] to eventually form $\hat{\mathcal{I}}^{(2)}$.

In what follows, we focus on three different types of noise:

- Gaussian noise: $Y_{i,j} \sim \mathcal{N}(X_{i,j}, V_{i,j})$ with $V \in (\mathbb{R}_*^+)^{n\times k}$ indicating the variance per pixel, sometimes referred to as "noisemap". In particular, for homoscedastic Gaussian noise, $V = \sigma^2 \mathbf{1}_n \mathbf{1}_k^\top$ where $\sigma > 0$ is the standard deviation, that is $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$,
- Poisson noise: $Y_{i,j} \sim \mathcal{P}(X_{i,j})$,
- Mixed Poisson-Gaussian noise: $Y_{i,j} \sim a\mathcal{P}(X_{i,j}/a) + \mathcal{N}(0, b)$ with $(a, b) \in (\mathbb{R}_*^+)^2$.

Note that in each case, $Y_{i,j}$ follows a noise model which is centered on $X_{i,j}$, *i.e.* $\mathbb{E}(Y_{i,j}) = X_{i,j}$, and, as the noise is assumed to be independent at each pixel, the $Y_{i,j}$ are independent along each column. Furthermore, the $Y_{i,j}$ are also independent along each row since there are no duplicate patches in each group.

2.3 Step 1: Unbiased risk estimate (URE)

Recall that our objective is to get an approximation of Θ^* from (3). While $\mathcal{R}_\Theta(X)$ is unattainable in practice, an estimate of this quantity can be calculated instead when dealing with the common types of noise given above. In any case, an unbiased estimate of the risk $\mathcal{R}_\Theta(X)$ is given by (see Appendix B.2):

$$\begin{aligned}\mathrm{URE}_\Theta(Y) &= \|Y\Theta - Y\|_F^2 + 2\,\mathrm{tr}(D_1\Theta) - \mathrm{tr}(D_1) \\ &= 2\,\mathrm{tr}\left(\frac{1}{2}\Theta^\top Q_1\Theta + C_1\Theta\right) + \mathrm{const}\,,\end{aligned} \tag{4}$$

where $Q_1 = Y^\top Y$ is a positive semi-definite matrix, $C_1 = D_1 - Q_1$ and D_1 is a diagonal matrix that depends on the type of noise:

$$D_1 = \begin{cases} \mathrm{diag}(V^\top \mathbf{1}_n) & \text{for Gaussian noise,} \\ \mathrm{diag}(Y^\top \mathbf{1}_n) & \text{for Poisson noise,} \\ \mathrm{diag}((aY+b)^\top \mathbf{1}_n) & \text{for mixed Poisson-Gaussian noise.} \end{cases} \tag{5}$$

Interestingly, (4) gives an unbiased estimate of the risk $\mathcal{R}_\Theta(X)$ that does not depend on X, but only on the observation Y. A common idea that has been previously exploited in image denoising, mainly for homoscedastic Gaussian noise (*e.g.*, see [2, 36, 3, 53, 54]), is to use such an estimate as a surrogate for minimizing the risk $\mathcal{R}_\Theta(X)$ in (3) which is inaccessible.

2.3.1 Minimization of the surrogate

By minimizing (4) with respect to Θ and assuming that $Q_1 = Y^\top Y$ is positive-definite, we get the following closed-form solutions, depending on whether affine combination constraints are imposed or not (see Proposition 5):

$$\begin{cases} \hat{\Theta}^{(1)}_{lin} = \underset{\Theta\in\mathbb{R}^{k\times k}}{\arg\min}\, \mathrm{URE}_\Theta(Y) = I_k - Q_1^{-1}D_1\,, \\ \hat{\Theta}^{(1)}_{aff} = \underset{\substack{\Theta\in\mathbb{R}^{k\times k} \\ \text{s.t. } \Theta^\top\mathbf{1}_k = \mathbf{1}_k}}{\arg\min}\, \mathrm{URE}_\Theta(Y) = I_k - \left[Q_1^{-1} - \dfrac{Q_1^{-1}\mathbf{1}_k(Q_1^{-1}\mathbf{1}_k)^\top}{\mathbf{1}_k^\top Q_1^{-1}\mathbf{1}_k}\right] D_1\,. \end{cases} \tag{6}$$

In the case of conical and convex combination constraints, there exist no closed-form solution. However, by noticing that:

$$\mathrm{tr}\left(\frac{1}{2}\Theta^\top Q_1\Theta + C_1^\top\Theta\right) = \sum_{j=1}^{k}\frac{1}{2}\theta_j^\top Q_1\theta_j + c_j^\top\theta_j\,, \tag{7}$$

where θ_j and c_j denotes the j^{th} column of Θ and C_1, respectively, minimizing (4) under conical or convex combination constraints is nothing else that solving k independent convex quadratic programming subproblems with linear constraints. Note that quadratic programs can always be solved in a finite amount of computation [44]. Indeed, if the contents of the optimal active set (the set identifying the active constraints in the set of inequality constraints) were known in advance, we could express the active constraints as equality constraints, thereby transforming the inequality-constrained problem into a simpler equality-constrained subproblem, which in our case has a closed-form solution by exploiting the Karush–Kuhn–Tucker conditions. Thus, if time computation were not an issue, we could iterate over all active sets (there are 2^k in our case, since there are k inequality constraints) in the search for this optimal active set, solve the associated equality-constrained subproblem, and finally select the best solution among the 2^k potential candidates. Hopefully, a variety of algorithms have been developed to speed up this naive heuristic, including active-set, interior-point, or gradient projection methods [44]. Interestingly, active-set methods solve the quadratic programming problem exactly by cleverly exploring the active sets, although they are much slower on large problems than the other methods [44].

In conclusion, even though conical and convex combination weights, $\hat{\Theta}^{(1)}_{cnl}$ and $\hat{\Theta}^{(1)}_{cvx}$ respectively, do not have closed-form expressions, they can be found exactly in a finite amount of computation using active-set methods.

2.3.2 On the positive definiteness of Q_1

Positive definiteness of Q_1 is important to ensure the uniqueness of the minimizer of the URE (4) since Q_1 is the Hessian matrix of the quadratic programming subproblems defined by (7). If Q_1 is positive definite, it means that the objective function is strictly convex, and as it is minimized on a convex set, the solution is unique. A priori $Q_1 = Y^\top Y$ is only positive semi-definite since for all $s \in \mathbb{R}^k$, $s^\top Q_1 s = \|Ys\|_2^2 \geq 0$. However, when $n \geq k$, Q_1 is almost surely

positive definite in general (in particular in the case of ideal additive white Gaussian noise) as almost surely the columns of Y are linearly independent. Indeed, when it is the case $s^\top Q_1 s = \|Ys\|_2^2 = 0 \Rightarrow Ys = 0 \Rightarrow s = 0$ and Q_1 is then positive definite. By the way, the closed-form expressions of the combination weights (6) require the inverse of Q_1, which can be computed efficiently based on Cholesky factorization, exploiting the positive definiteness of Q_1 [22].

For real-world noisy images, the actual noise may deviate from the assumed ideal noise models (in general mixed Poisson-Gaussian noise). Apart from the consequences this may have on the denoising performance, an unfortunate outcome is the non positive definiteness of Q_1, even when $n \geq k$ (think of constant regions of the image that are, for any reason, not affected by the noise: $Y \propto \mathbf{1}_n \mathbf{1}_k^\top$). To remedy to this issue, a possible way is to consider a "noisier" risk compared to (2) defined by:

$$\mathcal{R}_\Theta^{\text{Nr},\alpha}(X) = \mathbb{E}\|f_\Theta(Y + \alpha W) - X\|_F^2 \,, \tag{8}$$

with $\alpha > 0$ and $W \in \mathbb{R}^{n \times k}$ with $W_{i,j} \sim \mathcal{N}(0,1)$ independent. One can prove that (see Proposition 6):

$$\mathcal{R}_\Theta^{\text{Nr},\alpha}(X) = \mathcal{R}_\Theta(X) + \alpha^2 n \|\Theta\|_F^2 \,, \tag{9}$$

hence, an unbiased estimate of $\mathcal{R}_\Theta^{\text{Nr},\alpha}(X)$ is:

$$\text{URE}_\Theta^{\text{Nr},\alpha}(Y) = \text{URE}_\Theta(Y) + \alpha^2 n \|\Theta\|_F^2 = 2 \operatorname{tr}\left(\frac{1}{2}\Theta^\top (Q_1 + \alpha^2 n I_k)\Theta + C_1 \Theta\right) + \text{const} \,, \tag{10}$$

which is exactly the same expression as (4), up to a constant, replacing Q_1 by $Q_1 + \alpha^2 n I_k$. Its minimization is then given by formula (6) by substituting $Q_1 + \alpha^2 n I_k$ for Q_1 and $D_1 + n\alpha^2 I_k$ for D_1, respectively. The main advantage of using the "noisier" risk (8) instead of the usual one (2) resides in the positive definiteness of $Q_1 + \alpha^2 n I_k$ which is always guaranteed. Indeed, for all $s \in \mathbb{R}^k \setminus \{0\}$, $s^\top (Q_1 + \alpha^2 n I_k) s = \|Ys\|_2^2 + \alpha^2 n \|s\|_2^2 > 0$. The choice of α constitutes an hyperparameter. In practice, we want α to be as small as possible since, for $\alpha \to 0$, the minimizer of the "noisier" risk (8) converges to the minimizer of the usual risk (2).

2.3.3 Particular case: homoscedastic Gaussian noise

In the case of homoscedastic Gaussian noise, that is $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$, the expression of the URE is reduced to:

$$\text{SURE}_\Theta(Y) = \|Y\Theta - Y\|_F^2 + 2n\sigma^2 \operatorname{tr}(\Theta) - nk\sigma^2 \,, \tag{11}$$

which is nothing else than Stein's unbiased risk estimate [51]. Considering unconstrained minimization, the estimated optimal weights are:

$$\hat{\Theta}_{lin}^{(1)} = \arg \min_{\Theta \in \mathbb{R}^{k \times k}} \text{SURE}_\Theta(Y) = I_k - n\sigma^2 (Y^\top Y)^{-1} \,. \tag{12}$$

Note that $\hat{\Theta}^{(1)}$ is close to Θ^* as long as the variance of SURE is low. A rule of thumbs used in [3] states that the number of parameters must not be "too large" compared to the number of data in order for the variance of SURE to remain small. In our case, this suggests that $n > k$. This result suggests that NL-Ridge is expected to be efficient during this step if a few large patches are used. This is consistent with the condition for which $Q_1 = Y^\top Y$ is almost surely positive definite.

2.4 Step 2: Internal adaptation

At the end of Step 1, we get a first denoised image $\hat{\mathcal{I}}^{(1)}$ which will serve as a pilot in the second step. Once again, we focus on the solution of (3) to denoise locally similar patches. As X and $\hat{X}^{(1)}$, the corresponding group of similar patches in $\hat{\mathcal{I}}^{(1)}$, are supposed to be close, the "internal adaptation" procedure [52] consists in solving (3) by substituting $\hat{X}^{(1)}$ for X.

Interestingly, for any of the types of noise studied, the quadratic risk (2) has a closed-form expression (see Lemma 3):

$$\mathcal{R}_\Theta(X) = \|X\Theta - X\|_F^2 + \operatorname{tr}(\Theta^\top D_2 \Theta) = 2 \operatorname{tr}\left(\frac{1}{2}\Theta^\top Q_2 \Theta + C_2 \Theta\right) + \text{const} \,, \tag{13}$$

where $Q_2 = X^\top X + D_2$ is a positive semi-definite matrix, $C_2 = D_2 - Q_2$ and D_2 is a diagonal matrix that depends on the type of noise:

$$D_2 = \begin{cases} \operatorname{diag}(V^\top \mathbf{1}_n) & \text{for Gaussian noise,} \\ \operatorname{diag}(X^\top \mathbf{1}_n) & \text{for Poisson noise,} \\ \operatorname{diag}((aX + b)^\top \mathbf{1}_n) & \text{for mixed Poisson-Gaussian noise.} \end{cases} \tag{14}$$

Substituting $\hat{X}^{(1)}$ for X in expression (13), a natural surrogate for $\mathcal{R}_\Theta(X)$ is $\mathcal{R}_\Theta(\hat{X}^{(1)})$. A second approximation of (3) can then be deduced by minimizing this latter. Interestingly, this second estimate $\hat{\Theta}^{(2)}$ produces a significant boost in terms of denoising performance compared to $\hat{\Theta}^{(1)}$. Even if the second step can be iterated but we did not notice improvements in our experiments.

2.4.1 Minimization of the surrogate

By minimizing (13) where X is replaced by $\hat{X}^{(1)}$ with respect to Θ and assuming that Q_2 is positive-definite, we get the following closed-form solutions, depending on whether affine combination constraints are imposed or not (see Proposition 1):

$$\begin{cases} \hat{\Theta}^{(2)}_{lin} = \underset{\Theta\in\mathbb{R}^{k\times k}}{\arg\min}\, \mathcal{R}_\Theta(\hat{X}^{(1)}) = I_k - Q_2^{-1}D_2\,, \\ \hat{\Theta}^{(2)}_{aff} = \underset{\substack{\Theta\in\mathbb{R}^{k\times k} \\ \text{s.t. } \Theta^\top \mathbf{1}_k = \mathbf{1}_k}}{\arg\min}\, \mathcal{R}_\Theta(\hat{X}^{(1)}) = I_k - \left[Q_2^{-1} - \frac{Q_2^{-1}\mathbf{1}_k(Q_2^{-1}\mathbf{1}_k)^\top}{\mathbf{1}_k^\top Q_2^{-1}\mathbf{1}_k}\right] D_2\,. \end{cases} \tag{15}$$

As in Step 1, there is no closed-form solution in the case of conical and convex combination constraints. $\hat{\Theta}^{(2)}_{cnl}$ and $\hat{\Theta}^{(2)}_{cvx}$ can however be approximated using any of iterative algorithms [44] dedicated to the resolution of convex quadratic programming problems.

2.4.2 On the positive definiteness of Q_2

Compared to Step 1, $Q_2 = \hat{X}^{(1)\top}\hat{X}^{(1)} + D_2$ is much more likely to be positive definite. In fact, as soon as D_2 has positive diagonal elements, which is always the case except for Poisson noise with $\hat{X}^{(1)} = 0$, Q_2 is positive definite. But this case can be treated separately by setting arbitrarily $\hat{\Theta}^{(2)} = I_k$ for example.

2.4.3 Particular case: homoscedastic Gaussian noise

In the case of homoscedastic Gaussian noise, that is $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$, the expression of the quadratic risk is reduced to:

$$\mathcal{R}_\Theta(X) = \|X\Theta - X\|_F^2 + n\sigma^2\|\Theta\|_F^2\,, \tag{16}$$

which is nothing else than the expression of a multivariate Ridge regression. Considering unconstrained minimization, the estimated optimal weights are then:

$$\hat{\Theta}^{(2)}_{lin} = \arg\min_{\Theta\in\mathbb{R}^{k\times k}} \mathcal{R}_\Theta(\hat{X}^{(1)}) = I_k - n\sigma^2\left(\hat{X}^{(1)\top}\hat{X}^{(1)} + n\sigma^2 I_k\right)^{-1}\,. \tag{17}$$

It is interesting to compare the behavior of the weights $\hat{\Theta}^{(2)}_{lin}$ and $\hat{\Theta}^{(2)}_{aff}$ when σ tends to $+\infty$. In fact, the higher σ and the more important the second term $n\sigma^2\|\Theta\|_F^2$ is in the expression of the risk (16) (and the less the dependence on X). At the limit, when $\sigma \to +\infty$:

$$\begin{cases} \hat{\Theta}^{(2)}_{lin} = \underset{\Theta\in\mathbb{R}^{k\times k}}{\arg\min}\, \mathcal{R}_\Theta(X) \to \mathbf{0}_k\mathbf{0}_k^\top\,, \\ \hat{\Theta}^{(2)}_{aff} = \underset{\substack{\Theta\in\mathbb{R}^{k\times k} \\ \text{s.t. } \Theta^\top \mathbf{1}_k = \mathbf{1}_k}}{\arg\min}\, \mathcal{R}_\Theta(\hat{X}^{(1)}) \to \mathbf{1}_k\mathbf{1}_k^\top/k\,. \end{cases} \tag{18}$$

As a consequence, the final produced image $\hat{\mathcal{I}}^{(2)}$ tends towards the "zero-image" in the case of unconstrained minimization, whereas, in the case of affine combination of patches, $\hat{\mathcal{I}}^{(2)}$ consists of simple averages of similar patches. This fundamental difference in the asymptotic behavior of the weights may explain why affine patch combinations are more recommended when the noise level increases.

2.5 Weighted average reprojection

After the denoising of a group of similar patches, each denoised patch is repositioned at its right location in the image. As several pixels are denoised multiple times, a final step of aggregation, or reprojection [49], is necessary to produce a final denoised image $\hat{\mathcal{I}}^{(1)}$ or $\hat{\mathcal{I}}^{(2)}$. With inspiration from [49], each pixel belonging to column j of Y is assigned,

after denoising, the weight $w_j = 1/(\|\Theta_{\cdot,j}\|_2^2)$. Those weights are at the end pixel-wise normalized such that the sum of all weights associated to a same pixel equals one.

The complete NL-Ridge method for image denoising is summarized in Algorithm 2. Please note the difference between a freshly denoised similarity matrix, denoted $\check{X}_g$, and its aggregated equivalent $\hat{X}_g$.

Algorithm 2 NL-Ridge algorithm for image denoising

Input: Noisy image y, patch and group sizes for step 1 and step 2: $\sqrt{n_1}$, $\sqrt{n_2}$, k_1 and k_2
`/* Step 1`
for each $\sqrt{n_1} \times \sqrt{n_1}$ overlapping noisy patch in y **do**
  Extract its k_1 most similar patches to form similarity matrix $Y_g^{(1)}$
  Estimate combination weights $\hat{\Theta}_g^{(1)}$ with formula (6) *(closed-form expression)*
  Perform collaborative denoising $\check{X}_g^{(1)} = Y_g^{(1)}\hat{\Theta}_g^{(1)}$
end for
Aggregate all the denoised patches contained in the groups $\check{X}_g^{(1)}$ to form the estimated image $\hat{\mathcal{I}}^{(1)}$
`/* Step 2`
for each $\sqrt{n_2} \times \sqrt{n_2}$ overlapping patch in $\hat{\mathcal{I}}^{(1)}$ **do**
  Extract its k_2 most similar patches in $\hat{\mathcal{I}}^{(1)}$ to form similarity matrix $\hat{X}_g^{(1)}$
  Extract the k_2 corresponding noisy patches in y to form similarity matrix $Y_g^{(2)}$
  Estimate combination weights $\hat{\Theta}_g^{(2)}$ with (15) *(closed-form expression)*
  Perform collaborative denoising $\check{X}_g^{(2)} = Y_g^{(2)}\hat{\Theta}_g^{(2)}$
end for
Aggregate all the denoised patches contained in the groups $\check{X}_g^{(2)}$ to form the estimated image $\hat{\mathcal{I}}^{(2)}$
return $\hat{\mathcal{I}}^{(2)}$

3 A unified view of non-local denoisers

In NL-Ridge, the local denoiser f_Θ is arbitrarily of the form given by (1) involving the linear combinations of similar patches with closed-form aggregation weights given in (6) and (15) for unconstrained minimization. In this section, we show that NL-Ridge can serve to interpret two popular state-of-the-art non-local methods - NL-Bayes [26] and BM3D [8] - which were originally designed with two very different modeling and estimation frameworks. It amounts actually to considering two particular families (f_Θ) of local denoisers. In the rest, we focus exclusively on homoscedastic Gaussian noise, that is $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$. All the proofs of this section can be found in Appendix C and Appendix D.

3.1 Analysis of NL-Bayes algorithm

The NL-Bayes [26] algorithm has been established in the Bayesian setting and the resulting maximum a posteriori estimator is computed with a two-step procedure as NL-Ridge. Adopting a novel parametric view of this algorithm, let us consider the following family of local denoisers as starting point:

$$f_{\Theta,\beta} : Y \in \mathbb{R}^{n\times k} \mapsto \Theta Y + \beta \mathbf{1}_k^\top \tag{19}$$

where $\Theta \in \mathbb{R}^{n\times n}$ and $\beta \in \mathbb{R}^n$. Our objective is to find (Θ^*, β^*) that minimizes the quadratic risk $\mathcal{R}_{\Theta,\beta}(X) = \mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2$, that is:

$$\Theta^*, \beta^* = \arg\min_{\Theta,\beta} \mathcal{R}_{\Theta,\beta}(X)\,. \tag{20}$$

3.1.1 Step 1: Unbiased risk estimate (URE)

In the case of Gaussian noise, Stein's unbiased estimate of the quadratic risk $\mathcal{R}_{\Theta,\beta}(X) = \mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2$ is:

$$\mathrm{SURE}_{\Theta,\beta}(Y) = \|\Theta Y - Y + \beta \mathbf{1}_k^\top\|_F^2 + 2k\sigma^2\,\mathrm{tr}(\Theta) - nk\sigma^2\,, \tag{21}$$

which reaches its minimum for:

$$\hat{\Theta}^{(1)} = (C_Y - \sigma^2 I_n)C_Y^{-1} \quad \text{and} \quad \hat{\beta}^{(1)} = (I_n - \hat{\Theta}^{(1)})\mu_Y\,, \tag{22}$$

where $\mu_Y \in \mathbb{R}^k$ and $C_Y \in \mathbb{R}^{n\times n}$ denote the empirical mean and covariance matrix of a group of patches $Y \in \mathbb{R}^{n\times k}$, that is

$$\mu_Y = \frac{1}{k} Y \mathbf{1}_k \quad \text{and} \quad C_Y = \frac{1}{k}(Y - \mu_Y \mathbf{1}_k^\top)(Y - \mu_Y \mathbf{1}_k^\top)^\top . \tag{23}$$

Interestingly, $f_{\hat{\Theta}^{(1)},\hat{\beta}^{(1)}}(Y)$ is the expression given in [26] (Step 1), which is actually derived from the prior distribution of patches assumed to be Gaussian. Furthermore, our framework provides guidance on the choice of the parameters n and k. Indeed, SURE is helpful provided that its variance remains small which is achieved if $n < k$ (the number of parameters must not be "too large" compared to the number of data). This result suggests that NL-Bayes is expected to be efficient if small patches are used, as confirmed in the experiments in [27].

3.1.2 Step 2: "Internal adaptation"

The quadratic risk $\mathcal{R}_{\Theta,\beta}(X)$ associated with the family of functions defined in (19) has a closed-form expression:

$$\mathcal{R}_{\Theta,\beta}(X) = \|\Theta X - X + \beta \mathbf{1}_k^\top\|_F^2 + k\sigma^2 \|\Theta\|_F^2 . \tag{24}$$

Interpreting the second step in [26] as an "internal adaptation" step, we want to minimize the risk $\mathcal{R}_{\Theta,\beta}(X)$ by substituting $\hat{X}^{(1)}$, obtained at the end of step 1, for X, which is unknown. The updated parameters become:

$$\hat{\Theta}^{(2)} = C_{\hat{X}^{(1)}}(C_{\hat{X}^{(1)}} + \sigma^2 I_n)^{-1} \quad \text{and} \quad \hat{\beta}^{(2)} = (I_n - \hat{\Theta}^{(2)})\mu_{\hat{X}^{(1)}} , \tag{25}$$

and $f_{\hat{\Theta}^{(2)},\hat{\beta}^{(2)}}(Y)$ corresponds to the original second-step expression in [26].

3.2 Analysis of BM3D algorithm

BM3D [8] is probably the most popular non-local method for image denoising. It assumes a locally sparse representation of images in a transform domain. A two step algorithm was described in [8] to achieve state-of-the-art results for several years. By using the generic NL-Ridge formulation, we consider the following family of functions:

$$f_\Theta : Y \mapsto P^{-1}(\Theta \odot (PYQ))Q^{-1} \tag{26}$$

where $\Theta \in \mathbb{R}^{n\times m}$ and where $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are two orthogonal matrices that model a separable 3D-transform (typically a 2D and 1D *Discrete Cosine Transform*, respectively). Once again, our objective is to find Θ^* that minimizes the quadratic risk $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$, that is:

$$\Theta^* = \arg\min_\Theta \mathcal{R}_\Theta(X) . \tag{27}$$

3.2.1 Step 1: Unbiased risk estimate (URE)

Stein's unbiased risk estimate (SURE) is defined as follows:

$$\mathrm{SURE}_\Theta(Y) = \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top) \odot PYQ\|_F^2 + 2\sigma^2\langle \Theta, \mathbf{1}_n\mathbf{1}_k^\top\rangle_F - nk\sigma^2 , \tag{28}$$

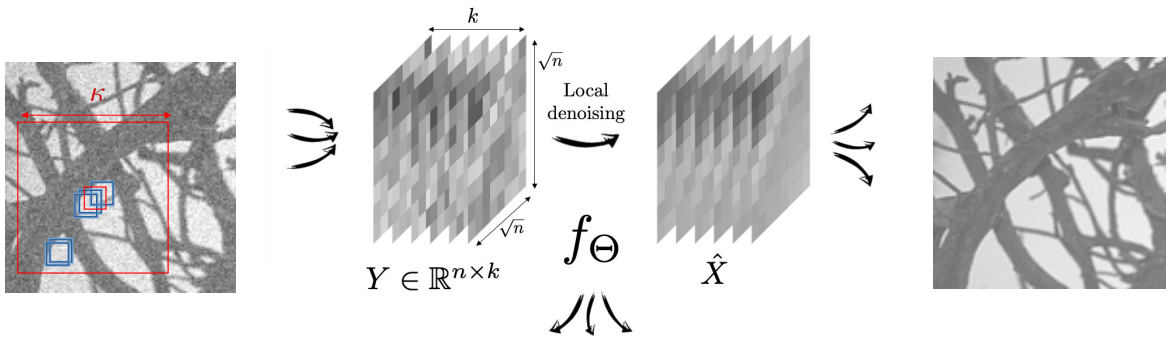
and its minimization yields:

$$\hat{\Theta}_a^{(1)} = \mathbf{1}_n\mathbf{1}_k^\top - \frac{\sigma^2}{(PYQ)^{\odot 2}} , \tag{29}$$

where the division is element-wise. Unfortunately, $f_{\hat{\Theta}_a^{(1)}}(Y)$ does not provide very satisfying denoising results. This result is actually expected as the number of parameters equals the size of data ($n \times k$), making SURE weakly efficient. To overcome this difficulty, we can force the elements of Θ to be either 0 or 1, *i.e.* by imposing the search space to be $\Theta \in \{0,1\}^{n\times k}$. Minimizing SURE under this constraint results in the alternative estimation:

$$\hat{\Theta}_b^{(1)} = \mathbb{1}_{\mathbb{R}\setminus[-\sqrt{2}\sigma,\sqrt{2}\sigma]}(PYQ) . \tag{30}$$

$f_{\hat{\Theta}_b^{(1)}}(Y)$ acts then as a hard thresholding estimator as in BM3D: the coefficients of the transform domain (*i.e.* the elements of the matrix PYQ) below $\sqrt{2}\sigma$, in absolute value, are canceled before applying the inverse 3D-transform. This result suggests that the threshold should be linearly dependent on σ but also that the threshold value is independent on the choice of the orthogonal transforms P and Q. In [8], a threshold value of 2.7σ was carefully chosen in Step 1, which is approximately twice the SURE-prescribed threshold.



BM3D [8] assumes a locally sparse representation in a transform domain:

$$f_\Theta(Y) = P^\top(\Theta \odot (PYQ))Q^\top\,,$$

P, Q: orthogonal matrices.

NL-Bayes [26] was originally established in the Bayesian setting:

$$f_{\Theta,\beta}(Y) = \Theta Y + \beta \mathbf{1}_k^\top\,,$$

$\mathbf{1}_k$: k-dimensional all-ones vector.

NL-Ridge (ours) leverages linear combinations of noisy patches:

$$\boxed{f_\Theta(Y) = Y\Theta}\,.$$

Figure 2: Illustration of the parametric view of several popular non-local denoisers. Examples of parameterized functions unequivocally identifying the denoiser are given, whose optimal parameters are eventually selected for each group of patches by "internal adaptation" optimization.

3.2.2 Step 2: "Internal adaptation"

The quadratic risk $\mathcal{R}_\Theta(X)$ associated with the family of functions defined in (26) has a closed-form expression:

$$\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top) \odot PXQ\|_F^2 + \sigma^2\|\Theta\|_F^2\,, \tag{31}$$

and the "internal adaptation" step yields the same expression as the Wiener filtering used in Step 2 in BM3D [8]:

$$\hat{\Theta}^{(2)} = \frac{(P\hat{X}^{(1)}Q)^2}{\sigma^2 + (P\hat{X}^{(1)}Q)^{\odot 2}}\,. \tag{32}$$

However, it is worth noting that the closed-form expression of the risk (31) is obtained by assuming that the coefficients of Y are all independent. Thus, theoretically, Y should gather together exclusively non-overlapping patches. This important limitation is not yet considered in the original paper [8]. Hopefully, this has little effect on the denoising performance. More recently, a new version of the algorithm has been published that takes into account when the noise in one patch is correlated with the noise in one of the other patches [43].

In summary, we have shown that BM3D [8] and NL-Bayes [26] can be interpreted under a parametric view within NL-Ridge framework in the case of homoscedastic Gaussian noise. Figure 2 summarizes the three different algorithms which are distinguished by their parametric families. Our novel paradigm has some advantages beyond the unification of methods: it enables to set the size of the patches and may potentially relax the need to specify the prior distribution of patches.

4 Experimental results

In this section, we compare the performance of our NL-Ridge method with state-of-the-art methods, including related network-based methods [59, 60, 52, 48, 1, 30, 57, 58, 56, 31, 39, 20] applied to standard gray images artificially corrupted with homoscedastic Gaussian noise with zero mean and variance σ^2 and on real-world noisy images, modeled by mixed Poisson-Gaussian noise. We used the implementations provided by the authors as well as the corresponding trained weights for supervised networks. Performances of NL-Ridge and other methods are assessed in terms of PSNR values when the ground truth is available. Unless specified, NL-Ridge is run without constraint on the weights of the linear combinations. The code can be downloaded at: https://github.com/sherbret/NL-Ridge/.

4.1 Setting of algorithm parameters

For the sake of computational efficiency, the search for similar patches, computed in the ℓ_2 sense, across the image is restricted to a small local window $\kappa \times \kappa$ centered around each reference patch (in our experiments $\kappa = 37$).

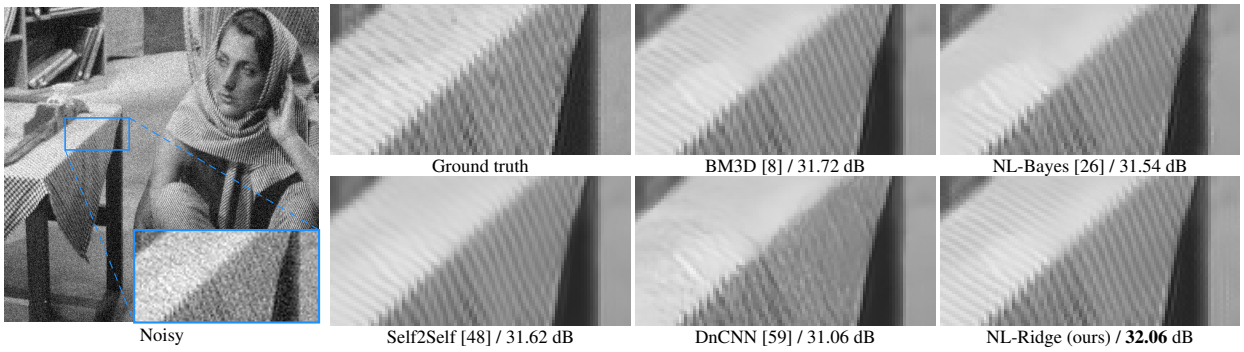


Figure 3: Denoising results (in PSNR) on *Barbara* corrupted with additive white Gaussian noise ($\sigma = 20$).

Table 1: Setting of algorithm parameters (patch size n, patch number k) depending on the noise standard deviation σ values.

σ	n_1	n_2	k_1	k_2
$0 < \sigma \leq 15$	7×7	7×7	18	55
$15 < \sigma \leq 35$	9×9	9×9	18	90
$35 < \sigma \leq 50$	11×11	9×9	20	120

Considering iteratively each overlapping patch of the image as reference patch is also computationally demanding, therefore only an overlapping patch over δ, both horizontally and vertically, is considered as a reference patch. The number of reference patches and thus the time spent searching for similar patches is then divided by δ^2. This common technique [8, 16, 13] is sometimes referred in the literature as the *step trick*. In our experiments, we take $\delta = 4$.

Finally, the choice of the parameters n and k depend on the noise level. Experimentally, bigger patches have to be considered for higher noise levels as well as a higher quantity of patches for the second step. An empirical analysis led us to choose the parameters reported in Table 1 for homoscedastic Gaussian noise of variance σ^2.

4.2 Results on test datasets

We have evaluated the performance of NL-Ridge on artificially noisy images, as well as on real noisy images.

4.2.1 Results on artificially noisy images corrupted by homoscedastic Gaussian noise

We tested the denoising performance of our method on three well-known datasets: Set12, BSD68 [41] and Urban100 [18]. A comparison with state-of-the-art algorithms is reported in Table 2. For the sake of a fair comparison, algorithms are divided into two categories: single-image methods, meaning that these methods (either traditional or deep learning-based) only have access to the input noisy image, and dataset-based methods (*i.e.* supervised neural networks) that require a training phase beforehand and an external dataset. Note that only the single-image extension was considered for Noise2Self [1] and the time-consuming "internal adaptation" option was not used for LIDIA [52].

NL-Ridge, exclusively based on weighted aggregation of noisy patches, performs surprisingly at least as well as its traditional two-step counterparts [8, 26]. It is particularly efficient on Urban100 dataset which contains abundant structural patterns and textures, achieving comparable performances with DnCNN [59] and FFDnet [60], popular supervised networks composed of hundreds of thousands of parameters. It is interesting to note that imposing constraints on the weights of combinations does not bring much improvement. It can even be detrimental in the case of conical or convex combinations of patches. This result is all the more surprising since many denoising algorithms are exclusively based on convex combinations of patches [5, 21, 19]. We note however a slight superiority of the affine version over the unconstrained one at higher noise levels, which can be explained by our observation at the end of Section 2.4. In the rest, only the affine version of NL-Ridge is considered, which has the advantage of being normalization-equivariant in the case of Gaussian noise [14].

Figure 3 illustrates the visual results of different methods. NL-Ridge is very competitive with respect to well-established methods such as BM3D [8]. The self-similarity assumption is particularly useful to recover subtle details such as the stripes on the *Barbara* image that are better reconstructed than DnCNN [59].

Table 2: The PSNR (dB) results of different methods applied to three datasets corrupted with synthetic white Gaussian noise. The best method among each category (dataset-based or single-image) is emphasized in bold.

	Methods	**Set12**	**BSD68**	**Urban100**
	Noise level σ	5 / 15 / 25 / 35 / 50	5 / 15 / 25 / 35 / 50	5 / 15 / 25 / 35 / 50
Dataset-based *Neural network*	DnCNN [59]	37.74 / 32.86 / 30.44 / 28.82 / 27.18	37.71 / 31.73 / 29.23 / 27.69 / 26.23	37.52 / 32.68 / 29.97 / 28.11 / 26.28
	FFDNet [60]	38.11 / 32.75 / 30.43 / 28.92 / 27.32	37.80 / 31.63 / 29.19 / 27.73 / 26.29	38.12 / 32.43 / 29.92 / 28.27 / 26.52
	LIDIA [52]	- / 32.85 / 30.41 / - / 27.19	- / 31.62 / 29.11 / - / 26.17	- / 32.80 / 30.12 / - / 26.51
	DRUNet [58]	38.64 / 33.25 / 30.94 / 29.45 / 27.90	38.07 / 31.91 / 29.48 / 28.00 / 26.59	38.91 / 33.44 / 31.11 / 29.61 / 27.96
	Restormer [56]	**38.70** / 33.42 / 31.08 / **29.57** / 28.01	**38.11** / 31.96 / 29.52 / **28.05** / 26.62	**39.06** / 33.79 / 31.46 / **30.00** / 28.33
	SCUNet [57]	- / **33.43** / **31.09** / - / **28.04**	- / **31.99** / **29.55** / - / **26.67**	- / **33.88** / **31.58** / - / **28.56**
Single-image *Neural network*	DIP [30]	- / 30.12 / 27.54 / - / 24.67	- / 28.83 / 26.59 / - / 24.13	- / - / - / - / -
	N2S [1]	- / 31.01 / 28.64 / - / 25.30	- / 29.46 / 27.72 / - / 24.77	- / - / - / - / -
	N2F [31]	- / 30.21 / 28.17 / - / 25.09	- / 29.82 / 27.79 / - / 25.05	- / - / - / - / -
	ZS-N2N [39]	- / 30.05 / 27.26 / - / 23.56	- / 29.61 / 26.86 / - / 23.46	- / - / - / - / -
	S2S [48]	- / 32.07 / **30.02** / - / 26.49	- / 30.62 / 28.60 / - / 25.70	- / - / - / - / -
	RDIP [20]	- / **32.26** / 29.79 / - / **26.60**	- / **31.21** / **28.78** / - / **25.81**	- / - / - / - / -
Traditional 2-step	BM3D [8]	38.02 / 32.37 / 29.97 / 28.40 / 26.72	37.55 / 31.07 / 28.57 / 27.08 / 25.62	38.30 / 32.35 / 29.70 / 27.97 / 25.95
	NL-Bayes [26]	38.12 / 32.25 / 29.88 / 28.30 / 26.45	37.62 / 31.16 / **28.70** / **27.18** / 25.58	38.33 / 31.96 / 29.34 / 27.61 / 25.56
	NL-Ridge *(linear)*	**38.19** / **32.46** / **30.00** / 28.41 / 26.73	**37.67** / **31.20** / 28.67 / 27.14 / 25.67	**38.56** / 32.53 / 29.90 / 28.13 / 26.29
	NL-Ridge *(affine)*	38.17 / 32.42 / 29.98 / **28.43** / **26.79**	37.65 / 31.18 / 28.68 / 27.16 / **25.71**	38.54 / **32.54** / **29.93** / **28.21** / **26.40**
	NL-Ridge *(conical)*	38.03 / 32.16 / 29.72 / 28.07 / 26.49	37.46 / 30.86 / 28.38 / 26.85 / 25.43	- / - / - / - / -
	NL-Ridge *(convex)*	38.00 / 32.14 / 29.70 / 28.14 / 26.51	37.45 / 30.85 / 28.38 / 26.86 / 25.44	- / - / - / - / -
Multi-step	SS-GMM [34]	38.02 / 32.41 / 29.88 / 28.24 / 26.53	37.17 / 31.28 / 28.76 / 27.19 / 25.71	- / - / - / - / -
	TWSC [55]	38.17 / 32.61 / 30.21 / 28.63 / 26.95	37.67 / 31.28 / 28.75 / 27.24 / 25.76	- / - / - / - / -
	WNNM [13]	**38.36** / **32.70** / **30.26** / **28.69** / **27.05**	**37.80** / **31.37** / **28.83** / **27.30** / **25.87**	**38.77** / **32.97** / **30.39** / **28.70** / **26.83**

Table 3: The PSNR (dB) results on raw data on Darmstadt Noise Dataset (DND) [47]

	Single-image						*Dataset-based*			
Methods	BM3D [8]	NL-Bayes [26]	**NL-Ridge**	KSVD [12]	NCSR [10]	WNNM [13]	MLP [6]	TNRD [7]	FFDNet [60]	DCT2net [15]
PSNR (in dB)	**47.15**	46.84	47.12	46.87	47.07	47.05	45.71	45.70	**47.40**	46.83

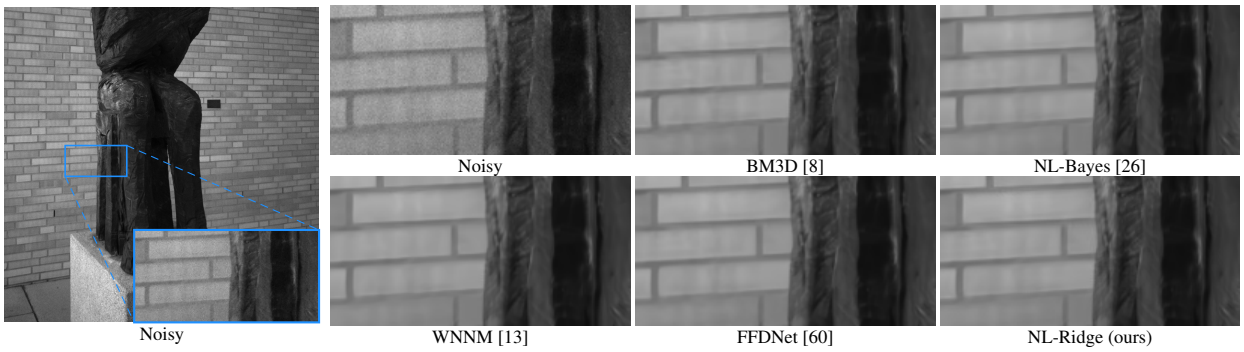


Figure 4: Qualitative comparison of image denoising results on real-world noisy images from Darmstadt Noise Dataset [47]. Zoom-in regions are indicated for each method.

4.2.2 Results on real-world noisy images

We tested the proposed method on the Darmstadt Noise Dataset [47] which is a dataset composed of 50 real-noisy photographs. It relies on captures of a static scene with different ISO values, where the nearly noise-free low-ISO image is carefully post-processed to derive the ground-truth. In this challenge, the ground-truth images are not available. Each competitor submits the denoising results on the official website[1]. The algorithms are then evaluated according to standard metrics and the ranking is made public[2].

The real noise can be modeled as a Poisson-Gaussian noise:

$$y \sim a\mathcal{P}(x/a) + \mathcal{N}(0, b)\,, \tag{33}$$

which can be further approximated with a heteroscedastic Gaussian noise whose variance is intensity-dependent:

$$y \sim \mathcal{N}(x, \mathrm{diag}(ax + b))\,, \tag{34}$$

where $(a, b) \in \mathbb{R}^+ \times \mathbb{R}^+$ are the noise parameters. For each noisy image, the authors [47] calculated the adequate noise parameters (a, b) based on this model and made them available to the user. Note that for applying denoisers exclusively dedicated to homoscedastic Gaussian noise removal, a variance-stabilizing transformation (VST) such as the generalized Anscombe transform [50, 4] is performed beforehand. In our case, stabilizing the variance is not necessary as NL-Ridge can handle mixed Poisson-Gaussian noise directly.

Figure 4 shows a qualitative comparison of the results obtained with state-of-the-art denoisers designed for the framework of additive white Gaussian noise. NL-Ridge does not suffer in comparison with more complex methods such as network-based ones [60]. Table 3 compares the average PSNR values obtained on this dataset for different methods. NL-Ridge obtains comparable results with BM3D [8], which is so far the best unsupervised method on this dataset.

4.3 Complexity

We want to emphasize that NL-Ridge, is relatively fast compared to its traditional and deep-learning-based single-image counterparts. The running times of different state-of-the-art algorithms are reported in Figure 5. It is provided for information purposes only, as the implementation, the language used and the machine on which the code is run, highly influence the execution time. The CPU used is a 2,3 GHz Intel Core i7. Note that NL-Ridge has been entirely implemented in Python with Pytorch, enabling it to run on GPU as well unlike its traditional counterparts, making it even faster. It is worth noting that traditional single-image methods are much less computationally demanding than single-image deep-learning-based ones [20, 48] that use time-consuming gradient descent algorithms for optimization, while traditional ones have closed-form solutions.

5 Conclusion

In this paper, we presented a unified view to reconcile two-step single-image non-local denoisers through the minimization of a risk from a family of estimators, exploiting unbiased risk estimates on the one hand and the "internal

[1] https://noise.visinf.tu-darmstadt.de/

[2] https://noise.visinf.tu-darmstadt.de/benchmark/

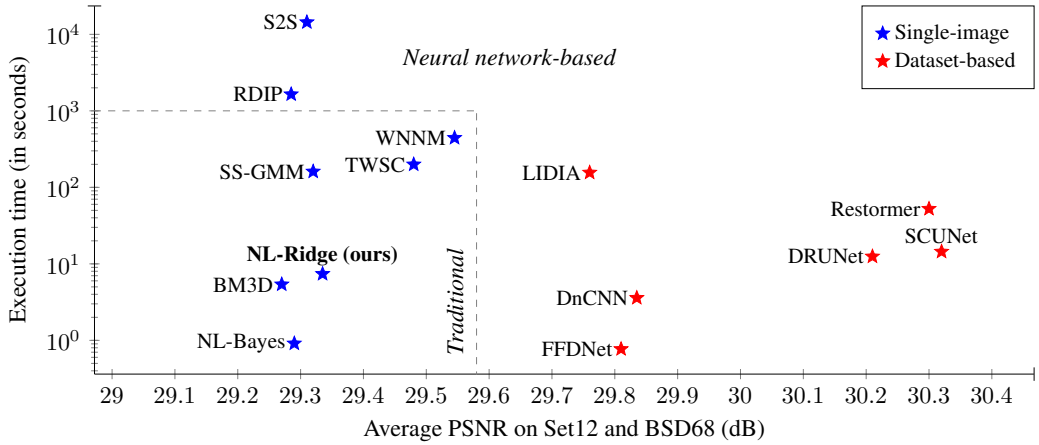


Figure 5: The execution time on CPU for an image of size 512×512 v.s the average PSNR results on Set12 and BSD68 [41] for synthetic Gaussian noise with $\sigma = 25$ of the most effective popular methods [58, 59, 60, 52, 57, 56, 16, 26, 8, 13, 34, 55, 20, 48]. These results are calculated based on Table 2.

adaptation" on the other. We derive NL-Ridge algorithm, which leverages local linear combinations of noisy similar patches. Our experimental results show that NL-Ridge compares favorably with its state-of-the-art counterparts, including recent single-image deep network methods which are much more computationally demanding. Moreover, NL-Ridge is very versatile and can deal with a lot of different types of noise. To the best of our knowledge, NL-Ridge achieves state-of-the-art performance in the field of fast single-image denoising. Admittedly, more efficient methods in terms of PSNR values exist in the literature, notably WNNM [13], but they are based on costly iterative schemes (involving a dozen steps), which may prove prohibitive in certain situations. An interesting line of research for future work is to study the benefits of iterating the linear combinations of patches while keeping a low computational burden.

A Multivariate quadratic programming under affine constraints

Lemma 1 (Multivariate quadratic programming) *Let $Q, C \in \mathbb{R}^{k\times k}$. If Q is symmetric positive definite,*

$$\begin{cases} \underset{\Theta\in\mathbb{R}^{k\times k}}{\arg\min} \quad \operatorname{tr}\left(\frac{1}{2}\Theta^\top Q\Theta + C^\top\Theta\right) = -Q^{-1}C = I_k - Q^{-1}(Q+C)\,, \\ \underset{\substack{\Theta\in\mathbb{R}^{k\times k} \\ \text{s.t. } \Theta^\top \mathbf{1}_k = \mathbf{1}_k}}{\arg\min} \quad \operatorname{tr}\left(\frac{1}{2}\Theta^\top Q\Theta + C^\top\Theta\right) = I_k - \left(Q^{-1} - \frac{Q^{-1}\mathbf{1}_k(Q^{-1}\mathbf{1}_k)^\top}{\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k}\right)(Q+C)\,. \end{cases}$$

Proof: Let θ_j and c_j denote the j^{th} column of matrix $\Theta \in \mathbb{R}^{k\times k}$ and $C \in \mathbb{R}^{k\times k}$, respectively.

First of all,

$$\operatorname{tr}\left(\frac{1}{2}\Theta^\top Q\Theta + C^\top\Theta\right) = \sum_{j=1}^{k}\frac{1}{2}\theta_j^\top Q\theta_j + c_j^\top\theta_j = \sum_{j=1}^{k} h_j(\theta_j)\,.$$

with $h_j : \theta \in \mathbb{R}^k \mapsto \frac{1}{2}\theta^\top Q\theta + c_j^\top\theta$. The minimization problem is then separable and amounts to solve k independent quadratic programming subproblems. As $\operatorname{Hess} h_j(\theta) = Q$ which is symmetric positive definite, h_j is strictly convex and so h_j has at most one global minimum. By canceling the gradient, we have:

$$\nabla h_j(\theta) = 0 \Leftrightarrow Q\theta + c_j = 0 \Leftrightarrow \theta = -Q^{-1}c_j\,.$$

Finally,

$$\arg\min_{\Theta\in\mathbb{R}^{k\times k}} \operatorname{tr}\left(\frac{1}{2}\Theta^\top Q\Theta + C^\top\Theta\right) = -Q^{-1}C = I_k - Q^{-1}(Q+C)\,.$$

Moreover, according to the Karush–Kuhn–Tucker conditions, the minimizer θ^* of h_j under the constraint $\mathbf{1}_k^\top\theta = 1$ satisfies $\nabla h_j(\theta) = \lambda\mathbf{1}_k$ with $\lambda \in \mathbb{R}$. Thus, $Q\theta^* + c_j = \lambda\mathbf{1}_k$, hence,

$$\theta^* = \lambda Q^{-1}\mathbf{1}_k - Q^{-1}c_j\,.$$

Since $\mathbf{1}_k^\top\theta^* = 1$, we deduce that $\mathbf{1}_k^\top\theta^* = \lambda\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k - \mathbf{1}_k^\top Q^{-1}c_j = 1$ then,

$$\lambda = \frac{1 + \mathbf{1}_k^\top Q^{-1}c_j}{\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k} = \frac{\mathbf{1}_k^\top Q^{-1}(Qe_j + c_j)}{\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k}\,.$$

Finally, by noticing that $-Q^{-1}c_j = e_j - Q^{-1}(Qe_j + c_j)$

$$\theta^* = \frac{\mathbf{1}_k^\top Q^{-1}(Qe_j + c_j)}{\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k}Q^{-1}\mathbf{1}_k - Q^{-1}c_j = e_j - \left(Q^{-1} - \frac{Q^{-1}\mathbf{1}_k(Q^{-1}\mathbf{1}_k)^\top}{\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k}\right)(Qe_j + c_j)\,,$$

hence,

$$\underset{\Theta\in\mathbb{R}^{k\times k}\text{s.t. } \Theta^\top\mathbf{1}_k=\mathbf{1}_k}{\arg\min} \operatorname{tr}\left(\frac{1}{2}\Theta^\top Q\Theta + C^\top\Theta\right) = I_k - \left(Q^{-1} - \frac{Q^{-1}\mathbf{1}_k(Q^{-1}\mathbf{1}_k)^\top}{\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k}\right)(Q+C)\,.$$

Lemma 2 *Let $A \in \mathbb{R}^{n\times k}$ and $v \in \mathbb{R}^k \setminus \{0\}$.*

$$\arg\min_{\beta\in\mathbb{R}^n} \|A - \beta v^\top\|_F^2 = \frac{Av}{\|v\|_2^2}$$

Proof: $\|A - \beta v^\top\|_F^2 = \sum_{i=1}^n \|A_{i,\cdot} - \beta_i v\|_2^2 = \sum_{i=1}^n \|A_{i,\cdot}\|_2^2 - 2\beta_i\langle A_{i,\cdot}, v\rangle + \beta_i^2\|v\|_2^2$. Now, as the univariate quadratic function $\beta_i \in \mathbb{R} \mapsto \|A_{i,\cdot}\|_2^2 - 2\beta_i\langle A_{i,\cdot}, v\rangle + \beta_i^2\|v\|_2^2$ is minimized for $\beta_i = \langle A_{i,\cdot}, v\rangle/\|v\|_2^2$, we have $\arg\min_\beta \|A - \beta v^\top\|_F^2 = Av/\|v\|_2^2$.

B Mathematical proofs for NL-Ridge

In what follows, $X, Y \in \mathbb{R}^{n\times k}$. In each case, $Y_{i,j}$ follows a noise model which is centered on $X_{i,j}$ (*i.e.* $\mathbb{E}(Y_{i,j}) = X_{i,j}$) and variables $Y_{i,j}$ are supposed independent along each row. More precisely, three types of noise are studied:

- Gaussian noise: $Y_{i,j} \sim \mathcal{N}(X_{i,j}, V_{i,j})$ with $V \in (\mathbb{R}_*^+)^{n\times k}$ representing the noisemap, *i.e.* the variance per pixel. In particular, for homoscedastic Gaussian noise, $V = \sigma^2 \mathbf{1}_n \mathbf{1}_k^\top$ with $\sigma > 0$, that is $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$,
- Poisson noise: $Y_{i,j} \sim \mathcal{P}(X_{i,j})$,
- Mixed Poisson-Gaussian noise: $Y_{i,j} \sim a\mathcal{P}(X_{i,j}/a) + \mathcal{N}(0, b)$ with $(a, b) \in (\mathbb{R}_*^+)^2$.

The local denoiser in NL-Ridge is of the form $f_\Theta : Y \in \mathbb{R}^{n\times k} \mapsto Y\Theta$ with $\Theta \in \mathbb{R}^{k\times k}$ and the quadratic risk is defined as $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$.

B.1 Minimization of the quadratic risk

Lemma 3 (A closed-form expression for the quadratic risk) *Let $X, Y \in \mathbb{R}^{n\times k}$ and $V \in (\mathbb{R}_*^+)^{n\times k}$ such that the $Y_{i,j}$ are independent along each row, $\mathbb{E}(Y_{i,j}) = X_{i,j}$ and $\mathbb{V}(Y_{i,j}) = V_{i,j}$.*

$$\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \|X\Theta - X\|_F^2 + \operatorname{tr}\left(\Theta^\top \operatorname{diag}(V^\top \mathbf{1}_n)\Theta\right) .$$

Proof: By development of the squared Frobenius norm:

$$\|Y\Theta - X\|_F^2 = \|Y\Theta\|_F^2 + \|X\|_F^2 - 2\langle Y\Theta, X\rangle_F .$$

Now by linearity of expectation $\mathbb{E}\langle Y\Theta, X\rangle_F = \langle X\Theta, X\rangle_F$, and, as $Y_{i,j}$ are independent along each row, and as $\mathbb{E}(Y_{i,j}^2) = \mathbb{E}(Y_{i,j})^2 + \mathbb{V}(Y_{i,j}) = X_{i,j}^2 + V_{i,j}$, we have:

$$\begin{aligned}
\mathbb{E}\|Y\Theta\|_F^2 &= \mathbb{E}\left(\sum_{i=1}^n\sum_{j=1}^k\left(\sum_{l=1}^k Y_{i,l}\Theta_{l,j}\right)^2\right)\\
&= \sum_{i=1}^n\sum_{j=1}^k \mathbb{E}\left(\left(\sum_{l=1}^k Y_{i,l}\Theta_{l,j}\right)^2\right)\\
&= \sum_{i=1}^n\sum_{j=1}^k\left(\sum_{l=1}^k (X_{i,j}^2 + V_{i,j})\Theta_{l,j}^2 + 2\sum_{1\le l<l'\le k} X_{i,l}\Theta_{l,j}X_{i,l'}\Theta_{l',j}\right)\\
&= \sum_{i=1}^n\sum_{j=1}^k\left(\sum_{l=1}^k V_{i,j}\Theta_{l,j}^2 + \sum_{\substack{1\le l\le k\\ 1\le l'\le k}} X_{i,l}\Theta_{l,j}X_{i,l'}\Theta_{l',j}\right)\\
&= \sum_{i=1}^n\sum_{j=1}^k\sum_{l=1}^k V_{i,j}\Theta_{l,j}^2 + \sum_{i=1}^n\sum_{j=1}^k\left(\sum_{l=1}^k X_{i,l}\Theta_{l,j}\right)^2\\
&= \sum_{j=1}^k\sum_{l=1}^k \Theta_{l,j}\left(\sum_{i=1}^n V_{i,j}\right)\Theta_{l,j} + \|X\Theta\|_F^2\\
&= \operatorname{tr}\left(\Theta^\top \operatorname{diag}(V^\top\mathbf{1}_n)\Theta\right) + \|X\Theta\|_F^2 .
\end{aligned}$$

Hence,

$$\mathbb{E}\|Y\Theta - X\|_F^2 = \|X\Theta - X\|_F^2 + \operatorname{tr}\left(\Theta^\top \operatorname{diag}(V^\top \mathbf{1}_n)\Theta\right) .$$

Proposition 1 (Minimization of the quadratic risk) *Let $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$ the quadratic risk and $Q = X^\top X + D$ with D a diagonal matrix defined as:*

$$D = \begin{cases} n\sigma^2 I_k & \textit{(for homoscedastic Gaussian noise)}\\ \operatorname{diag}(V^\top\mathbf{1}_n) & \textit{(for heteroscedastic Gaussian noise)}\\ \operatorname{diag}(X^\top\mathbf{1}_n) & \textit{(for Poisson noise)}\\ \operatorname{diag}((aX+b)^\top\mathbf{1}_n) & \textit{(for mixed Poisson-Gaussian noise)}\end{cases} .$$

If Q is positive definite:

$$\begin{cases} \underset{\Theta\in\mathbb{R}^{k\times k}}{\arg\min} \quad \mathcal{R}_\Theta(X) = I_k - Q^{-1}D\,, \\ \underset{\substack{\Theta\in\mathbb{R}^{k\times k} \\ \text{s.t. } \Theta^\top \mathbf{1}_k = \mathbf{1}_k}}{\arg\min} \quad \mathcal{R}_\Theta(X) = I_k - \left[Q^{-1} - \frac{Q^{-1}\mathbf{1}_k(Q^{-1}\mathbf{1}_k)^\top}{\mathbf{1}_k^\top Q^{-1}\mathbf{1}_k}\right] D\,. \end{cases}$$

Proof: By Lemma 3,

$$\begin{aligned} \mathcal{R}_\Theta(X) &= \|X\Theta - X\|_F^2 + \operatorname{tr}\left(\Theta^\top \operatorname{diag}(V^\top \mathbf{1}_n)\Theta\right) \\ &= \operatorname{tr}\left((X\Theta - X)^\top (X\Theta - X)\right) + \operatorname{tr}\left(\Theta^\top \operatorname{diag}(V^\top \mathbf{1}_n)\Theta\right) \\ &= \operatorname{tr}\left(\Theta^\top X^\top X\Theta - 2X^\top X\Theta + X^\top X + \Theta^\top \operatorname{diag}(V^\top \mathbf{1}_n)\Theta\right) \\ &= \operatorname{tr}\left(\Theta^\top (X^\top X + \operatorname{diag}(V^\top \mathbf{1}_n))\Theta - 2X^\top X\Theta\right) + \operatorname{tr}\left(X^\top X\right) \end{aligned}$$

Lemma 1 allows to conclude.

B.2 Unbiased risk estimates (URE)

The three following propositions introduce unbiased risk estimates for $\mathcal{R}_\Theta(X)$ depending on the noise model assumed on Y, denoted $\mathrm{URE}_\Theta(Y)$ in a generic way.

Proposition 2 (Gaussian noise) *An unbiased estimate of the risk $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$ is:*

$$\mathrm{SURE}_\Theta(Y) = \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)\,,$$

with $D = \operatorname{diag}(V^\top \mathbf{1}_n)$. In particular, for homoscedastic Gaussian noise, $\mathrm{SURE}_\Theta(Y) = \|Y\Theta - Y\|_F^2 + 2n\sigma^2 \operatorname{tr}(\Theta) - nk\sigma^2$.

Proof: For $n = 1$, all components of Y are independent and generalized Stein's unbiased risk estimate (SURE) [51] is given by:

$$\mathrm{SURE}_\Theta(Y) = \|f_\Theta(Y) - Y\|_F^2 + 2\operatorname{tr}(\operatorname{diag}(V^\top \mathbf{1}_n)\Theta) - \operatorname{tr}(\operatorname{diag}(V^\top \mathbf{1}_n))\,.$$

For $n \geq 1$,

$$\mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \sum_{i=1}^n \mathbb{E}\|Y_{i,\cdot}\Theta - X_{i,\cdot}\|_F^2 = \sum_{i=1}^n \mathbb{E}\left(\mathrm{SURE}_\Theta(Y_{i,\cdot})\right) = \mathbb{E}\left(\sum_{i=1}^n \mathrm{SURE}_\Theta(Y_{i,\cdot})\right)\,,$$

hence,

$$\begin{aligned} \mathrm{SURE}_\Theta(Y) = \sum_{i=1}^n \mathrm{SURE}_\Theta(Y_{i,\cdot}) &= \sum_{i=1}^n \|Y_{i,\cdot}\Theta - Y_{i,\cdot}\|_F^2 + 2\operatorname{tr}(\operatorname{diag}(V_{i,\cdot}^\top)\Theta) - \operatorname{tr}(\operatorname{diag}(V_{i,\cdot}^\top)) \\ &= \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)\,. \end{aligned}$$

Proposition 3 (Poisson noise) *An unbiased estimate of the risk $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$ is:*

$$\mathrm{PURE}_\Theta(Y) = \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)\,,$$

with $D = \operatorname{diag}(Y^\top \mathbf{1}_n)$.

Proof: For $n = 1$, all components of Y are independent and Poisson unbiased risk estimate (PURE) [37, 25] is given by:

$$\mathrm{PURE}_\Theta(Y) = \|f_\Theta(Y)\|_F^2 + \|Y\|_F^2 - Y\mathbf{1}_k - 2\langle f_\Theta^{[-1]}(Y), Y\rangle_F$$

with $f_\Theta^{[-1]}$ is such that $f_\Theta^{[-1]i}(Y) = f_\Theta^i(Y - e_i)$. We have:

$$\begin{aligned} \langle f_\Theta^{[-1]}(Y), Y\rangle_F &= \sum_{j=1}^k \left(\sum_{l=1}^k (Y_{1,l} - \delta_{l,j})\Theta_{l,j}\right) Y_{1,j} \\ &= \sum_{j=1}^k \left(\sum_{l=1}^k Y_{1,l}\Theta_{l,j}\right) Y_{1,j} - \sum_{j=1}^k \left(\sum_{l=1}^k \delta_{l,j}\Theta_{l,j}\right) Y_{1,j} \end{aligned}$$

$$= \langle Y\Theta, Y\rangle_F - \sum_{j=1}^{k} \Theta_{j,j} Y_{1,j} = \langle Y\Theta, Y\rangle_F - Y \operatorname{diag}(\Theta)\,.$$

So finally, $\mathrm{PURE}_\Theta(Y) = \|Y\Theta - Y\|_F^2 - Y\mathbf{1}_k + 2Y\operatorname{diag}(\Theta) = \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)$.

For $n \geq 1$,

$$\mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \sum_{i=1}^{n} \mathbb{E}\|Y_{i,\cdot}\Theta - X_{i,\cdot}\|_F^2 = \sum_{i=1}^{n} \mathbb{E}(\mathrm{PURE}_\Theta(Y_{i,\cdot})) = \mathbb{E}\left(\sum_{i=1}^{n} \mathrm{PURE}_\Theta(Y_{i,\cdot})\right),$$

hence,

$$\begin{aligned}\mathrm{PURE}_\Theta(Y) = \sum_{i=1}^{n} \mathrm{PURE}_\Theta(Y_{i,\cdot}) &= \sum_{i=1}^{n} \|Y_{i,\cdot}\Theta - Y_{i,\cdot}\|_F^2 + 2\operatorname{tr}(\operatorname{diag}(Y_{i,\cdot}^\top)\Theta) - \operatorname{tr}(\operatorname{diag}(Y_{i,\cdot}^\top)) \\ &= \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)\,.\end{aligned}$$

Proposition 4 (Mixed Poisson-Gaussian noise) *An unbiased estimate of the risk $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$ is:*

$$\text{PG-URE}_\Theta(Y) = \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)\,,$$

with $D = \operatorname{diag}((aY+b)^\top \mathbf{1}_n)$.

Proof: For $n = 1$, all components of Y are independent and the Poisson-Gaussian unbiased risk estimate (PG-URE) [25] is given by:

$$\text{PG-URE}_\Theta(Y) = \|f_\Theta(Y)\|_F^2 + \|Y\|_F^2 - 2\langle f_\Theta^{[-a]}(Y), Y\rangle_F - (aY+b)\mathbf{1}_k + 2b\operatorname{div}(f_\Theta^{[-a]})(Y)$$

with $f_\Theta^{[-a]}$ is such that $f_\Theta^{[-a]i}(Y) = f_\Theta^i(Y - ae_i)$.

$$\begin{aligned}\langle f_\Theta^{[-a]}(Y), Y\rangle_F &= \sum_{j=1}^{k}\left(\sum_{l=1}^{k}(Y_{1,l} - a\delta_{l,j})\Theta_{l,j}\right) Y_{1,j} \\ &= \sum_{j=1}^{k}\left(\sum_{l=1}^{k} Y_{1,l}\Theta_{l,j}\right) Y_{1,j} - \sum_{j=1}^{k}\left(\sum_{l=1}^{k} a\delta_{l,j}\Theta_{l,j}\right) Y_{1,j} \\ &= \langle Y\Theta, Y\rangle_F - a\sum_{j=1}^{k}\Theta_{j,j}Y_{1,j} = \langle Y\Theta, Y\rangle_F - aY\operatorname{diag}(\Theta)\,.\end{aligned}$$

and $\operatorname{div}(f_\Theta^{[-a]})(Y) = \sum_{j=1}^{k} \dfrac{\partial f_\Theta^{[-a]j}}{\partial y_j}(Y) = \sum_{j=1}^{k} \dfrac{\partial}{\partial y_j} f_\Theta^j(Y - ae_j) = \sum_{j=1}^{k} \Theta_{j,j} = \mathbf{1}_k^\top \operatorname{diag}(\Theta)\,.$

So finally, $\text{PG-URE}_\Theta(Y) = \|Y\Theta - Y\|_F^2 - (aY+b)\mathbf{1}_k + 2(aY+b)\operatorname{diag}(\Theta) = \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)$.

For $n \geq 1$,

$$\mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \sum_{i=1}^{n} \mathbb{E}\|Y_{i,\cdot}\Theta - X_{i,\cdot}\|_F^2 = \sum_{i=1}^{n} \mathbb{E}(\text{PG-URE}_\Theta(Y_{i,\cdot})) = \mathbb{E}\left(\sum_{i=1}^{n} \text{PG-URE}_\Theta(Y_{i,\cdot})\right),$$

hence,

$$\begin{aligned}\text{PG-URE}_\Theta(Y) &= \sum_{i=1}^{n} \text{PG-URE}_\Theta(Y_{i,\cdot}) \\ &= \sum_{i=1}^{n} \|Y_{i,\cdot}\Theta - Y_{i,\cdot}\|_F^2 + 2\operatorname{tr}(\operatorname{diag}((aY_{i,\cdot}+b)^\top)\Theta) - \operatorname{tr}(\operatorname{diag}((aY_{i,\cdot}+b)^\top)) \\ &= \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D)\,.\end{aligned}$$

In the following, we denote $\mathrm{URE}_\Theta(Y)$ either the $\mathrm{SURE}_\Theta(Y)$, $\mathrm{PURE}_\Theta(Y)$ or $\text{PG-URE}_\Theta(Y)$ estimate, depending on the noise model assumed on Y.

Proposition 5 (Minimization of the URE) *Let $Q = Y^\top Y$ and D a positive diagonal one defined as:*

$$D = \begin{cases} n\sigma^2 I_k & \textit{(for homoscedastic Gaussian noise)} \\ \operatorname{diag}(V^\top \mathbf{1}_n) & \textit{(for heteroscedastic Gaussian noise)} \\ \operatorname{diag}(Y^\top \mathbf{1}_n) & \textit{(for Poisson noise)} \\ \operatorname{diag}((aY+b)^\top \mathbf{1}_n) & \textit{(for mixed Poisson-Gaussian noise)} \end{cases}.$$

If Q is definite positive,

$$\begin{cases} \underset{\Theta \in \mathbb{R}^{k\times k}}{\arg\min} \quad \mathrm{URE}_\Theta(Y) = I_k - Q^{-1}D\,, \\ \underset{\substack{\Theta \in \mathbb{R}^{k\times k} \\ \textit{s.t. } \Theta^\top \mathbf{1}_k = \mathbf{1}_k}}{\arg\min} \quad \mathrm{URE}_\Theta(Y) = I_k - \left[Q^{-1} - \dfrac{Q^{-1}\mathbf{1}_k (Q^{-1}\mathbf{1}_k)^\top}{\langle Q^{-1}, \mathbf{1}_k \mathbf{1}_k^\top \rangle_F}\right] D\,. \end{cases}$$

Proof: Using Proposition 2, 3 and 4,

$$\begin{aligned} \mathrm{URE}_\Theta(Y) &= \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D) \\ &= \operatorname{tr}\left(\Theta^\top Y^\top Y \Theta + 2(D - Y^\top Y)\Theta\right) + \text{const} \end{aligned}$$

Lemma 1 allows to conclude.

Proposition 6 (URE for a noisier risk and its minimization) *Let $\alpha > 0$ and $W \in \mathbb{R}^{n\times k}$ with $W_{i,j} \sim \mathcal{N}(0,1)$ independent along each row. We define the noisier risk as $\mathcal{R}_\Theta^{Nr,\alpha}(X) = \mathbb{E}\|f_\Theta(Y + \alpha W) - X\|_F^2$. An unbiased estimate of the noisier risk $\mathcal{R}_\Theta^{Nr,\alpha}(X)$ is:*

$$\mathrm{URE}_\Theta^{Nr,\alpha}(Y) = \mathrm{URE}_\Theta(Y) + n\alpha^2 \|\Theta\|_F^2\,.$$

and its minimization yields:

$$\begin{cases} \underset{\Theta \in \mathbb{R}^{k\times k}}{\arg\min} \quad \mathrm{URE}_\Theta^{Nr,\alpha}(Y) = I_k - Q^{-1}(D + n\alpha^2 I_k)\,, \\ \underset{\substack{\Theta \in \mathbb{R}^{k\times k} \\ \textit{s.t. } \Theta^\top \mathbf{1}_k = \mathbf{1}_k}}{\arg\min} \quad \mathrm{URE}_\Theta^{Nr,\alpha}(Y) = I_k - \left[Q^{-1} - \dfrac{Q^{-1}\mathbf{1}_k (Q^{-1}\mathbf{1}_k)^\top}{\mathbf{1}_k^\top Q^{-1} \mathbf{1}_k}\right] (D + n\alpha^2 I_k)\,, \end{cases}$$

with $Q = Y^\top Y + n\alpha^2 I_k$ a symmetric definite-positive matrix and D a diagonal one defined as:

$$D = \begin{cases} n\sigma^2 I_k & \textit{(for homoscedastic Gaussian noise)} \\ \operatorname{diag}(V^\top \mathbf{1}_n) & \textit{(for heteroscedastic Gaussian noise)} \\ \operatorname{diag}(Y^\top \mathbf{1}_n) & \textit{(for Poisson noise)} \\ \operatorname{diag}((aY+b)^\top \mathbf{1}_n) & \textit{(for mixed Poisson-Gaussian noise)} \end{cases}.$$

Proof: As f_Θ is a linear function and Y and W are independent:

$$\begin{aligned} \mathcal{R}_\Theta^{\mathrm{Nr},\alpha}(X) &= \mathbb{E}\|f_\Theta(Y + \alpha W) - X\|_F^2 \\ &= \mathbb{E}\left[\|f_\Theta(Y) - X\|_F^2 + \alpha^2 \|f_\Theta(W)\|_F^2 + 2\langle f_\Theta(Y) - X, \alpha f_\Theta(W)\rangle_F\right] \\ &= \mathcal{R}_\Theta(X) + \alpha^2 \mathbb{E}\|f_\Theta(W)\|_F^2 \\ &= \mathbb{E}\left[\mathrm{URE}_\Theta(Y)\right] + \alpha^2 \mathbb{E}\|W\Theta\|_F^2 \end{aligned}$$

with, as the $W_{i,j}$ are independent along each row:

$$\mathbb{E}\|W\Theta\|_F^2 = \sum_{i=1}^n \sum_{j=1}^k \mathbb{E}\left(\left(\sum_{l=1}^k W_{i,l}\Theta_{l,j}\right)^2\right) = \sum_{i=1}^n \sum_{j=1}^k \sum_{l=1}^k \Theta_{l,j}^2 = n\|\Theta\|_F^2\,.$$

Now, using Proposition 2, 3 and 4,

$$\begin{aligned} \mathrm{URE}_\Theta(Y) &= \|Y\Theta - Y\|_F^2 + 2\operatorname{tr}(D\Theta) - \operatorname{tr}(D) + n\alpha^2\|\Theta\|_F^2 \\ &= \operatorname{tr}\left(\Theta^\top (Y^\top Y + n\alpha^2 I_k)\Theta + 2(D - Y^\top Y)\Theta\right) + \text{const} \end{aligned}$$

Lemma 1 allows to conclude.

C Mathematical proofs for NL-Bayes

In what follows, $X, Y \in \mathbb{R}^{n\times k}$, with $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$ independent along each column. The local denoiser in NL-Bayes is of the form $f_{\Theta,\beta} : Y \in \mathbb{R}^{n\times k} \mapsto \Theta Y + \beta \mathbf{1}_k^\top$ with $\Theta \in \mathbb{R}^{n\times n}$ and $\beta \in \mathbb{R}^n$. The quadratic risk is defined as $\mathcal{R}_{\Theta,\beta}(X) = \mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2$. We denote by $\mu_Z \in \mathbb{R}^k$ and $C_Z \in \mathbb{R}^{n\times n}$ the empirical mean and covariance matrix of a group of patches $Z \in \mathbb{R}^{n\times k}$, that is

$$\mu_Z = \frac{1}{k} Z \mathbf{1}_k \quad \text{and} \quad C_Z = \frac{1}{k}(Z - \mu_Z \mathbf{1}_k^\top)(Z - \mu_Z \mathbf{1}_k^\top)^\top .$$

C.1 Minimization of the quadratic risk

Lemma 4 (A closed-form expression for the quadratic risk)

$$\mathcal{R}_{\Theta,\beta}(X) = \mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2 = \|\Theta X - X + \beta \mathbf{1}_k^\top\|_F^2 + k\sigma^2 \|\Theta\|_F^2 .$$

Proof: By development of the Frobenius norm and using Lemma 3:

$$\begin{aligned}
\mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2 &= \mathbb{E}\left[\|\Theta Y - X\|_F^2 + \|\beta \mathbf{1}_k^\top\|_F^2 + 2\langle \Theta Y - X, \beta \mathbf{1}_k^\top \rangle_F\right] \\
&= \mathbb{E}\|Y^\top \Theta^\top - X^\top\|_F^2 + \|\beta \mathbf{1}_k^\top\|_F^2 + 2\mathbb{E}\langle \Theta Y - X, \beta \mathbf{1}_k^\top \rangle_F \\
&= \|X^\top \Theta^\top - X^\top\|_F^2 + k\sigma^2 \|\Theta^\top\|_F^2 + \|\beta \mathbf{1}_k^\top\|_F^2 + 2\langle \Theta X - X, \beta \mathbf{1}_k^\top \rangle_F \\
&= \|X\Theta - X + \beta \mathbf{1}_k^\top\|_F^2 + k\sigma^2 \|\Theta\|_F^2
\end{aligned}$$

Proposition 7 (Minimization of the quadratic risk)

$$\arg \min_{\substack{\Theta \in \mathbb{R}^{n\times n} \\ \beta \in \mathbb{R}^n}} \mathcal{R}_{\Theta,\beta}(X) = \hat{\Theta}, \hat{\beta}$$

with $\hat{\Theta} = C_X (C_X + \sigma^2 I_n)^{-1}$ *and* $\hat{\beta} = (I_n - \hat{\Theta})\mu_X$.

Proof: According to Lemma 4, $\mathcal{R}_{\Theta,\beta}(X) = \mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2 = \|\Theta X - X + \beta \mathbf{1}_k^\top\|_F^2 + k\sigma^2\|\Theta\|_F^2$. For Θ fixed and using Lemma 2, it is minimized for $\beta = -(\Theta X - X)\mathbf{1}_k/k = (I_n - \Theta)\mu_X$.

Injecting it in the expression of the risk:

$$\|(X - \mu_X \mathbf{1}_k^\top)^\top \Theta^\top - (X - \mu_X \mathbf{1}_k^\top)^\top\|_F^2 + k\sigma^2 \|\Theta^\top\|_F^2$$

This quantity is minimal, using Lemma 1, for

$$\hat{\Theta}^\top = I_n - k\sigma^2 ((X - \mu_X \mathbf{1}_k^\top)(X - \mu_X \mathbf{1}_k^\top)^\top + k\sigma^2 I_n)^{-1} = I_n - k\sigma^2 (kC_X + k\sigma^2 I_n)^{-1} ,$$

i.e.

$$\hat{\Theta} = I_n - \sigma^2 (C_X + \sigma^2 I_n)^{-1} = C_X (C_X + \sigma^2 I_n)^{-1} .$$

C.2 Unbiased risk estimate (URE)

Proposition 8 (Gaussian noise) *An unbiased estimate of the risk* $\mathcal{R}_{\Theta,\beta}(X) = \mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2$ *is:*

$$\mathrm{SURE}_{\Theta,\beta}(Y) = \|\Theta Y - Y + \beta \mathbf{1}_k^\top\|_F^2 + 2k\sigma^2 \operatorname{tr}(\Theta) - nk\sigma^2 .$$

Proof: For $k = 1$, all components of Y are independent and Stein's unbiased risk estimate (SURE) [51] is given by:

$$\mathrm{SURE}_{\Theta,\beta}(Y) = -n\sigma^2 + \|f_{\Theta,\beta}(Y) - Y\|_F^2 + 2\sigma^2 \operatorname{div} f_{\Theta,\beta}(Y)$$

with $\operatorname{div} f_{\Theta,\beta}(Y) = \displaystyle\sum_{i=1}^n \frac{\partial f^i_{\Theta,\beta}}{\partial y_i}(Y) = \sum_{i=1}^n \frac{\partial}{\partial y_i} \sum_{l=1}^n \Theta_{i,l} Y_{l,1} = \sum_{i=1}^n \Theta_{i,i} = \operatorname{tr}(\Theta).$

For $k \geq 1$,

$$\mathbb{E}\|f_{\Theta,\beta}(Y) - X\|_F^2 = \sum_{j=1}^m \mathbb{E}\|\Theta Y_{\cdot,j} + \beta - X_{\cdot,j}\|_F^2 = \sum_{j=1}^k \mathbb{E}(\mathrm{SURE}_{\Theta,\beta}(Y_{\cdot,j})) = \mathbb{E}\sum_{j=1}^k \mathrm{SURE}_{\Theta,\beta}(Y_{\cdot,j}),$$

hence,

$$\mathrm{SURE}_{\Theta,\beta}(Y) = \sum_{j=1}^k \mathrm{SURE}_{\Theta,\beta}(Y_{\cdot,j}) = \|\Theta Y - Y + \beta \mathbf{1}_k^\top\|_F^2 + 2k\sigma^2 \operatorname{tr}(\Theta) - nk\sigma^2 .$$

Proposition 9 (Minimization of the URE)

$$\arg\min_{\Theta,\beta} \mathrm{SURE}_{\Theta,\beta}(Y) = \hat{\Theta}, \hat{\beta}$$

with $\hat{\Theta}, \hat{\beta} = (C_Y - \sigma^2 I_n)C_Y^{-1}, (I_n - \hat{\Theta})\mu_Y$.

Proof: For Θ fixed and using Lemma 2, $\mathrm{SURE}_{\Theta,\beta}(Y)$ is minimal for $\beta = -(\Theta Y - Y)\mathbf{1}_k/k = (I_n - \Theta)\mu_Y$. Injecting it in the expression of SURE:

$$\|(Y - \mu_Y \mathbf{1}_k^\top)^\top \Theta^\top - (Y - \mu_Y \mathbf{1}_k^\top)^\top\|_F^2 + 2k\sigma^2 \operatorname{tr}(\Theta) - nk\sigma^2 \,.$$

This quantity is minimal, using Lemma 1, for

$$\hat{\Theta}^\top = I_n - k\sigma^2((Y - \mu_Y \mathbf{1}_k^\top)(Y - \mu_Y \mathbf{1}_k^\top)^\top)^{-1} = I_n - \sigma^2 C_Y^{-1} \,,$$

i.e.

$$\hat{\Theta} = (C_Y - \sigma^2 I_n)C_Y^{-1} \,.$$

D Mathematical proofs for BM3D

In what follows, $X, Y \in \mathbb{R}^{n\times k}$, with $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$ independent. The local denoiser in BM3D is of the form $f_\Theta : Y \mapsto P^{-1}(\Theta \odot (PYQ))Q^{-1}$ with $\Theta \in \mathbb{R}^{n\times k}$ and $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{k\times k}$ are two orthogonal matrices (*i.e.* $PP^\top = I_n$ and $QQ^\top = I_k$). The quadratic risk is defined as $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$.

D.1 Minimization of the quadratic risk

Lemma 5 (A closed-form expression for the quadratic risk)

$$\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top) \odot PXQ\|_F^2 + \sigma^2\|\Theta\|_F^2 \,.$$

Proof: Let $W = Y - X$. We have $W_{i,j} \sim \mathcal{N}(0, \sigma^2)$. As P and Q are orthogonal matrices, they preserve the ℓ_2 norm:

$$\begin{aligned}\|f_\Theta(Y) - X\|_F^2 &= \|\Theta \odot (PYQ) - PXQ\|_F^2 = \|\Theta \odot (PXQ) + \Theta \odot (PWQ) - PXQ\|_F^2 \\ &= \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top) \odot PXQ\|_F^2 + \|\Theta \odot (PWQ)\|_F^2 + 2\langle(\Theta - \mathbf{1}_n\mathbf{1}_k^\top) \odot PXQ, \Theta \odot (PWQ)\rangle_F \,.\end{aligned}$$

Now computing the expected value for each term yields:

$$\mathbb{E}\langle(\Theta - \mathbf{1}_n\mathbf{1}_k^\top) \odot (PXQ), \Theta \odot (PWQ)\rangle_F = 0$$

and

$$\begin{aligned}\mathbb{E}\|\Theta \odot (PWQ)\|_F^2 = \sum_{i=1}^n\sum_{j=1}^k \mathbb{E}[(\Theta_{i,j}(PWQ)_{i,j})^2] &= \sum_{i=1}^n\sum_{j=1}^k \mathbb{V}[\Theta_{i,j}(PWQ)_{i,j}] \\ &= \sum_{i=1}^n\sum_{j=1}^k \Theta_{i,j}^2 \mathbb{V}[(PWQ)_{i,j}]\end{aligned}$$

with, as $W_{i,j}$ are independent and P and Q are orthogonal,

$$\begin{aligned}\mathbb{V}[(PWQ)_{i,j}] = \mathbb{V}\left(\sum_{l=1}^k \left(\sum_{l'=1}^n P_{i,l'}W_{l',l}\right) Q_{l,j}\right) &= \sum_{l=1}^k Q_{l,j}^2 \mathbb{V}\left(\sum_{l'=1}^n P_{i,l'}W_{l',l}\right) \\ &= \sum_{l=1}^k Q_{l,j}^2 \sum_{l'=1}^n P_{i,l'}^2 \mathbb{V}\left(W_{l',l}\right) = \sigma^2 \,.\end{aligned}$$

Finally, $\mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top) \odot PXQ\|_F^2 + \sigma^2\|\Theta\|_F^2$.

Proposition 10 (Minimization of the quadratic risk)

$$\arg\min_{\Theta\in\mathbb{R}^{n\times k}} \mathcal{R}_\Theta(X) = \frac{(PXQ)^{\odot 2}}{\sigma^2 + (PXQ)^{\odot 2}}\,.$$

Proof: According to Lemma 5, $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top)\odot PXQ\|_F^2 + \sigma^2\|\Theta\|_F^2$.

Let $\alpha \in \mathbb{R}$. The minimum of $x \mapsto \alpha^2(x-1)^2 + \sigma^2 x^2$ is obtained for $x = \dfrac{\alpha^2}{\sigma^2+\alpha^2}$. Finally,

$$\arg\min_{\Theta}\ \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top)\odot PXQ\|_F^2 + \sigma^2\|\Theta\|_F^2 = \frac{(PXQ)^{\odot 2}}{\sigma^2 + (PXQ)^{\odot 2}}.$$

D.2 Unbiased risk estimate (URE)

Proposition 11 (Gaussian noise) *An unbiased estimate of the risk $\mathcal{R}_\Theta(X) = \mathbb{E}\|f_\Theta(Y) - X\|_F^2$ is:*

$$\mathrm{SURE}_\Theta(Y) = \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top)\odot PYQ\|_F^2 + 2\sigma^2\langle\Theta, \mathbf{1}_n\mathbf{1}_k^\top\rangle_F - nk\sigma^2\,.$$

Proof: Let $W = Y - X$. By development of the squared Frobenius norm, $\|f_\Theta(Y) - Y\|_F^2 = \|f_\Theta(Y) - X\|_F^2 + \|W\|_F^2 - 2\langle f_\Theta(Y) - X, W\rangle_F$. As $P^{-1} = P^\top$ and $Q^{-1} = Q^\top$:

$$\begin{aligned}\langle f_\Theta(Y), W\rangle_F &= \langle P^{-1}(\Theta\odot(PYQ))Q^{-1}, W\rangle_F\\ &= \langle\Theta\odot(PYQ), PWQ\rangle_F\\ &= \langle\Theta\odot(PXQ), PWQ\rangle_F + \langle\Theta\odot(PWQ), PWQ\rangle_F\,.\end{aligned}$$

Now computing the expected value for each term yields:

$$\mathbb{E}\langle\Theta\odot(PXQ), PWQ\rangle_F = 0,\quad \mathbb{E}\|W\|_F^2 = nk\sigma^2,\quad \mathbb{E}\langle X, W\rangle_F = 0$$

and

$$\mathbb{E}\langle\Theta\odot(PWQ), PWQ\rangle_F = \sigma^2\langle\mathbf{1}_n\mathbf{1}_k^\top, \Theta\rangle_F.$$

Indeed, as the $W_{i,j}$ are independent and P and Q are orthogonal matrices, and according to the proof of Lemma 5:

$$\mathbb{E}[\Theta_{i,j}(PWQ)_{i,j}^2] = \Theta_{i,j}\mathbb{E}[(PWQ)_{i,j}^2] = \Theta_{i,j}\left(\underbrace{\mathbb{E}[(PWQ)_{i,j}]^2}_{=0} + \underbrace{\mathbb{V}[(PWQ)_{i,j}]}_{=\sigma^2}\right) = \sigma^2\Theta_{i,j}\,.$$

Finally, we get $\mathbb{E}\|f_\Theta(Y) - X\|_F^2 = \mathbb{E}\left[\|f_\Theta(Y) - Y\|_F^2 + 2\sigma^2\langle\mathbf{1}_n\mathbf{1}_k^\top, \Theta\rangle_F - nk\sigma^2\right]$ with $\|f_\Theta(Y) - Y\|_F^2 = \|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top)\odot PYQ\|_F^2$.

Proposition 12 (Minimization of the URE)

$$\arg\min_{\Theta\in\mathbb{R}^{n\times k}} \mathrm{SURE}_\Theta(Y) = \mathbf{1}_n\mathbf{1}_k^\top - \frac{\sigma^2}{(PYQ)^{\odot 2}}\,,$$

and

$$\arg\min_{\Theta\in\{0,1\}^{n\times k}} \mathrm{SURE}_\Theta(Y) = \mathbb{1}_{\mathbb{R}\setminus[-\sqrt{2}\sigma,\sqrt{2}\sigma]}(PYQ)\,.$$

Proof:

$$\|(\Theta - \mathbf{1}_n\mathbf{1}_k^\top)\odot PYQ\|_F^2 + 2\sigma^2\langle\Theta, \mathbf{1}_n\mathbf{1}_k^\top\rangle_F = \sum_{i=1}^n\sum_{j=1}^n((PYQ)_{i,j}(\Theta_{i,j} - 1))^2 + 2\sigma^2\Theta_{i,j}$$

Let $\alpha \in \mathbb{R}^*$. The minimum of $x \in \mathbb{R} \mapsto (\alpha(x-1))^2 + 2\sigma^2 x$ is obtained for $x_{\min,1} = 1 - \dfrac{\sigma^2}{\alpha^2}$. Hence,

$$\arg\min_{\Theta\in\mathbb{R}^{n\times k}} \mathrm{SURE}_\Theta(Y) = \mathbf{1}_n\mathbf{1}_k^\top - \frac{\sigma^2}{(PYQ)^{\odot 2}}\,.$$

The minimum of $x \in \{0,1\} \mapsto (\alpha(x-1))^2 + 2\sigma^2 x$ is obtained for $x_{\min,2} = \mathbb{1}_{\mathbb{R}\setminus[-\sqrt{2}\sigma,\sqrt{2}\sigma]}(\alpha)$. Hence,

$$\arg\min_{\Theta\in\{0,1\}^{n\times k}} \mathrm{SURE}_\Theta(Y) = \mathbb{1}_{\mathbb{R}\setminus[-\sqrt{2}\sigma,\sqrt{2}\sigma]}(PYQ)\,.$$

Acknowledgments

This work was supported by Bpifrance agency (funding) through the LiChIE contract. Computations were performed on the Inria Rennes computing grid facilities partly funded by France-BioImaging infrastructure (French National Research Agency - ANR-10-INBS-04-07, "Investments for the future").

We would like to thank R. Fraisse (Airbus) for fruitful discussions.

References

[1] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *International Conference on Machine Learning (ICML)*, volume 97, pages 524–533, 2019.

[2] A. Benazza-Benyahia and J.-C. Pesquet. An extended SURE approach for multicomponent image denoising. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages ii–945, 2004.

[3] Thierry Blu and Florian Luisier. The SURE-LET approach to image denoising. *IEEE Transactions on Image Processing*, 16(11):2778–2786, 2007.

[4] Jérôme Boulanger, Charles Kervrann, Patrick Bouthemy, Peter Elbau, Jean-Baptiste Sibarita, and Jean Salamero. Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. *IEEE Transactions on Medical Imaging*, 29(2):442–454, 2010.

[5] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[6] Harold C. Burger, Christian J. Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399, 2012.

[7] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017.

[8] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[9] Weisheng Dong, Guangming Shi, and Xin Li. Nonlocal image restoration with bilateral variance estimation: A low-rank approach. *IEEE Transactions on Image Processing*, 22(2):700–711, 2013.

[10] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013.

[11] Vincent Duval, Jean-François Aujol, and Yann Gousseau. A bias-variance approach for the nonlocal means. *SIAM Journal on Imaging Sciences*, 4(2):760–788, 2011.

[12] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[13] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2862–2869, 2014.

[14] Sébastien Herbreteau, Emmanuel Moebel, and Charles Kervrann. Normalization-equivariant neural networks with application to image denoising. *arXiv preprint arXiv:2306.05037*, 2023.

[15] Sébastien Herbreteau and Charles Kervrann. DCT2net: an interpretable shallow CNN for image denoising. *IEEE Transactions on Image Processing*, 31:4292–4305, 2022.

[16] Sébastien Herbreteau and Charles Kervrann. Towards a unified view of unsupervised non-local methods for image denoising: the NL-Ridge approach. In *IEEE International Conference on Image Processing (ICIP)*, pages 3376–3380, 2022.

[17] Haijuan Hu, Jacques Froment, and Quansheng Liu. A note on patch-based low-rank minimization for fast image denoising. *Journal of Visual Communication and Image Representation*, 50:100–110, 2018.

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015.

[19] Qiyu Jin, Ion Grama, Charles Kervrann, and Quansheng Liu. Nonlocal means and optimal weights for noise removal. *SIAM Journal on Imaging Sciences*, 10(4):1878–1920, 2017.

[20] Yeonsik Jo, Se Young Chun, and Jonghyun Choi. Rethinking Deep Image Prior for denoising. In *International Conference on Computer Vision (ICCV)*, pages 5087–5096, 2021.

[21] Charles Kervrann. PEWA: Patch-based exponentially weighted aggregation for image denoising. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27, 2014.

[22] Aravindh Krishnamoorthy and Deepak Menon. Matrix inversion using Cholesky decomposition. In *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 70–72, 2013.

[23] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void - Learning denoising from single noisy images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2124–2132, 2019.

[24] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[25] Yoann Le Montagner, Elsa D. Angelini, and Jean-Christophe Olivo-Marin. An unbiased risk estimator for image denoising in the presence of mixed Poisson–Gaussian noise. *IEEE Transactions on Image Processing*, 23(3):1255–1268, 2014.

[26] M. Lebrun, A. Buades, and J. M. Morel. A nonlocal Bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

[27] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. Implementation of the "Non-Local Bayes" (NL-Bayes) image denoising algorithm. *Image Processing On Line*, 3:1–42, 2013.

[28] Marc Lebrun, Miguel Colom, A. Buades, and Jean-Michel Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21(4):475 – 576, 2012.

[29] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *International Conference on Machine Learning (ICML)*, volume 80, pages 2965–2974, 2018.

[30] Victor Lempitsky, Andrea Vedaldi, and Dmitry Ulyanov. Deep image prior. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9446–9454, 2018.

[31] Jason Lequyer, Reuben Philip, Amit Sharma, Wen-Hsin Hsu, and Laurence Pelletier. A fast blind zero-shot denoiser. *Nature Machine Intelligence*, 4(11):953–963, 2022.

[32] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using Swin Transformer. In *International Conference on Computer Vision Workshops (ICCVW)*, pages 1833–1844, 2021.

[33] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[34] Haosen Liu, Xuan Liu, Jiangbo Lu, and Shan Tan. Self-supervised image prior learning with GMM from a single noisy image. In *International Conference on Computer Vision (ICCV)*, pages 2825–2834, 2021.

[35] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-CNN for image restoration. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 773–782, 2018.

[36] Florian Luisier, Thierry Blu, and Michael Unser. A new SURE approach to image denoising: interscale orthonormal wavelet thresholding. *IEEE Transactions on Image Processing*, 16(3):593–606, 2007.

[37] Florian Luisier, Cédric Vonesch, Thierry Blu, and Michael Unser. Fast interscale wavelet denoising of Poisson-corrupted images. *Signal Processing*, 90:415–427, 2010.

[38] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *International Conference on Computer Vision (ICCV)*, pages 2272–2279, 2009.

[39] Youssef Mansour and Reinhard Heckel. Zero-Shot Noise2Noise: Efficient image denoising without any data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14018–14027, 2023.

[40] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, 2016.

[41] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 416–423 vol.2, 2001.

[42] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2Noise: Learning to denoise from unpaired noisy data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12061–12069, 2020.

[43] Ymir Mäkinen, Lucio Azzari, and Alessandro Foi. Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching. *IEEE Transactions on Image Processing*, 29:8339–8354, 2020.

[44] Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer, 1999.

[45] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-Recorrupted: Unsupervised deep learning for image denoising. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2043–2052, 2021.

[46] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[47] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2750–2759, 2017.

[48] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2Self with dropout: Learning self-supervised denoising from single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1887–1895, 2020.

[49] J. Salmon and Y. Strozecki. From patches to pixels in non-local methods: Weighted-average reprojection. In *IEEE International Conference on Image Processing (ICIP)*, pages 1929–1932, 2010.

[50] Jean-Luc Starck, Fionn D Murtagh, and Albert Bijaoui. *Image Processing and Data Analysis: the multiscale approach*. Cambridge University Press, 1998.

[51] Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151, 1981.

[52] Gregory Vaksman, Michael Elad, and Peyman Milanfar. LIDIA: Lightweight learned image denoising with instance adaptation. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2220–2229, 2020.

[53] Dimitri Van De Ville and Michel Kocher. SURE-based Non-Local Means. *IEEE Signal Processing Letters*, 16(11):973–976, 2009.

[54] Yi-Qing Wang and Jean-Michel Morel. SURE guided Gaussian mixture image denoising. *SIAM Journal on Imaging Sciences*, 6(2):999–1034, 2013.

[55] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *European Conference on Computer Vision (ECCV)*, pages 20–36, 2018.

[56] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming–Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5718–5729, 2022.

[57] Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Deng-Ping Fan, Radu Timofte, and Luc Van Gool. Practical blind image denoising via Swin-Conv-UNet and data synthesis. *Machine Intelligence Research*, 2023.

[58] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-Play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2022.

[59] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[60] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.