Automated data-driven selection of the hyperparameters for Total-Variation based texture segmentation

Barbara Pascal, Samuel Vaiter, Nelly Pustelnik, Patrice Abry

Automated data-driven selection of the hyperparameters for Total-Variation based texture segmentation*

Barbara Pascal† Samuel Vaiter‡ Nelly Pustelnik†
Patrice Abry†

April 2020

1 Introduction

Numerous problems in signal and image processing consist in finding the best possible estimate $\widehat{\boldsymbol{x}}$ of a quantity $\bar{\boldsymbol{x}} \in \mathcal{H}$ from an observation $\boldsymbol{y} \in \mathcal{G}$ (where $\mathcal{H}$ and $\mathcal{G}$ are Hilbert spaces isomorphic to $\mathbb{R}^N$ and $\mathbb{R}^P$ respectively), potentially corrupted by a linear operator $\boldsymbol{\Phi} : \mathcal{H} \to \mathcal{G}$, which encapsulates deformation or information loss, and by some additive zero-mean Gaussian noise $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}_P, \boldsymbol{\mathcal{S}})$, with *known* covariance matrix $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{P\times P}$, leading to the general observation model

$$\boldsymbol{y} = \boldsymbol{\Phi}\bar{\boldsymbol{x}} + \boldsymbol{\zeta}. \tag{1}$$

Examples resorting to inverse problems include image restoration [13, 55], inpainting [17], texture-geometry decomposition [3], but also texture segmentation as recently proposed in [51]. A widely investigated path for the estimation of underlying $\bar{\boldsymbol{x}}$ is linear regression [10,41], providing an unbiased linear regression estimator $\widehat{\boldsymbol{x}}_{\mathrm{LR}}$. Yet corresponding estimates suffer from large variances, which can lead to dramatic errors in the presence of noise $\boldsymbol{\zeta}$ [6].

An alternative relies on the construction of *parametric estimators*

$$\begin{array}{ccc} \mathcal{G}\times\mathbb{R}^L & \longrightarrow & \mathcal{H} \\ (\boldsymbol{y},\boldsymbol{\Lambda}) & \longmapsto & \widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) \end{array} \tag{2}$$

allowing some estimation bias, and thus leading to drastic decrease of the variance. Given some prior knowledge about ground truth $\bar{\boldsymbol{x}}$, e.g. [35], either sparsity of the variable $\bar{\boldsymbol{x}}$ [63], of its derivative [37, 59, 66] or of its wavelet transform [27], one can build *parametric estimators* performing a compromise

*Work supported by Defi Imag'in SIROCCO and by ANR-16-CE33-0020 MultiFracs, France and by ANR GraVa ANR-18-CE40-0005.

†Univ Lyon, ENS de Lyon, Univ Lyon 1, CNRS, Laboratoire de Physique, F-69342 Lyon, France (firstname.lastname@ens-lyon.fr).

‡CNRS & Université de Bourgogne Franche-Comté, Dijon, France. (samuel.vaiter@u-bourgogne.fr)

between fidelity to the model (1) and structure constraints on the estimation. In general, the compromise is tuned by a small number $L = \mathcal{O}(1)$ parameters, stored in a vector $\boldsymbol{\Lambda} \in \mathbb{R}^L$. A very popular class of *parametric estimators* relies on a penalization of a least squares data fidelity term formulated as a minimization problem

$$\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \in \underset{\boldsymbol{x} \in \mathcal{H}}{\operatorname{Argmin}} \, \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\boldsymbol{\mathcal{W}}}^2 + \|\mathbf{U}_{\boldsymbol{\Lambda}}\boldsymbol{x}\|_q^q \tag{3}$$

with $\|\cdot\|_{\boldsymbol{\mathcal{W}}}$ the Mahalanobis distance associated with $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{P \times P}$, defined as

$$\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\boldsymbol{\mathcal{W}}} \triangleq \sqrt{(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x})^\top \boldsymbol{\mathcal{W}} (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x})}. \tag{4}$$

$\mathbf{U}_{\boldsymbol{\Lambda}} : \mathcal{H} \to \mathcal{Q}$ is a linear operator parametrized by $\boldsymbol{\Lambda}$ and $\|\cdot\|_q$ the ℓ_q-norm with $q \geq 1$ in Hilbert space $\mathcal{Q}$.

Least Squares. While Ordinary Least Squares involve usual ℓ_2 squared norm as data-fidelity term, that is $\boldsymbol{\mathcal{W}} = \boldsymbol{I}_P$, Generalized Least Squares [61] make use of the covariance structure of the noise through $\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{S}}^{-1}$, encapsulating all the observation statistics in the case of Gaussian noise. This generalized approach is equivalent to decorrelating the data and equalizing noise levels before performing the regression. Further, the Gauss-Markov theorem [1] asserts that minimizing Weighted Least Squares provides the best linear estimator of $\bar{\boldsymbol{x}}$, advocating for the use of Mahalanobis distance as data fidelity term in penalized Least Squares. Yet, in practice, Generalized Least Squares (or Weighted Least Squares in the case when $\boldsymbol{\mathcal{S}}$ is diagonal) requires not only the knowledge of the covariance matrix, but also to be able to invert it. For uncorrelated data, $\boldsymbol{\mathcal{S}}$ is diagonal and, provided that it is well-conditioned, it is easy to invert numerically. On the contrary, computing $\boldsymbol{\mathcal{S}}^{-1}$ might be extremely challenging for correlated data since $\boldsymbol{\mathcal{S}}$ is not diagonal anymore and has a size scaling like the square of the dimension of $\mathcal{G}$. Thus, to handle possibly correlated Gaussian noise $\boldsymbol{\zeta}$, using Ordinary Least Squares is often mandatory, even though it does not benefit from same theoretical guarantees that Generalized Least Squares. Nevertheless, we will show that the knowledge of $\boldsymbol{\mathcal{S}}$ is far from being useless, since it is possible to take advantage of it when estimating the quadratic risk.

Penalization. Appropriate choice of q and $\mathbf{U}_{\boldsymbol{\Lambda}}$ covers a large variety of well-known estimators. Linear filtering is obtained for $q = 2$ [31], the shape of the filter being encapsulated in operator $\mathbf{U}_\Lambda$ [37], the hyperparameters $\boldsymbol{\Lambda}$ tuning e.g. its band-width. It is very common in image processing to impose priors on the spatial gradients of the image, using the finite discrete horizontal and vertical difference operator $\mathbf{D}$ and one regularization parameter $\boldsymbol{\Lambda} = \lambda > 0$ ($L = 1$). For example, smoothness of the estimate is favored using ℓ_2 squared norm, performing Tikhonov regularization [37, 66], in which $q = 2$ and $\|\mathbf{U}_{\boldsymbol{\Lambda}}\boldsymbol{x}\|_q^q \triangleq \lambda\|\mathbf{D}\boldsymbol{x}\|_2^2$. Another standard penalization is the anisotropic total variation [59] $\|\mathbf{U}_{\boldsymbol{\Lambda}}\boldsymbol{x}\|_q^q \triangleq \lambda\|\mathbf{D}\boldsymbol{x}\|_1$, corresponding to $q = 1$, where the ℓ_1-norm enforces sparsity of spatial gradients.

Risk estimation. The purpose of Problem (3) is to obtain a faithful estimation $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ of ground truth $\bar{\boldsymbol{x}}$, the error being measured by the so-called *quadratic*

risk

$$\mathbb{E}\|\mathbf{B}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \mathbf{B}\bar{\boldsymbol{x}}\|^2_{\boldsymbol{\mathcal{W}}} \tag{5}$$

with $\mathbf{B}$ a linear operator, which enables to consider various types of risk. For instance, when $\mathbf{B} = \boldsymbol{\Pi}$ is a projector on a subset of $\mathcal{H}$ [30], the *projected quadratic risk* (5) measures the estimation error on the projected quantity $\boldsymbol{\Pi}\bar{\boldsymbol{x}}$. This case includes the usual quadratic risk when $\mathbf{B} = \boldsymbol{I}_N$. Conversely, when $\mathbf{B} = \boldsymbol{\Phi}$, the risk (5) quantifies the quality of the prediction $\widehat{\boldsymbol{y}}(\boldsymbol{y};\boldsymbol{\Lambda}) \triangleq \boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ with respect to the noise-free observation $\bar{\boldsymbol{y}} \triangleq \boldsymbol{\Phi}\bar{\boldsymbol{x}}$ lying in $\mathcal{G}$, and is known as the *prediction risk*.

The main issue is that one does not have access to ground truth $\bar{\boldsymbol{x}}$. Hence, measuring the quadratic risk (5) first requires to derive an estimator of

$$\mathbb{E}\,\|\mathbf{B}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \mathbf{B}\bar{\boldsymbol{x}}\|^2_{\boldsymbol{\mathcal{W}}}$$

not involving $\bar{\boldsymbol{x}}$.

This problem was handled originally in the case of independent, identically distributed, (i.i.d.) Gaussian linear model, that is for scalar covariance matrix $\boldsymbol{\mathcal{S}} = \rho^2\boldsymbol{I}_P$, by Stein [60, 64], performing a clever integration by part, leading to Stein's Unbiased Risk Estimate (SURE) [26, 47, 57, 65], initially formulated for the *prediction risk*,

$$\|(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \boldsymbol{y})\|^2_{\boldsymbol{\mathcal{W}}} + 2\rho^2\mathrm{Tr}\left(\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right) - P\rho^2, \tag{6}$$

whose expected value equals quadratic risk (5) with $\mathbf{B} = \boldsymbol{\Phi}$. In the past years SURE was intensively used both in statistical, signal and image processing applications [11, 26, 53]. It was recently extended to the case of independent but not identically distributed noise [19, 73], corresponding to diagonal covariance matrix $\boldsymbol{\mathcal{S}} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_P^2)$, and to the case when the noise is Gaussian with potential correlations, with very general covariance matrix $\boldsymbol{\mathcal{S}}$. Yet, to the best of our knowledge, very few numerical assessments are available for Gaussian noise with non-scalar covariance matrices. A notable exception is [19], in which numerical experiments are run on uncorrelated multi-component data, the components experiencing different noise levels. The noise being assumed independent, this corresponds to a diagonal covariance matrix $\boldsymbol{\mathcal{S}} = \mathrm{diag}(\rho_1^2, \ldots, \rho_P^2)$, with ρ_i^2 the variance of the noise of the i^{th} component.

Further, in the case when the noise is neither independent identically distributed nor Gaussian, Generalized Stein Unbiased Risk Estimators were proposed, e.g. for Exponential Families [30, 38] or Poisson noise [39, 42, 45].

As for practical evaluation of Stein estimator, more sophisticated tools might be required to evaluate the second term of (6), notably when $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is obtained from a proximal splitting algorithm [4, 20, 21, 50] solving Problem (3). Indeed Stein estimator involves the Jacobian of $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ with respect to observations $\boldsymbol{y}$, which might not be directly accessible in this case. In order to manage this issue, Vonesch *et al* proposed in [69] to perform recursive forward differentiation inside the splitting scheme solving (3), which benefits from few theoretical results from [32]. This approach, even if remaining partially heuristic, proved to be efficient for a large class of problems [24].

Hyperparameter tuning. Equation (3) clearly shows that the estimate $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ drastically depends on the choice of regularization parameters $\boldsymbol{\Lambda}$. Thus, fine-tuning of regularization parameters is a long-standing problem in signal and

image processing. A common formulation of this problem consists in minimizing the quadratic risk with respect to regularization parameters $\mathbf{\Lambda}$, solving:

$$\underset{\mathbf{\Lambda}}{\text{minimize}}\ \mathbb{E}\,\|\mathbf{B}\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda}) - \mathbf{B}\bar{\boldsymbol{x}}\|_{\boldsymbol{\mathcal{W}}}^2\,. \tag{7}$$

As emphasized in [30], approximate solution of (7) found selecting among the estimates $(\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda}))_{\mathbf{\Lambda}\in\mathbb{R}^L}$ the one reaching lowest SURE (6), as proposed in pioneering work [60], leads to lower mean square error than classical Maximum Likelihood approaches applied to Model (1).

The most direct method solving (7) consists in computing SURE (6) over a grid of parameters [27, 30, 57], and to select the parameter of the grid for which SURE is minimal. Yet, grid search methods suffers from a high computation cost for several reasons. First of all, the size of the grid scaling algebraically with the number of regularization parameters L, exhaustive grid search is often inaccessible. Recently, random strategies were proposed to improve grid search efficiency [7]. Yet, for $L \geq 3$, it remains very challenging if not unfeasible. Further, an additional difficulty might appear in the case when $\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda})$ is obtained from a splitting algorithm solving Problem (3). Indeed when the regularization term $\|\mathbf{U}_{\mathbf{\Lambda}}\boldsymbol{x}\|_q^q$ is nonsmooth, the proximal algorithms solving (3) suffers from slow convergence rate, making the evaluation of Stein estimator at each point of the grid very time consuming. Although accelerated schemes were proposed [5, 16], grid search with $L \geq 2$ remains very costly, preventing from practical use.

When a closed-form expression of Stein estimator is available, exact function minimization over the regularization parameters $\mathbf{\Lambda}$ might be possible. This is the case for instance for the Tikhonov penalization for which Thompson *et al.* [62], Galatsanos *et al.* in [33], and Desbat *et al.* in [25] took advantage of the linear closed-form expression of $\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda})$ to find the "best" regularization parameter, i.e. to solve (7). Another well-known closed-form expression holds for soft-thresholding, which is widely used for wavelet-shrinkage denoising e.g. [27, 44]. Note that Generalized Cross Validation [36] also makes use of closed-form expression for parameters tuning, but in a slightly different way, working on prediction risk, solving (7) for $\mathbf{B} = \mathbf{\Phi}$. Generalized Cross Validation and Stein-based estimators were compared independently by Li [43], Thompson [62], and Desbat *et al.* in [25]. Further, Bayesian methods were proposed to deal with very large number of hyperparameters $L \gg 1$, among which Sequential Model-Based Optimization (SMBO), providing smart sampling of the hyperparameter domain [8]. Such methods are particularly adapted to machine learning, as they manage huge amount of hyperparameters without requiring knowledge of the gradient of the cost function [9].

In order to go further than (random) sampling methods, elaborated approaches relying on minimization schemes were proposed, requiring sufficiently smooth risk estimator, as well as access to its derivative with respect to $\mathbf{\Lambda}$. From a C^∞ closed-form expression of Poisson Unbiased Risk Estimate, Deledalle *et al.* [23] proposed a Newton algorithm solving (7). Nevertheless, it does not generalize, since it is very rare that one has access to all the derivatives of the risk estimator. In the case when the noise is Gaussian i.i.d., Chaux *et al.* [19] proposed and assessed numerically an empirical descent algorithm for automatic choice of regularization parameter, but with no convergence guarantee. For i.i.d. Gaussian noise and estimators built as the solution of (3), Deledalle *et al.* [24] proposed

sufficient conditions so that $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is differentiable with respect to $\boldsymbol{\Lambda}$, and then derived the differentiability of Stein's Unbiased Risk Estimate. Further, they elaborated a Stein Unbiased GrAdient estimator of the Risk (SUGAR) with the aim of performing a quasi-Newton descent solving (7) using BFGS strategy. SUGAR proved its efficiency in the automated hyperparameter selection in a spatial-spectral deconvolution method for large multispectral data corrupted by i.i.d. Gaussian noise [2]

Contributions and outline. We propose a Generalized Stein Unbiased GrAdient estimator of the Risk, for the case of Gaussian noise $\boldsymbol{\zeta}$ with *any* covariance matrix $\boldsymbol{\mathcal{S}}$, using the framework of Ordinary Least Squares, that is (5) with $\boldsymbol{\mathcal{W}} = \boldsymbol{I}_P$, enabling to manage different noise levels and correlations in the observed data.
Section 2 revisits Stein's Unbiased Estimator of the Risk in the particular case of correlated Gaussian noise with covariance matrix $\boldsymbol{\mathcal{S}}$ and derives the Finite Difference Monte Carlo SURE for this framework, extending [24]. Further, we include a projection operator $\mathbf{B} = \boldsymbol{\Pi}$ making the model versatile enough to fit various applications.
In this context, Finite Difference Monte Carlo SURE is differentiated with respect to regularization parameters leading to Finite Difference Monte Carlo Generalized Stein Unbiased GrAdient estimator of the Risk, whose asymptotic unbiasedness is demonstrated in Section 3. Generalized Stein Unbiased Risk Estimate and Generalized Stein Unbiased GrAdient estimate of the Risk are embedded in a quasi-Newton optimization scheme for automatic parameters tuning, presented in Section 3.3. Moreover, the case of sequential estimators is discussed in Section 3.2.
Then, in Section 4, the entire proposed procedure is particularized to an original application to texture segmentation based on a wavelet (multiscale) estimation of fractal attributes, proposed in [51, 52]. The texture model is cast into the general formulation (1), $\boldsymbol{y}$ corresponding to a nonlinear multiscale transform of the image to be segmented. Hence the noise $\boldsymbol{\zeta}$ presents both inter-scale and intra-scale correlations, leading to a non-diagonal covariance matrix $\boldsymbol{\mathcal{S}}$. Both Stein Unbiased Risk Estimate and Stein Unbiased GrAdient estimate of the Risk are evaluated with a Finite Difference Monte Carlo strategy, all steps of which are made explicit for the texture segmentation problem.
Finally, Section 5 is devoted to exhaustive numerical simulations assessing the performance of the proposed texture segmentation with automatic regularization parameters tuning. We notably emphasize the importance of taking into account the *full* covariance structure into account in Stein-based approaches.

2 Stein Unbiased Risk Estimate (SURE) with correlated noise

This Section details the extension of Stein Unbiased Risk Estimator (6) when $\boldsymbol{\mathcal{W}} = \boldsymbol{I}_P$ to the case when observations evidence correlated noise, leading to the Finite Difference Monte Carlo Generalized Stein Unbiased Risk Estimator, $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, defined in (17).

Notations. For a linear operator $\boldsymbol{\Phi} : \mathcal{H} \to \mathcal{G}$, the *adjoint* operator is denoted $\boldsymbol{\Phi}^*$ and characterized by: for every $\boldsymbol{x} \in \mathcal{H}$, and $\boldsymbol{y} \in \mathcal{G}$, $\langle \boldsymbol{y}, \boldsymbol{\Phi}\boldsymbol{x} \rangle = \langle \boldsymbol{\Phi}^*\boldsymbol{y}, \boldsymbol{x} \rangle$. The Jacobian with respect to observations $\boldsymbol{y}$ of a differentiable estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ is denoted $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$.

2.1 Observation model

In this work, we consider observations $\boldsymbol{y}$, supposed to follow Model (1), as stated in Assumption 1 with a degradation operator $\boldsymbol{\Phi}$ assumed to be full-rank, as stated in Assumption 2.

Assumption 1 (Gaussianity). The additive noise $\boldsymbol{\zeta} \in \mathcal{G}$ is Gaussian: $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}_P, \boldsymbol{\mathcal{S}})$, where $\mathbf{0}_P$ is the null vector of $\mathcal{G}$ and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{P \times P}$ is the covariance matrix of the noise, where $P = \dim(\mathcal{G})$. Thus, the density probability law associated with the model (1) writes

$$\boldsymbol{y} \sim \frac{1}{\sqrt{(2\pi)^P |\det \boldsymbol{\mathcal{S}}|}} \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{\Phi}\bar{\boldsymbol{x}}\|^2_{\boldsymbol{\mathcal{S}}^{-1}}}{2} \right). \tag{8}$$

Assumption 2 (Full-rank). The linear operator $\boldsymbol{\Phi} : \mathcal{H} \to \mathcal{G}$ is full rank, or equivalently $\boldsymbol{\Phi}^*\boldsymbol{\Phi}$ is invertible.

2.2 Estimation problem

Let $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ be a parametric estimator of ground truth $\bar{\boldsymbol{x}} \in \mathcal{H}$, defined in a unique manner from observations $\boldsymbol{y} \in \mathcal{G}$ and hyperparameters $\boldsymbol{\Lambda} \in \mathbb{R}^L$.

Remark 1. For instance, $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ can be the Penalized Ordinary Least Squares estimator, defined in (3). In this case full-rank Assumption 2 ensures the unicity of the minimizer. Nevertheless, we emphasize that Sections 2 and 3 address Problem (7) in a more general framework.

The possibility that the quantity of interest might be a projection of $\bar{\boldsymbol{x}}$ on a the subspace $\mathcal{I}$ of $\mathcal{H}$ is considered. One can think for instance of physics problems, in which only part of variables have a physical interpretation.

Definition 1. The linear operator $\boldsymbol{\Pi} : \mathcal{H} \to \mathcal{H}$ performs the orthogonal projection on subspace $\mathcal{I}$ capturing relevant information about $\bar{\boldsymbol{x}}$. Moreover, from both Assumption 2 and the projection operator $\boldsymbol{\Pi}$, we define the linear operator $\mathbf{A} : \mathcal{G} \to \mathcal{H}$ as the composition

$$\mathbf{A} \triangleq \boldsymbol{\Pi} \left(\boldsymbol{\Phi}^*\boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^*. \tag{9}$$

The risk is defined as the *projected estimation* error made on the quantity of interest $\boldsymbol{\Pi}\bar{\boldsymbol{x}}$ by the estimator, measured via an ordinary squared ℓ_2-norm.

$$R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) \triangleq \mathbb{E}_{\boldsymbol{\zeta}} \left\| \boldsymbol{\Pi}\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) - \boldsymbol{\Pi}\bar{\boldsymbol{x}} \right\|_2^2. \tag{10}$$

Remark 2. Another usual definition of the risk involves the inverse of the covariance matrix [30] through a Mahalanobis distance writing

$$R_{\mathrm{M}}[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) \triangleq \mathbb{E}_{\boldsymbol{\zeta}} \left\| \boldsymbol{\Pi}\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) - \boldsymbol{\Pi}\bar{\boldsymbol{x}} \right\|^2_{\boldsymbol{\mathcal{S}}^{-1}}. \tag{11}$$

requiring the knowledge of $\boldsymbol{\mathcal{S}}^{-1}$, which might be non-trivial or even inaccessible for correlated noise presenting non-diagonal covariance matrix. Hence our approach uses exclusively ordinary quadratic risk defined in (10). Nevertheless, these two approaches, even though being different, shares interesting common points which will be mentioned briefly in the following (see Remark 3).

The aim of this work is automatic fine-tuning the regularization parameters $\boldsymbol{\Lambda}$ in order to minimize the ordinary risk (10) defined above. Yet in practice, the optimal regularization parameters $\boldsymbol{\Lambda}^\dagger$ satisfying

$$\boldsymbol{\Lambda}^\dagger \in \underset{\boldsymbol{\Lambda}\in\mathbb{R}^L}{\operatorname{Argmin}}\ R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) \tag{12}$$

is inaccessible. In the following, we propose a detailed procedure to closely approach $\boldsymbol{\Lambda}^\dagger$, by minimizing a Generalized Stein Unbiased Risk Estimator approximating $R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda})$.

2.3 Generalized Stein Unbiased Risk Estimator

The risk defined in (10) depends explicitly on ground truth $\bar{\boldsymbol{x}}$ and hence is inaccessible. Stein proposed an unbiased estimator of this risk, known as Stein Unbiased Risk Estimator (SURE) in the case of i.i.d. Gaussian noise, recalled in Equation (6). This estimator was then extended to very general noise distributions (see e.g. [30] for Exponential Families, including Gaussian densities). In particular, when the noise $\boldsymbol{\zeta}$ is Gaussian, with possible non-trivial covariance matrix, Theorem 1 provides a generalization of Stein's original estimator, which constitutes the starting point of this work.
Stein's approach for risk estimation crucially relies the following hypothesis on estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$:

Assumption 3 (Regularity and integrability). The estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is continuous and weakly differentiable with respect to observations $\boldsymbol{y}$. Moreover, the quantities $\langle \mathbf{A}^*\boldsymbol{\Pi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}), \boldsymbol{\zeta}\rangle$ and $\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ are integrable against the Gaussian density:

$$\frac{1}{\sqrt{(2\pi)^P|\det\boldsymbol{\mathcal{S}}|}}\exp\left(-\frac{\|\boldsymbol{\zeta}\|^2_{\boldsymbol{\mathcal{S}}^{-1}}}{2}\right)\,\mathrm{d}\boldsymbol{\zeta}.$$

Theorem 1. *Consider Model* (1)*, together with Assumptions 1 (Gaussianity), 2 (Full-rank), 3 (Integrability), and linear operator* $\boldsymbol{A}$ *defined in* (9)*. Then generalized Stein's lemma applies, and leads to*

$$R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) = \mathbb{E}_{\boldsymbol{\zeta}}\left[\|\boldsymbol{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})-\boldsymbol{y}\right)\|_2^2 + 2\mathrm{Tr}\left(\boldsymbol{\mathcal{S}}\boldsymbol{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right) - \mathrm{Tr}(\boldsymbol{A}\boldsymbol{\mathcal{S}}\boldsymbol{A}^*)\right], \tag{13}$$

the quantity in the brackets being the so-called Generalized Stein Unbiased Risk Estimator.

Proof. A detailed proof is provided in Appendix A. □

Remark 3. Interestingly, when considering the squared Mahalanobis distance in the defintion of the risk (11), Stein Unbiased Risk Estimator has the same

global structure, yet, instead of involving the covariance matrix $\boldsymbol{\mathcal{S}}$ it involves its inverse writing

$$\widetilde{R}[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) = \mathbb{E}_{\boldsymbol{\zeta}}\left[\|\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \boldsymbol{y}\right)\|^2_{\boldsymbol{\mathcal{S}}^{-1}} + 2\mathrm{Tr}\left(\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right) - \mathrm{Tr}(\mathbf{A}\mathbf{A}^*)\right]. \tag{14}$$

2.4 Finite Difference Monte Carlo SURE

In the proposed SURE expression (13), the quantity $\mathrm{Tr}\left(\boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right)$ appearing in (13), called the *degrees of freedom*, concentrates the major difficulties in computing Stein's estimator in data processing problems, as evidenced by the prolific literature addressing this issue in the case $\boldsymbol{\mathcal{S}} \propto \boldsymbol{I}_P$ [28, 40, 65, 67]. Indeed, it involves the product of the $P \times P$ matrix $\boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}$ with the $P \times P$ Jacobian matrix $\partial_{\boldsymbol{y}}(\widehat{\boldsymbol{x}}(\boldsymbol{y},\boldsymbol{\Lambda}))$. Not only the product of two $P \times P$ matrices might be extremely costly in computational efforts but also the Jacobian matrix, because of its large size, $P \gg 1$, might also be very demanding to compute (or even to estimate). Two-step Finite Difference Monte Carlo strategy together with Assumption 4 presented below, enable to overcome theses difficulties and to built a usable Stein Unbiased Risk Estimator, denoted $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, defined in Equation (17).

Assumption 4 (Lipschitzianity w.r.t. observations). Let $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ an estimator of $\bar{\boldsymbol{x}}$, depending on observations $\boldsymbol{y}$, and parametrized by $\boldsymbol{\Lambda}$.
(i) The mapping $\boldsymbol{y} \mapsto \widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is uniformly L_1-Lipschitz .
(ii) $\forall\, \boldsymbol{\Lambda} \in \mathbb{R}^L$, $\widehat{\boldsymbol{x}}(\mathbf{0}_P;\boldsymbol{\Lambda}) = \mathbf{0}_N$, with $\mathbf{0}_N$ (resp. $\mathbf{0}_P$) the null vector of $\mathcal{H}$ (resp. $\mathcal{G}$).

Step 1. Trace estimation via Monte Carlo:
In the way to practical degrees of freedom estimation, the first step is to remark that it far less costly to compute the product of the $P \times P$ matrix $\boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}$ with $\partial_{\boldsymbol{y}}(\widehat{\boldsymbol{x}}(\boldsymbol{y},\boldsymbol{\Lambda}))[\boldsymbol{\varepsilon}] \in \mathbb{R}^P$, the Jacobian matrix applied on a vector $\boldsymbol{\varepsilon} \in \mathbb{R}^P$. Further, straightforward computation shows that if $\boldsymbol{\varepsilon} \in \mathbb{R}^P$ is a normalized random variable $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_P, \boldsymbol{I}_P)$, and $\mathbf{M} \in \mathbb{R}^{P\times P}$ any matrix, then

$$\mathrm{Tr}(\mathbf{M}) = \mathbb{E}_{\boldsymbol{\varepsilon}}\langle \mathbf{M}\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}\rangle.$$

Thus, following the suggestion of [24, 34, 57], if one has access to $\partial_{\boldsymbol{y}}(\widehat{\boldsymbol{x}}(\boldsymbol{y},\boldsymbol{\Lambda}))\,[\boldsymbol{\varepsilon}]$, then, since $\boldsymbol{\mathcal{S}}$ is a covariance matrix and hence is symmetric,

$$\begin{aligned}\mathrm{Tr}\left(\boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right) &= \mathbb{E}_{\boldsymbol{\varepsilon}}\left\langle \boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})[\boldsymbol{\varepsilon}], \boldsymbol{\varepsilon}\right\rangle \\ &= \mathbb{E}_{\boldsymbol{\varepsilon}}\left\langle \mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})[\boldsymbol{\varepsilon}], \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle, \end{aligned} \tag{15}$$

and $\left\langle \mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})[\boldsymbol{\varepsilon}], \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle$ provides an estimator of degrees of freedom.

Step 2. First-order derivative estimation with Finite Differences:
Second step consists in tackling the problem of estimating $\partial_{\boldsymbol{y}}(\widehat{\boldsymbol{x}}(\boldsymbol{y},\boldsymbol{\Lambda}))\,[\boldsymbol{\varepsilon}]$ when no direct access to the Jacobian $\partial_{\boldsymbol{y}}(\widehat{\boldsymbol{x}}(\boldsymbol{y},\boldsymbol{\Lambda}))$ is possible. In this case, the derivative can be estimated using the normalized random variable $\boldsymbol{\varepsilon}$ and a step $\nu > 0$

making use of Taylor expansion

$$\begin{aligned} & \widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) \underset{\nu\to 0}{\simeq} \partial_{\boldsymbol{y}}(\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}))\,[\nu\boldsymbol{\varepsilon}] \\ \iff \quad & \partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\,[\boldsymbol{\varepsilon}] = \lim_{\nu\to 0}\frac{1}{\nu}\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right). \end{aligned}$$

It follows

$$\begin{aligned} \operatorname{Tr}\left(\boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right) &= \mathbb{E}_{\boldsymbol{\varepsilon}}\lim_{\nu\to 0}\frac{1}{\nu}\left\langle \boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right),\boldsymbol{\varepsilon}\right\rangle \\ &= \mathbb{E}_{\boldsymbol{\varepsilon}}\lim_{\nu\to 0}\frac{1}{\nu}\left\langle \mathbf{A}^*\boldsymbol{\Pi}\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right),\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle. \quad (16) \end{aligned}$$

Elaborating on Formula (16) and Assumption 4, the following theorem provides an asymptotically unbiased Finite Differences Monte Carlo estimator of the risk, which can be used in a vast variety of estimation problems.

Theorem 2. *Consider the observation Model* (1)*, the operator* $\boldsymbol{A}$ *defined in* (9) *together with Assumptions 1 (Gaussianity), 2 (Full-rank), 3 (Integrability), and 4 (Lipschitzianity w.r.t.* $\boldsymbol{y}$*). Generalized Finite Differences Monte Carlo SURE, writing*

$$\begin{aligned} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) &\triangleq \|\boldsymbol{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})-\boldsymbol{y}\right)\|_2^2 \quad (17) \\ &+\frac{2}{\nu}\left\langle \boldsymbol{A}^*\boldsymbol{\Pi}\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right),\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle - \operatorname{Tr}(\boldsymbol{A}\boldsymbol{\mathcal{S}}\boldsymbol{A}^*), \end{aligned}$$

is an asymptotically unbiased estimator of the risk $R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda})$ *as* $\nu\to 0$*, meaning that*

$$\lim_{\nu\to 0}\mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) = R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}). \quad (18)$$

Proof. The proof of Theorem 2 is postponed to Appendix B. □

Remark 4. The use of Monte Carlo strategy is advocated in [24] so that to reduce the complexity of SURE evaluation, replacing costly $P\times P$ matrices product by products of $P\times P$ matrix by vector of size P. Yet, the product of $\boldsymbol{\mathcal{S}}\in\mathbb{R}^{P\times P}$ with $\boldsymbol{\varepsilon}\in\mathbb{R}^P$, as well as the product of $\mathbf{A}^*\boldsymbol{\Pi}\in\mathbb{R}^{P\times N}$ with $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\in\mathbb{R}^N$, might still be extremely costly. Hopefully, we will see that, in data processing problems (e.g. for texture segmentation in Section 4), both the covariance matrix $\boldsymbol{\mathcal{S}}$ and linear operator $\mathbf{A}$ (through the degradation $\boldsymbol{\Phi}$) benefit from sufficient sparsity so that the calculations can be handled at a reasonable cost.

3 Stein's Unbiased GrAdient estimator of the Risk (SUGAR)

From the estimator of the risk $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ provided in previous Section 2.3, basic grid search approach could be performed, in order to estimate the optimal $\boldsymbol{\Lambda}^\dagger$, as defined in (12). Yet, the exploration of a fine grid of $\boldsymbol{\Lambda}\in\mathbb{R}^L$ might be

time consuming if the evaluation of $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is costly, which is the case when $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is *sequential*, *i.e.* obtained from an optimization scheme. Moreover, the size of a grid in $\mathbb{R}^L$ with given step size grows algebraically with L. Altogether, this precludes grid search when $L > 2$.
Inspiring from [24], this section addresses this issue in the extended case of correlated noise. We provide in Equation (19) a generalized estimator $\partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) \in \mathbb{R}^L$ of the gradient of the risk with respect to hyperparameters $\boldsymbol{\Lambda}$. Further, we demonstrate that the Finite Difference Monte Carlo estimator $\partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ is an asymptotically unbiased estimator of the gradient of the risk (10) with respect to $\boldsymbol{\Lambda}$.
In Algorithm 1, we provide an example of sequential estimator, relying on an accelerated primal-dual scheme, designed to solve (3), with its differentiated counterpart, providing both $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ and its Jacobian $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) \in \mathbb{R}^{N\times L}$ with respect to $\boldsymbol{\Lambda}$.
Hence, costly grid search can be avoided, the estimation of $\boldsymbol{\Lambda}^\dagger$ being performed by a quasi-Newton descent, described in Algorithm 3, which minimizes the estimated risk $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, making use of its gradient $\partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$.

3.1 Differentiation of Stein Unbiased Risk Estimate

Proposition 1. *Consider the observation Model* (1)*, the operator* $\boldsymbol{A}$ *defined in* (9) *together with Assumptions 1 (Gaussianity), 2 (Full-rank), 3 (Integrability), 4 (Lipschitzianity w.r.t.* $\boldsymbol{y}$*), and 5 (Lipschitzianity w.r.t.* $\boldsymbol{\Lambda}$*) Assumptions. Then the Finite Difference Monte Carlo SURE* $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$*, defined in* (17)*, is weakly differentiable with respect to both observations* $\boldsymbol{y}$ *and parameters* $\boldsymbol{\Lambda}$*, and its gradient with respect to* $\boldsymbol{\Lambda}$*, as an element of* $\mathbb{R}^L$*, is given by*

$$\begin{aligned}\partial_{\boldsymbol{\Lambda}} \left[\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})\right] \triangleq\ & 2\left(\boldsymbol{A}\boldsymbol{\Phi}\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right)^* \boldsymbol{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \boldsymbol{y}\right) \\ & + \frac{2}{\nu}\left(\boldsymbol{A}^*\boldsymbol{\Pi}\left(\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}) - \partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right)\right)^* \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon},\end{aligned} \tag{19}$$

Proof. The Finite Difference Monte Carlo SURE $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, defined by Formula (17) is a combination of continuous and weakly differentiable functions with respect to both observations $\boldsymbol{y}$ and parameters $\boldsymbol{\Lambda}$, composed with (bounded) linear operators, and thus is continuous and weakly differentiable. Further, the derivation rules apply and lead to the expression of Finite Difference Monte Carlo SUGAR estimator given in Formula (19).

□

Assumption 5 (Lipschitzianity w.r.t. hyperparameters). Let $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ be an estimator of $\bar{\boldsymbol{x}}$, depending on observations $\boldsymbol{y}$, and parametrized by $\boldsymbol{\Lambda}$. The mapping $\boldsymbol{\Lambda} \mapsto \widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is uniformly L_2-Lipschitz continuous with constant L_2 being independent of $\boldsymbol{y}$.

Remark 5. As argued in [24], when the estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ can be expressed as a (composition of) proximal operator(s) of gauge(s) of compact set(s)[1], Assumption 5 holds. Thus, in the case of (3) when $\mathcal{G} = \mathcal{H}$, $\boldsymbol{\Phi} = \boldsymbol{I}_{\mathcal{H}}$, and $\|\mathbf{U}_{\boldsymbol{\Lambda}}\boldsymbol{x}\|_q^q = \lambda\|\boldsymbol{x}\|_q^q$, for any $q \geq 1$ the Lipschitzianity w.r.t. $\boldsymbol{\Lambda}$ is ensured. Moreover, in the case of Tikhonov regularization, i.e. $q = 2$ and $\|\mathbf{U}_{\boldsymbol{\Lambda}}\boldsymbol{x}\|_q^q = \lambda\|\mathbf{D}\boldsymbol{x}\|_2^2$

[1] For $\mathcal{C} \subset \mathcal{G}$ a non-empty closed convex set containing $\mathbf{0}_{\mathcal{G}}$, the gauge of $\mathcal{C}$ is defined as $\gamma_{\mathcal{C}}(\boldsymbol{y}) \triangleq \inf\{\omega > 0 \,|\, \boldsymbol{y} \in \omega\mathcal{C}\}$.

in (3), if $\boldsymbol{\Phi} = \boldsymbol{I}_{\mathcal{H}}$ and $\mathbf{D}^*\mathbf{D}$ is diagonalizable with strictly positive eigenvalues, then Assumption 5 is verified. Apart from these two well-known examples, proving the validity of Assumption 5 in the general case of Penalized Least Square is a difficult problem and is foreseen for future work.

Theorem 3. *Consider the observation Model* (1)*, the operator* $\boldsymbol{A}$ *defined in* (9) *together with Gaussianity 1, Full-rank 2, Integrability 3, Lipschitzianity w.r.t.* $\boldsymbol{y}$ *4, and Lipschitzianity w.r.t.* $\boldsymbol{\Lambda}$ *5 Assumptions. Then generalized Finite Difference Monte Carlo SUGAR,* $\partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ *defined in Equation* (19)*, is an asymptotically unbiased estimate of the gradient of the risk as* $\nu \to 0$*, that is*

$$\partial_{\boldsymbol{\Lambda}} R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) = \lim_{\nu \to 0} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}; \boldsymbol{\Lambda} | \boldsymbol{\mathcal{S}}) \tag{20}$$

Proof. The proof of Theorem 3 is postponed to Appendix C. □

Remark 6. Finite Difference Monte Carlo estimator of the gradient of the risk, $\partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}; \boldsymbol{\Lambda} | \boldsymbol{\mathcal{S}})$, defined in Equation (19), involves the Jacobian $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \in \mathbb{R}^{N \times L}$ which could be a very large matrix, raising difficulties for practical use. Nevertheless, in most applications, the regularization hyperparameters $\boldsymbol{\Lambda} \in \mathbb{R}^L$, have a "low" dimensionality $L = \mathcal{O}(1) \ll N$. Thus, it is reasonable to expect that the Jacobian matrix $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \in \mathbb{R}^{N \times L}$ can be stored and manipulated, with similar memory and computational costs than for $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ (see Section 3.2).

3.2 Sequential estimators and forward iterative differentiation

The evaluation of $\partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ from Formula (19) requires the Jacobian $\partial_{\boldsymbol{\Lambda}} \widehat{x}(\boldsymbol{y}; \boldsymbol{\Lambda})$. Yet, when no closed-form expression of estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ is available, computing the gradient $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ might be a complicated task. A large class of estimators $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ lacking closed-form expression are those obtained as the limit of iterates as

$$\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) = \lim_{k \to \infty} \boldsymbol{x}^{[k]}(\boldsymbol{y}; \boldsymbol{\Lambda}), \tag{21}$$

for instance when $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ is defined as the solution of a minimization problem, e.g. (3).
In the case when $\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ is a sequential estimator, given an observation $\boldsymbol{y}$, it is only possible to sample the function $\boldsymbol{\Lambda} \mapsto \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$, for a *discrete* set of regularization hyperparameters $\{\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \ldots\}$, running the minimization algorithm for each hyperparameters $\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \ldots$. It is a classical fact in signal processing that no robust estimator of the differential can be built from samples of the function, thus more sophisticated tools are needed. Provided some smoothness conditions on the iterations of the minimization algorithm, iterative differentiation strategy [24] gives access to a sequence of Jacobian $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{x}}^{[k]}(\boldsymbol{y}; \boldsymbol{\Lambda})$, relying on *chain rule* differentiation presented in Proposition 2.

Considering Problem (3), splitting algorithms [4, 20, 50] are advocated to perform the minimization. We chose the primal-dual scheme proposed in [16], Algorithm 2, taking advantage of closed-form expressions of the proximal op-

erators [58] of both the data-fidelity term and the penalization[2]. Chambolle-Pock algorithm, particularized to (3), is presented in Algorithm 1. Further, $\boldsymbol{\Phi}$ being full-rank (Assumption 2), denoting by $\mathrm{Sp}(\boldsymbol{\Phi}^*\boldsymbol{\Phi})$ the spectrum of $\boldsymbol{\Phi}^*\boldsymbol{\Phi}$, $\gamma = 2\min \mathrm{Sp}(\boldsymbol{\Phi}^*\boldsymbol{\Phi})$ is strictly positive. Hence the data-fidelity in (3) term turns out to be γ-strongly convex, and the primal-dual algorithm can be accelerated thank to Step (31) of Algorithm 1, following [16]. The iterative differentiation strategy providing $\partial_{\boldsymbol{\Lambda}} \boldsymbol{x}^{[k]}(\boldsymbol{y}; \boldsymbol{\Lambda})$ is presented in the second part of Algorithm 1. Other iterative differentiation schemes are detailed in [24].

Proposition 2. *Let $\boldsymbol{\Psi} : \mathbb{R}^{N\times L} \to \mathbb{R}^N$ be a differentiable function of variable $\boldsymbol{x} \in \mathcal{H}$, differentiably parametrized by $\boldsymbol{\Lambda} \in \mathbb{R}^L$, and $\left(\boldsymbol{x}^{[k]}\right)_{k\in\mathbb{N}}$ the sequential estimator defined by iterations of the form*

$$\boldsymbol{x}^{[k+1]} = \boldsymbol{\Psi}(\boldsymbol{x}^{[k]}; \boldsymbol{\Lambda}). \tag{22}$$

The gradient of $\boldsymbol{x}^{[k]}$ with respect to $\boldsymbol{\Lambda}$ can be computed making use of the chain rule differentiation

$$\partial_{\boldsymbol{\Lambda}} \boldsymbol{x}^{[k+1]} = \partial_{\boldsymbol{\Lambda}} \left(\boldsymbol{\Psi}(\boldsymbol{x}^{[k]}; \boldsymbol{\Lambda}) \right) = \partial_{\boldsymbol{x}} \boldsymbol{\Psi}(\boldsymbol{x}^{[k]}; \boldsymbol{\Lambda})[\partial_{\boldsymbol{\Lambda}} \boldsymbol{x}^{[k]}] + \partial_{\boldsymbol{\Lambda}} \boldsymbol{\Psi}(\boldsymbol{x}^{[k]}; \boldsymbol{\Lambda}), \tag{23}$$

where $\partial_{\boldsymbol{x}} \boldsymbol{\Psi}(\boldsymbol{x}; \boldsymbol{\Lambda})[\boldsymbol{\delta}]$ denotes the differential of $\boldsymbol{\Psi}$ with respect to variable $\boldsymbol{x}$ applied on vector $\boldsymbol{\delta}$, and $\partial_{\boldsymbol{\Lambda}} \boldsymbol{\Psi}(\boldsymbol{x}; \boldsymbol{\Lambda})$ the gradient of $\boldsymbol{\Psi}$ with respect to $\boldsymbol{\Lambda}$. The differentiability of $\boldsymbol{\Psi}$ should be understood in the weak sense.

Remark 7. Two particular cases are often encountered in iterative differentiation (see Algorithm 1):
(i) Linear operator $\boldsymbol{\Psi}(\boldsymbol{x}; \boldsymbol{\Lambda}) \triangleq \mathbf{U}_{\boldsymbol{\Lambda}} \boldsymbol{x}$. Assuming that $(\boldsymbol{x} \mapsto \mathbf{U}_{\boldsymbol{\Lambda}} \boldsymbol{x})_{\boldsymbol{\Lambda}}$ is a family of linear operators, with a differentiable parametrization by $\boldsymbol{\Lambda}$, the *chain rule* writes

$$\partial_{\boldsymbol{\Lambda}} \boldsymbol{x}^{[k+1]} = \mathbf{U}_{\boldsymbol{\Lambda}} \partial_{\boldsymbol{\Lambda}} \boldsymbol{x}^{[k]} + \left(\partial_{\boldsymbol{\Lambda}} \mathbf{U}_{\boldsymbol{\Lambda}}\right) \boldsymbol{x}^{[k]}, \tag{24}$$

since the differential of the linear operator $\mathbf{U}_{\boldsymbol{\Lambda}}$ with respect to $\boldsymbol{x}$ is itself. See (33) and (35), in Algorithm 1 for applications of the *chain rule* with linear operators.
(ii) Proximal operator $\boldsymbol{\Psi}(\boldsymbol{x}; \boldsymbol{\Lambda}) \triangleq \mathrm{prox}_{\tau\|\cdot\|_{2,1}}(\boldsymbol{x})$. The proximal operator being independent of $\boldsymbol{\Lambda}$, the *chain rule* simplifies to

$$\partial_{\boldsymbol{\Lambda}} \boldsymbol{x}^{[k+1]} = \partial_{\boldsymbol{x}} \mathrm{prox}_{\tau\|\cdot\|_{2,1}}(\boldsymbol{x}^{[k]})[\partial_{\boldsymbol{\Lambda}} \boldsymbol{x}^{[k]}] \tag{25}$$

with the differential of the so-called $\ell_2 - \ell_1$ *soft-thresholding* $\mathrm{prox}_{\tau\|\cdot\|_{2,1}}$ with respect to $\boldsymbol{x} = (x_1, x_2)$, applied on $\boldsymbol{\delta} = (\delta_1, \delta_2)$ having the closed-form expression

$$\partial_{\boldsymbol{x}} \mathrm{prox}_{\tau\|\cdot\|_{2,1}}(\boldsymbol{x})[\boldsymbol{\delta}] = \begin{cases} \mathbf{0} & \text{if} \quad \|\boldsymbol{x}\|_2 \leq \tau \\ \boldsymbol{\delta} - \frac{\tau}{\|\boldsymbol{x}\|_2} \left(\boldsymbol{\delta} - \frac{\langle \boldsymbol{\delta}, \boldsymbol{x} \rangle}{\|\boldsymbol{x}\|_2^2} \boldsymbol{x} \right) & \text{else.} \end{cases} \tag{26}$$

See (34) and (36), in Algorithm 1 for applications of the *chain rule* with proximal operators.

[2] see `http://proximity-operator.net` for numerous proximal operator closed-form expressions

Algorithm 1 Accelerated primal-dual scheme for solving (3) with iterative differentiation with respect to regularization parameters $\boldsymbol{\Lambda}$.

Routines: $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) = \mathrm{PD}(\boldsymbol{y},\boldsymbol{\Lambda})$
$(\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}), \partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})) = \partial\mathrm{PD}(\boldsymbol{y},\boldsymbol{\Lambda})$

Inputs: Observations $\boldsymbol{y}$
Regularization hyperparameters $\boldsymbol{\Lambda}$
Strong-convexity modulus of data-fidelity term $\gamma = 2\min \mathrm{Sp}(\boldsymbol{\Phi}^*\boldsymbol{\Phi})$

Initialization: Descent steps $\boldsymbol{\tau}^{[0]} = (\tau_1^{[0]}, \tau_2^{[0]})$ such that $\tau_1^{[0]}\tau_2^{[0]}\|\mathbf{U}_{\boldsymbol{\Lambda}}\|^2 < 1$
Primal, auxiliary and dual variables $\boldsymbol{x}^{[0]} \in \mathcal{H}$, $\boldsymbol{w}^{[0]} = \boldsymbol{x}^{[0]}$, $\boldsymbol{z}^{[0]} \in \mathcal{Q}$

Preliminaries: Jacobian with respect to $\boldsymbol{\Lambda}$: $\partial_{\boldsymbol{\Lambda}}\boldsymbol{x}^{[0]}$, $\partial_{\boldsymbol{\Lambda}}\boldsymbol{w}^{[0]}$, $\partial_{\boldsymbol{\Lambda}}\boldsymbol{z}^{[0]}$

for $k = 1$ **to** $K_{\max}$ **do**

{Accelerated Primal-Dual}

$$\widetilde{\boldsymbol{z}}^{[k]} = \boldsymbol{z}^{[k]} + \tau_1^{[k]}\mathbf{U}_{\boldsymbol{\Lambda}}\boldsymbol{w}^{[k]} \tag{27}$$

$$\boldsymbol{z}^{[k+1]} = \mathrm{prox}_{\tau_1^{[k]}(\|\cdot\|_q^q)^*}\left(\widetilde{\boldsymbol{z}}^{[k]}\right) \tag{28}$$

$$\widetilde{\boldsymbol{x}}^{[k]} = \boldsymbol{x}^{[k]} - \tau_2^{[k]}\mathbf{U}_{\boldsymbol{\Lambda}}^*\boldsymbol{z}^{[k+1]} \tag{29}$$

$$\boldsymbol{x}^{[k+1]} = \mathrm{prox}_{\tau_2^{[k]}\|\boldsymbol{y}-\boldsymbol{\Phi}\cdot\|_2^2}\left(\widetilde{\boldsymbol{x}}^{[k]}\right) \tag{30}$$

$$\theta^{[k]} = 1/\sqrt{1+2\gamma\tau_2^{[k]}}, \quad \tau_1^{[k+1]} = \tau_1^{[k]}/\theta^{[k]}, \quad \tau_2^{[k+1]} = \theta^{[k]}\tau_2^{[k]} \tag{31}$$

$$\boldsymbol{w}^{[k+1]} = \boldsymbol{x}^{[k]} + \theta^{[k]}\left(\boldsymbol{x}^{[k+1]} - \boldsymbol{x}^{[k]}\right) \tag{32}$$

{Accelerated Differentiated Primal-Dual}

$$\partial_{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{z}}^{[k]} = \partial_{\boldsymbol{\Lambda}}\boldsymbol{z}^{[k]} + \tau_1^{[k]}\mathbf{U}_{\boldsymbol{\Lambda}}\partial_{\boldsymbol{\Lambda}}\boldsymbol{w}^{[k]} + \tau_1^{[k]}\frac{\partial\mathbf{U}_{\boldsymbol{\Lambda}}}{\partial\boldsymbol{\Lambda}}\boldsymbol{w}^{[k]} \tag{33}$$

$$\partial_{\boldsymbol{\Lambda}}\boldsymbol{z}^{[k+1]} = \partial_{\widetilde{\boldsymbol{z}}}\mathrm{prox}_{\tau_1^{[k]}(\|\cdot\|_q^q)^*}\left(\widetilde{\boldsymbol{z}}^{[k]}\right)\left[\partial_{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{z}}^{[k]}\right] \tag{34}$$

$$\partial_{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{x}}^{[k]} = \partial_{\boldsymbol{\Lambda}}\boldsymbol{x}^{[k]} - \tau_2^{[k]}\mathbf{U}_{\boldsymbol{\Lambda}}^*\partial_{\boldsymbol{\Lambda}}\boldsymbol{z}^{[k+1]} - \tau_2^{[k]}\frac{\partial\mathbf{U}_{\boldsymbol{\Lambda}}}{\partial\boldsymbol{\Lambda}}\boldsymbol{z}^{[k+1]} \tag{35}$$

$$\partial_{\boldsymbol{\Lambda}}\boldsymbol{x}^{[k+1]} = \partial_{\widetilde{\boldsymbol{x}}}\mathrm{prox}_{\tau_2^{[k]}\|\boldsymbol{y}-\boldsymbol{\Phi}\cdot\|_2^2}\left(\widetilde{\boldsymbol{x}}^{[k]}\right)\left[\partial_{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{x}}^{[k]}\right] \tag{36}$$

$$\partial_{\boldsymbol{\Lambda}}\boldsymbol{w}^{[k+1]} = \partial_{\boldsymbol{\Lambda}}\boldsymbol{x}^{[k]} + \theta^{[k]}\left(\partial_{\boldsymbol{\Lambda}}\boldsymbol{x}^{[k+1]} - \partial_{\boldsymbol{\Lambda}}\boldsymbol{x}^{[k]}\right) \tag{37}$$

end for

Outputs: *Finite-time* solution of Problem (3) $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) \triangleq \widehat{\boldsymbol{x}}^{[K_{\max}]}$
Finite-time Jacobian w.r.t. hyperparameters $\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) \triangleq \partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}^{[K_{\max}]}$

Definition 2 (Generalized SURE and SUGAR for sequential estimators)**.** Let $\widehat{\boldsymbol{x}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ be a sequential estimator in the sense of (21). The associated risk es-

timate $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ and gradient of the risk estimate $\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ are computed running Algorithm 1 twice: first with input $\boldsymbol{y}$ (observations), second with input $\boldsymbol{y}+\nu\boldsymbol{\varepsilon}$ (perturbed observations). Then, generalized SURE is computed from Formula (17), and generalized SUGAR from Formula (19). These steps are summarized into routines respectively called "SURE" and "SUGAR", detailed in Algorithm 2.

3.3 Automatic risk minimization

Theorem 2 provides an asymptotically unbiased estimator of the risk $R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda})$, denoted $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, based on Finite Difference Monte Carlo strategy. Hence, for sufficiently small Finite Difference step $\nu > 0$, we can expect that the solution $\boldsymbol{\Lambda}^\dagger$ of Problem (7), minimizing the true risk, is well approximated by the hyperparameters $\widehat{\boldsymbol{\Lambda}}^\dagger_{\nu,\boldsymbol{\varepsilon}}$ minimizing the estimated risk

$$\widehat{\boldsymbol{\Lambda}}^\dagger_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}|\boldsymbol{\mathcal{S}}) \in \underset{\boldsymbol{\Lambda}\in\mathbb{R}^L}{\operatorname{Argmin}}\ \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}). \tag{45}$$

Then, since the dimensionality of $\boldsymbol{\Lambda} \in \mathbb{R}^L$ is "low" enough (see Remark 6), Problem (45) is addressed performing a quasi-Newton descent algorithm, using the estimated gradient of the risk $\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, provided by Theorem 3.

A sketch of quasi-Newton descent, particularized to Problem (45), is detailed in Algorithm 3. It generates a sequence $\left(\boldsymbol{\Lambda}^{[t]}\right)_{t\in\mathbb{N}}$ converging toward a minimizer of $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$. This algorithm relies on a gradient descent step (48) involving a descent direction $\boldsymbol{d}^{[t]}$ obtained from the product of BFGS approximated inverse Hessian matrix $\boldsymbol{H}^{[t]}$ and the gradient $\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ obtained from SUGAR (see Algorithm 2). The descent step size $\alpha^{[t]}$ is obtained from a line search, derived in (47), which stops when Wolfe conditions are fulfilled [22, 49]. Finally, the approximated inverse Hessian matrix $\boldsymbol{H}^{[t]}$ is updated according to Definition 3.

Remark 8. The line search, Step (47), is the most time consuming. Indeed, the routines SURE and SUGAR are called for several hyperparameters of the form $\boldsymbol{\Lambda}^{[t]} + \alpha\boldsymbol{d}^{[t]}$, each call requiring to run differentiated primal-dual scheme twice.

Definition 3 (BroydenFletcherGoldfarbShanno (BFGS))**.** Let $\boldsymbol{d}^{[t]}$ be the descent direction and $\boldsymbol{u}^{[t]}$ the gradient increment at iteration t, the approximated inverse Hessian matrix $\boldsymbol{H}^{[t]}$ BFGS update writes

$$\boldsymbol{H}^{[t+1]} = \left(\boldsymbol{I}_L - \frac{\boldsymbol{d}^{[t]}\left(\boldsymbol{u}^{[t]}\right)^\top}{\left(\boldsymbol{u}^{[t]}\right)^\top\boldsymbol{d}^{[t]}}\right)\boldsymbol{H}^{[t]}\left(\boldsymbol{I}_L - \frac{\boldsymbol{u}^{[t]}\left(\boldsymbol{d}^{[t]}\right)^\top}{\left(\boldsymbol{u}^{[t]}\right)^\top\boldsymbol{d}^{[t]}}\right) + \alpha^{[t]}\frac{\boldsymbol{d}^{[t]}\left(\boldsymbol{d}^{[t]}\right)^\top}{\left(\boldsymbol{u}^{[t]}\right)^\top\boldsymbol{d}^{[t]}}. \tag{52}$$

This step constitutes a routine, named "BFGS", defined as

$$\boldsymbol{H}^{[t+1]} \triangleq \mathrm{BFGS}(\boldsymbol{H}^{[t]}, \boldsymbol{d}^{[t]}, \boldsymbol{u}^{[t]}). \tag{53}$$

Algorithm 2 Generalized SURE and SUGAR for sequential $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$.

Routines: $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) = \mathrm{SURE}(\boldsymbol{y},\boldsymbol{\Lambda},\boldsymbol{\mathcal{S}},\nu,\boldsymbol{\varepsilon})$
$\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) = \mathrm{SUGAR}(\boldsymbol{y},\boldsymbol{\Lambda},\boldsymbol{\mathcal{S}},\nu,\boldsymbol{\varepsilon})$

Inputs: Observations $\boldsymbol{y}$
Regularization hyperparameters $\boldsymbol{\Lambda}$
Covariance matrix $\boldsymbol{\mathcal{S}}$
Monte Carlo vector $\boldsymbol{\varepsilon} \in \mathbb{R}^P \sim \mathcal{N}(\mathbf{0}_P, \boldsymbol{I}_P)$
Finite Difference step $\nu > 0$

{Solution of (3) from Algorithm 1}

$$\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) = \mathrm{PD}(\boldsymbol{y},\boldsymbol{\Lambda}) \tag{38}$$
$$\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}) = \mathrm{PD}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon},\boldsymbol{\Lambda}) \tag{39}$$

{Finite Difference Monte Carlo SURE (17)}

$$\begin{aligned}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) &= \|\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \boldsymbol{y}\right)\|_2^2 \\ &+ \frac{2}{\nu}\left\langle \mathbf{A}^*\boldsymbol{\Pi}\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right), \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle - \mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*)\end{aligned} \tag{40}$$

{Solution of (3) and its differential w.r.t. $\boldsymbol{\Lambda}$ from Algorithm 1}

$$(\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}), \partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})) = \partial\mathrm{PD}(\boldsymbol{y},\boldsymbol{\Lambda}) \tag{41}$$
$$(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}), \partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})) = \partial\mathrm{PD}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon},\boldsymbol{\Lambda}) \tag{42}$$

{Finite Difference Monte Carlo estimators (17) and (19)}

$$\begin{aligned}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) &= \|\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \boldsymbol{y}\right)\|_2^2 \\ &+ \frac{2}{\nu}\left\langle \mathbf{A}^*\boldsymbol{\Pi}\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right), \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle - \mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*)\end{aligned} \tag{43}$$
$$\begin{aligned}\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) &= 2\left(\mathbf{A}\boldsymbol{\Phi}\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right)^*\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \boldsymbol{y}\right) \\ &+ \frac{2}{\nu}\left(\mathbf{A}^*\boldsymbol{\Pi}\left(\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}) - \partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right)\right)^*\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\end{aligned} \tag{44}$$

Output: Risk estimate $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$
Gradient of the risk estimate $\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$

For detailed discussions on low memory implementations of BFGS, box constraints management, and others algorithmic tricks the interested reader is referred to [12, 22, 49].

Convergence conditions for quasi-Newton algorithms relies on the behavior of

Algorithm 3 Automated selection of hyperparameters minimizing quadratic risk.

Inputs: Observations $\boldsymbol{y}$
Covariance matrix $\boldsymbol{\mathcal{S}}$
Monte Carlo vector $\boldsymbol{\varepsilon} \in \mathbb{R}^P \sim \mathcal{N}(\mathbf{0}_P, \boldsymbol{I}_P)$
Finite Difference step $\nu > 0$

Initialization: $\boldsymbol{\Lambda}^{[0]} \in \mathbb{R}^L$, Inverse Hessian $\boldsymbol{H}^{[0]} \in \mathbb{R}^{L \times L}$,
Gradient $\partial_{\boldsymbol{\Lambda}} \widehat{R}^{[0]} = \mathrm{SUGAR}(\boldsymbol{y}, \boldsymbol{\Lambda}^{[0]}, \boldsymbol{\mathcal{S}}, \nu, \boldsymbol{\varepsilon})$

for $t = 0$ **to** $T_{\max} - 1$ **do**

{Descent direction from gradient of the risk estimate:}

$$\boldsymbol{d}^{[t]} = -\boldsymbol{H}^{[t]} \partial_{\boldsymbol{\Lambda}} \widehat{R}^{[t]} \tag{46}$$

{Line search to find descent step:}

$$\alpha^{[t]} \in \underset{\alpha \in \mathbb{R}}{\mathrm{Argmin}}\, \widehat{R}(\boldsymbol{\Lambda}^{[t]} + \alpha \boldsymbol{d}^{[t]}), \quad \text{with } \widehat{R}(\boldsymbol{\Lambda}) = \mathrm{SURE}(\boldsymbol{y}, \boldsymbol{\Lambda}, \boldsymbol{\mathcal{S}}, \nu, \boldsymbol{\varepsilon}) \tag{47}$$

{Quasi-Newton descent step on $\boldsymbol{\Lambda}$:}

$$\boldsymbol{\Lambda}^{[t+1]} = \boldsymbol{\Lambda}^{[t]} + \alpha^{[t]} \boldsymbol{d}^{[t]} \tag{48}$$

{Gradient update:}

$$\partial_{\boldsymbol{\Lambda}} \widehat{R}^{[t+1]} = \mathrm{SUGAR}(\boldsymbol{y}, \boldsymbol{\Lambda}^{[t+1]}, \boldsymbol{\mathcal{S}}, \nu, \boldsymbol{\varepsilon}) \tag{49}$$

{Gradient increment}

$$\boldsymbol{u}^{[t]} = \partial_{\boldsymbol{\Lambda}} \widehat{R}^{[t+1]} - \partial_{\boldsymbol{\Lambda}} \widehat{R}^{[t]} \tag{50}$$

{BFGS update of inverse Hessian (52):}

$$\boldsymbol{H}^{[t+1]} = \mathrm{BFGS}(\boldsymbol{H}^{[t]}, \boldsymbol{d}^{[t]}, \boldsymbol{u}^{[t]}) \tag{51}$$

end for

Outputs: *Finite-time* solution of Problem (45) $\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}|\boldsymbol{\mathcal{S}}) \triangleq \boldsymbol{\Lambda}^{[T_{\max}]}$
Estimate with automated selection of $\boldsymbol{\Lambda}$ $\widehat{\boldsymbol{x}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}|\boldsymbol{\mathcal{S}}) \triangleq \mathrm{PD}(\boldsymbol{y}, \boldsymbol{\Lambda}^{[T_{\max}]})$

second derivatives of the objective function [49]. Most of the time, when it comes to sequential estimators, one has no information about the twice differentiability of generalized SURE with respect to hyperparameters. Hence, the convergence of Algorithm 3 will be assessed numerically. Further, quasi-Newton algorithms being known to be sensitive to initialization, special attention needs to be paid to the initialization of both hyperparameters $\boldsymbol{\Lambda}$ and approximated inverse Hessian

$\boldsymbol{H}$ (see Section 5.2.4).

Remark 9. Given a parametric estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$, possibly obtained by another routine than PD, Algorithms 2 and 3 can be used, provided that one has a routine equivalent to ∂PD, computing $\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$. The reader can find other differentiated proximal algorithms in [24].

4 Hyperparameter tuning for texture segmentation

The formalism proposed above for the automated selection of the regularization hyperparameters is now specified to total-variation based texture segmentation. Section 4.1 formulates the texture segmentation problem as the minimization of a convex objective function. Then, in Section 4.2, this segmentation procedure is cast into the general formalism of Sections 2 and 3. The hypothesis needed to apply Theorems 2 and 3 are discussed one by one in the context of texture segmentation. Finally, the practical evaluation of the estimators of the risk $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ and of the gradient of the risk $\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ is discussed in Section 4.3.

4.1 Total-variation based texture segmentation

4.1.1 Piecewise homogeneous fractal texture model

Let $X \in \mathbb{R}^{N_1\times N_2}$ denote the texture to be segmented, consisting of a real-valued discrete field defined on a grid of pixels $\Omega = \{1,\dots,N_1\}\times\{1,\dots,N_2\}$. Texture X is assumed to be formed as the union of M independent Gaussian textures, existing on a set of disjoint supports,

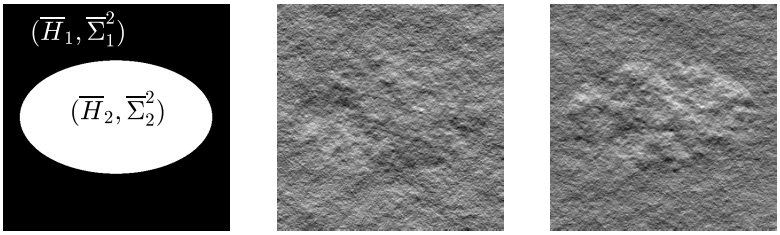
$$\Omega = \overline{\Omega}_1 \cup \cdots \cup \overline{\Omega}_M, \quad \text{with} \quad \overline{\Omega}_m \cap \overline{\Omega}_{m'} = \emptyset \quad \text{if} \quad m \neq m'. \tag{54}$$

Each homogeneous Gaussian texture, defined on Ω_m is characterized by two global fractal features, the scaling (or Hurst) exponent $\overline{H}_m$ and the variance $\overline{\Sigma}^2_m$, that fully control its statistics. Interested readers are referred to e.g., [51] for the detailed definition of Gaussian fractal textures. Figures 1b and 1c propose examples of such piecewise Gaussian fractal textures, with $M = 2$ and mask shown in Figure 1a.

4.1.2 Local regularity and wavelet *leader* coefficients

It was abundantly discussed in the literature (cf. e.g. [48, 54, 70–72]) that textures can be well-analyzed by local fractal features (local regularity and local variance), that can be accurately estimated from wavelet *leader* coefficients, as extensively described and studied in e.g. [56, 72], to which the reader is referred for a detailed presentation.

Let $\chi^{(d)}_{j,\underline{n}}$ denote the coefficients of the undecimated 2D Discrete Wavelet Transform of image X, at octave $j = j_1,\dots,j_2$ and pixel $\underline{n} \in \Omega$, with the 2D-wavelet basis being defined from the 4 combination (hence the orientations $d \in \{0,1,2,3\}$) of 1D wavelet ψ and scaling functions. Interested readers are referred to e.g., [46] for a full definition of the $\chi^{(d)}_{j,\underline{n}}$. Wavelet *leaders*,



(a) Elliptic mask (b) X: Texture "**D**" (c) X: Texture "**E**"

Figure 1: (a) Mask for piecewise textures composed of two regions: "background" (in black) on which the texture is characterized by homogeneous local regularity $\bar{\boldsymbol{h}} \equiv \overline{H}_0$ and local variance $\bar{\boldsymbol{\sigma}}^2 \equiv \overline{\Sigma}_0^2$ and "foreground" (in white) on which the texture is characterized by homogeneous local regularity $\bar{\boldsymbol{h}} \equiv \overline{H}_1$ and local variance $\bar{\boldsymbol{\sigma}}^2 \equiv \overline{\Sigma}_1^2$. (b) and (c) Synthetic piecewise homogeneous textures used for performance assessment, with resolution 256×256 pixels.

$\{\mathcal{L}_{j,\underline{n}},\, j = j_1, \ldots, j_2,\, \underline{n} \in \Omega\}$, are further defined as local suprema over a spatial neighborhood and across all finest scales of the $\chi^{(d)}_{j,\underline{n}}$ [72]:

$$\mathcal{L}_{j,\underline{n}} = \sup_{\substack{d = \{1,2,3\} \\ \lambda_{j',\underline{n}'} \subset 3\lambda_{j,\underline{n}}}} \left| 2^j \chi^{(d)}_{j',\underline{n}'} \right|, \quad \text{where} \quad \left\{ \begin{array}{l} \lambda_{j,\underline{n}} = \big[\underline{n}, \underline{n} + 2^j\big[\,, \\ 3\lambda_{j,\underline{n}} = \bigcup\limits_{\underline{p} \in \{-2^j, 0, 2^j\}^2} \lambda_{j,\underline{n}+\underline{p}}. \end{array} \right. \tag{55}$$

Local regularity $\bar{h}_{\underline{n}}$ and local variance $\bar{\sigma}^2_{\underline{n}}$ at pixel n can be defined via the local power law behavior of the wavelet leaders across scales [71, 72]:

$$\mathcal{L}_{j,\underline{n}} = \bar{\sigma}_{\underline{n}} 2^{j\bar{h}_{\underline{n}}} \beta_{j,\underline{n}}, \quad \text{as } 2^j \to 0, \tag{56}$$

where $\beta_{j,\underline{n}}$ can be well approximated for large classes of textures [70] as log-normal random variables, with log-mean $\mu = 0$. For piecewise fractal textures X described in Section 4.1.1, local regularity $\bar{\boldsymbol{h}} \in \mathbb{R}^{N_1 \times N_2}$ and local variance $\bar{\boldsymbol{\sigma}}^2 \in \mathbb{R}^{N_1 \times N_2}$ maps are piecewise constant, reflecting the global scaling exponent H and variance Σ^2 of the homogeneous textures as:

$$(\forall m \in \{1, \ldots, M\})\ \big(\forall \underline{n} \in \overline{\Omega}_m\big) \quad \bar{h}_{\underline{n}} \equiv \overline{H}_m \text{ and } \bar{\sigma}^2_{\underline{n}} \equiv \overline{\Sigma}^2_m F(\overline{H}_m, \psi), \tag{57}$$

with $F(\overline{H}_m, \psi)$ a deterministic function studied in [68] and not of interest here. Taking the logarithm of Equation (56) leads to the following linear formulation

$$\ell_{j,\underline{n}} = \bar{v}_{\underline{n}} + j\bar{h}_{\underline{n}} + \zeta_{j,\underline{n}}, \quad \text{as } 2^j \to 0 \tag{58}$$

with log-*leaders* $\ell_{j,\underline{n}} = \log_2(\mathcal{L}_{j,\underline{n}})$, log-variance $\bar{v}_{\underline{n}} = \log_2 \bar{\sigma}_{\underline{n}}$ and zero-mean Gaussian noise $\zeta_{j,\underline{n}} = \log_2(\beta_{j,\underline{n}})$. In the following, the *leader* coefficients at scale 2^j are denoted $\boldsymbol{\ell}_j \in \mathbb{R}^{N_1 N_2}$, and the complete collection of *leaders* is stored in $\boldsymbol{\ell} \in \mathbb{R}^{J N_1 N_2}$.

4.1.3 Total variation regularization and iterative thresholding

The linear regression estimator inspired by (58)

$$\begin{pmatrix}\widehat{\boldsymbol{h}}_{\mathrm{LR}}(\boldsymbol{\ell})\\ \widehat{\boldsymbol{v}}_{\mathrm{LR}}(\boldsymbol{\ell})\end{pmatrix} = \underset{\binom{\boldsymbol{h}}{\boldsymbol{v}}\in\mathbb{R}^{2N_1N_2}}{\operatorname{argmin}} \sum_{j=j_1}^{j_2} \|j\boldsymbol{h}+\boldsymbol{v}-\boldsymbol{\ell}_j\|_2^2 \tag{59}$$

achieves poor performance in estimating piecewise constant local regularity and local power, hence precluding an accurate segmentation of the piecewise homogeneous textures. Thus, a functional for *joint* attribute estimation and segmentation was proposed in [51], leading to the following Penalized Least Squares (3):

$$\begin{pmatrix}\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})\\ \widehat{\boldsymbol{v}}(\boldsymbol{\ell};\boldsymbol{\Lambda})\end{pmatrix} \in \underset{\binom{\boldsymbol{h}}{\boldsymbol{v}}\in\mathbb{R}^{2N_1N_2}}{\operatorname{Argmin}} \sum_{j=j_1}^{j_2} \|j\boldsymbol{h}+\boldsymbol{v}-\boldsymbol{\ell}_j\|_2^2 + \lambda_h \mathrm{TV}(\boldsymbol{h}) + \lambda_v \mathrm{TV}(\boldsymbol{v}), \tag{60}$$

where TV stands for the well-known isotropic Total Variation, defined as a mixed $\ell_{2,1}$-norm composed with spatial gradient operators

$$\mathrm{TV}(\boldsymbol{h}) = \sum_{\underline{n}\in\Omega} \sqrt{(\mathbf{D}_1\boldsymbol{h})_{\underline{n}}^2 + (\mathbf{D}_2\boldsymbol{h})_{\underline{n}}^2} = \sum_{\underline{n}\in\Omega} \|(\mathbf{D}\boldsymbol{h})_{\underline{n}}\|_2, \tag{61}$$

where $\mathbf{D}_1 : \mathbb{R}^{N_1N_2} \to \mathbb{R}^{N_1N_2}$ (resp. $\mathbf{D}_2 : \mathbb{R}^{N_1N_2} \to \mathbb{R}^{N_1N_2}$) stand for the discrete spatial horizontal (resp. vertical) gradient operator. This TV-penalized least square estimator is designed to favor piecewise constancy of the estimates $\widehat{\boldsymbol{h}}$ and $\widehat{\boldsymbol{v}}$, making used of ℓ_1-norm, i.e. $q = 1$ in (3).
Finally, following [14,15], the estimate $\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ is *thresholded* to yield a posterior *piecewise constant* map of local regularity $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$, taking *exactly* M different values $\widehat{H}_1(\boldsymbol{\ell};\boldsymbol{\Lambda}),\ldots,\widehat{H}_M(\boldsymbol{\ell};\boldsymbol{\Lambda})$. The resulting segmentation

$$\Omega = \widehat{\Omega}_1(\boldsymbol{\ell};\boldsymbol{\Lambda}) \cup \cdots \cup \widehat{\Omega}_M(\boldsymbol{\ell};\boldsymbol{\Lambda}) \tag{62}$$

is deduced from $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$, defining

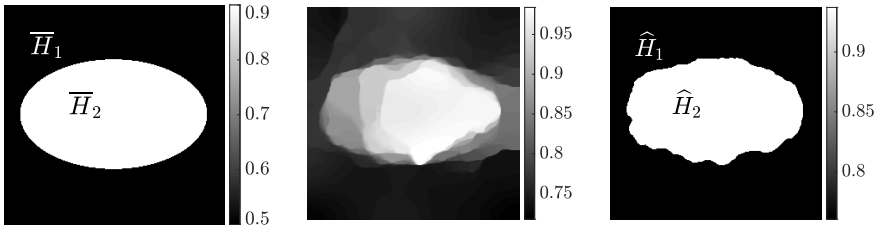
$$(\forall m \in \{1,\ldots,M\})\,,\ \widehat{\Omega}_m(\boldsymbol{\ell};\boldsymbol{\Lambda}) = \left\{ \underline{n}\in\Omega \,\middle|\, \left(T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})\right)_{\underline{n}} \equiv \widehat{H}_m(\boldsymbol{\ell};\boldsymbol{\Lambda}) \right\}. \tag{63}$$

This is illustrated in Figure 2, for a two-region synthetic texture with ground truth piecewise constant local regularity $\bar{\boldsymbol{h}}$ in Figure 2a.

4.2 Reformulation in term of Model (1)

4.2.1 Observation $\boldsymbol{y}$

To cast the log-linear behavior (58) into the general model (1), vectorized quantities for $\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{h}}$ and $\boldsymbol{\ell}$ are used. The $N_1 \times N_2$ maps $\bar{\boldsymbol{h}}$ and $\bar{\boldsymbol{v}}$ are reshapped into vectors $\bar{\boldsymbol{x}} \in \mathbb{R}^N$, with $N = 2N_1N_2$, ordering the pixels in the lexicographic order. The log-*leaders* $\boldsymbol{\ell} = (\boldsymbol{\ell}_j)_{j_1\le j\le j_2}$, composed of $J \triangleq j_2 - j_1 + 1$ octaves of resolution $N_1 \times N_2$ are vectorized, octaves by octaves, with lexical ordering of pixels, $\boldsymbol{\ell} \in \mathbb{R}^P$, with $P = JN_1N_2$.



(a) Ground truth $\bar{\boldsymbol{h}}$ (b) Estimate $\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ (c) Thresholded $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$

Figure 2: Example of the thresholding of $\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ to obtain a two-region segmentation where $\boldsymbol{\ell}$ denotes the wavelet *learders* associated with the Texture X: "**E**" displayed in Figure 1c

Equation (58) can then be cast into general model (1) as:

$$\text{Observations} \qquad \boldsymbol{y} = \boldsymbol{\ell} \in \mathbb{R}^P, \qquad P = JN_1N_2 \tag{64}$$

$$\text{Ground truth} \qquad \bar{\boldsymbol{x}} = \begin{pmatrix} \bar{\boldsymbol{h}} \\ \bar{\boldsymbol{v}} \end{pmatrix} \in \mathbb{R}^N, \qquad N = 2N_1N_2 \tag{65}$$

$$\text{Linear degradation} \qquad \boldsymbol{\Phi} : \left\{ \begin{array}{l} \mathbb{R}^N \to \mathbb{R}^P \\ \begin{pmatrix} \bar{\boldsymbol{h}} \\ \bar{\boldsymbol{v}} \end{pmatrix} \mapsto \left(j\bar{\boldsymbol{h}} + \bar{\boldsymbol{v}} \right)_{j_1 \le j \le j_2} . \end{array} \right. \tag{66}$$

4.2.2 Full-rank operator $\boldsymbol{\Phi}$

Proposition 3 asserts that $\boldsymbol{\Phi}^*\boldsymbol{\Phi}$ is invertible (Assumption 2).

Proposition 3. *The linear operator $\boldsymbol{\Phi}$ defined in* (66) *is bounded and its adjoint writes*

$$\boldsymbol{\Phi}^* : \left\{ \begin{array}{l} \mathbb{R}^P \to \mathbb{R}^N \\ \\ (\boldsymbol{\ell}_j)_{j_1 \le j \le j_2} \mapsto \begin{pmatrix} \sum_{j=j_1}^{j_2} j\boldsymbol{\ell}_j \\ \\ \sum_{j=j_1}^{j_2} \boldsymbol{\ell}_j \end{pmatrix} \end{array} \right. \tag{67}$$

Further, $\boldsymbol{\Phi}$ is full rank, and the following inversion formula holds

$$(\boldsymbol{\Phi}^*\boldsymbol{\Phi})^{-1} = \frac{1}{F_2F_0 - F_1^2} \begin{pmatrix} F_0\boldsymbol{I}_{N/2} & -F_1\boldsymbol{I}_{N/2} \\ -F_1\boldsymbol{I}_{N/2} & F_2\boldsymbol{I}_{N/2} \end{pmatrix}, \; F_\alpha \triangleq \sum_{j=j_1}^{j_2} j^\alpha, \; \alpha \in \{0,1,2\}. \tag{68}$$

Proof. Formula (67) is obtained from straightforward computations. Then, combining (66) and (67), leads to

$$\boldsymbol{\Phi}^*\boldsymbol{\Phi} = \begin{pmatrix} F_2\boldsymbol{I}_{N/2} & F_1\boldsymbol{I}_{N/2} \\ F_1\boldsymbol{I}_{N/2} & F_0\boldsymbol{I}_{N/2} \end{pmatrix} \tag{69}$$

which is finally inverted using the 2×2 cofactor matrix formula. □

4.2.3 Projection operator

Performing texture segmentation the discriminant attribute is the local regularity $\boldsymbol{h}$, while local power $\boldsymbol{v}$ is an auxiliary feature. Hence the *projected* quadratic risk (10) customized to texture segmentation reads:

$$R[\widehat{\boldsymbol{h}}](\boldsymbol{\Lambda}) \triangleq \mathbb{E}_{\boldsymbol{\zeta}} \left\| \widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda}) - \bar{\boldsymbol{h}} \right\|_2^2, \tag{70}$$

with $\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ defined in (60).

Then, the particularized projection operator in Definition 1 takes the matrix form

$$\boldsymbol{\Pi} \triangleq \begin{pmatrix} \boldsymbol{I}_{N/2} & \boldsymbol{Z}_{N/2} \\ \boldsymbol{Z}_{N/2} & \boldsymbol{Z}_{N/2} \end{pmatrix} \quad \text{so that} \quad \boldsymbol{\Pi} \begin{pmatrix} \bar{\boldsymbol{h}} \\ \bar{\boldsymbol{v}} \end{pmatrix} = \begin{pmatrix} \bar{\boldsymbol{h}} \\ \boldsymbol{0}_{N/2} \end{pmatrix} \tag{71}$$

where $\boldsymbol{I}_{N/2}$ (resp. $\boldsymbol{Z}_{N/2}$) denotes the identity (resp. null) matrix of size $N/2 \times N/2$ and $\boldsymbol{0}_{N/2}$ the null vector of $\mathbb{R}^{N/2}$.

4.2.4 Regularity of the estimates

Proposition 4. *Problem* (60) *has a unique solution* $\begin{pmatrix} \widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda}) \\ \widehat{\boldsymbol{v}}(\boldsymbol{\ell};\boldsymbol{\Lambda}) \end{pmatrix}$.
This solution is continuous and weakly differentiable w.r.t. $\boldsymbol{\ell}$ *and integrable against the Gaussian probability density function (Assumption 3). Further, both* $\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ *and* $\widehat{\boldsymbol{v}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ *are uniformly* L_1*-Lipschitz w.r.t.* $\boldsymbol{\ell}$ *(Assumption 4).*

Proof. As shown in [51], the objective function

$$(\boldsymbol{h},\boldsymbol{v}) \mapsto \underbrace{\sum_{j=j_1}^{j_2} \|j\boldsymbol{h} + \boldsymbol{v} - \boldsymbol{\ell}_j\|_2^2}_{\left\| \boldsymbol{\Phi} \begin{pmatrix} \boldsymbol{h} \\ \boldsymbol{v} \end{pmatrix} - \boldsymbol{\ell} \right\|_2^2} + \lambda_h \mathrm{TV}(\boldsymbol{v}) + \lambda_v \mathrm{TV}(\boldsymbol{v}) \tag{72}$$

is convex, being the sum of convex terms. Further, computing the eigenvalues of $\boldsymbol{\Phi}^*\boldsymbol{\Phi}$ shows that the least squares data fidelity term is γ-strongly convex, with

$$\gamma = 2 \min \mathrm{Sp}(\boldsymbol{\Phi}^*\boldsymbol{\Phi}) > 0 \tag{73}$$

where $\mathrm{Sp}(\boldsymbol{\Phi}^*\boldsymbol{\Phi})$ stand for the spectrum of the (bounded) linear operator $\boldsymbol{\Phi}^*\boldsymbol{\Phi}$. Hence, the objective function (72) has a unique minimum, being the unique solution of Problem (60), as mentioned in Remark 1.
Further, (60) falls under the general formulation of Penalized Least Squares (3), which can be written

$$\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{H}} \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\boldsymbol{\mathcal{W}}}^2 + \mathcal{J}_{\boldsymbol{\Lambda}}(\boldsymbol{x}), \tag{74}$$

where $\mathcal{J}_{\boldsymbol{\Lambda}}(\boldsymbol{x}) = \|\mathbf{U}_{\boldsymbol{\Lambda}}\boldsymbol{x}\|_1$ is built from a linear operator $\mathbf{U}_{\boldsymbol{\Lambda}}$ depending on regularization parameters λ_h and λ_v as

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_h \\ \lambda_v \end{pmatrix} \in \mathbb{R}_+^2, \quad \text{and} \quad \mathbf{U}_{\boldsymbol{\Lambda}} = \left\{ \begin{array}{l} \mathbb{R}^{2N_1N_2} \to \mathbb{R}^{2N_1N_2} \times \mathbb{R}^{2N_1N_2} \\ \begin{pmatrix} \boldsymbol{h} \\ \boldsymbol{v} \end{pmatrix} \mapsto \left(\begin{pmatrix} \lambda_h \mathbf{D}_1 \boldsymbol{h} \\ \lambda_v \mathbf{D}_1 \boldsymbol{v} \end{pmatrix}, \begin{pmatrix} \lambda_h \mathbf{D}_2 \boldsymbol{h} \\ \lambda_v \mathbf{D}_2 \boldsymbol{v} \end{pmatrix} \right). \end{array} \right. \tag{75}$$

$\mathcal{J}_{\boldsymbol{\Lambda}}$ is convex, proper and lower semicontinuous, then, following [67], (74) can be rewritten as a constrained optimization problem

$$(\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}),\widehat{\boldsymbol{z}}(\boldsymbol{y};\boldsymbol{\Lambda})) = \underset{\boldsymbol{x}\in\mathcal{H},\boldsymbol{z}\in\mathcal{G}}{\operatorname{argmin}} \|\boldsymbol{y}-\boldsymbol{z}\|^2_{\boldsymbol{\mathcal{W}}} + \mathcal{J}_{\boldsymbol{\Lambda}}(\boldsymbol{x}), \text{ such that } \boldsymbol{z}=\boldsymbol{\Phi}\boldsymbol{x} \tag{76}$$

$$\iff \quad \widehat{\boldsymbol{z}}(\boldsymbol{y};\boldsymbol{\Lambda}) = \underset{\boldsymbol{z}\in\mathcal{G}}{\operatorname{argmin}} \|\boldsymbol{y}-\boldsymbol{z}\|^2_{\boldsymbol{\mathcal{W}}} + (\boldsymbol{\Phi}\mathcal{J}_{\boldsymbol{\Lambda}})(\boldsymbol{z}) \tag{77}$$

$$\iff \quad \widehat{\boldsymbol{z}}(\boldsymbol{y};\boldsymbol{\Lambda}) = \operatorname{prox}_{(1/2)\boldsymbol{\Phi}\mathcal{J}_{\boldsymbol{\Lambda}}}(\boldsymbol{y}) \tag{78}$$

where

$$(\boldsymbol{\Phi}\mathcal{J}_{\boldsymbol{\Lambda}})(\boldsymbol{z}) \triangleq \min_{\{\boldsymbol{x}|\boldsymbol{\Phi}\boldsymbol{x}=\boldsymbol{z}\}} \mathcal{J}_{\boldsymbol{\Lambda}}(\boldsymbol{x}) \tag{79}$$

denotes the pre-image of $\mathcal{J}_{\boldsymbol{\Lambda}}$ under $\boldsymbol{\Phi}$, which is as well convex, proper and lower semicontinuous.
Then, from (78), the estimator $\widehat{\boldsymbol{z}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is non expansive, i.e. 1-Lipschitz, because the proximal operators share that same property. Moreover, from (76), $\widehat{\boldsymbol{z}}(\boldsymbol{y};\boldsymbol{\Lambda}) = \boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$, and since $\boldsymbol{\Phi}$ is full-rank according to Proposition 3

$$\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) = (\boldsymbol{\Phi}^*\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^*\widehat{\boldsymbol{z}}(\boldsymbol{y};\boldsymbol{\Lambda}). \tag{80}$$

$\boldsymbol{\Phi}$ being bounded, we conclude that the estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is uniformly L_1-Lipschitz, with $L_1 = \|\boldsymbol{\Phi}\|^{-1}$ justifying Assumption 4, *(i)*.
Being uniformly L_1-Lipschitz, $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is continuous and weakly-differentiable (see Theorem 5 of Section 4.2.3 in [32]). As a consequence, both $\langle \mathbf{A}^*\boldsymbol{\Pi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}),\boldsymbol{\zeta}\rangle$ and $\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ are integrable against the Gaussian density and Assumption 3 holds.
Finally, setting $\boldsymbol{y}=\mathbf{0}_P$, for any $\boldsymbol{\Lambda}\in\mathbb{R}^L$, $\widehat{\boldsymbol{x}}(\mathbf{0}_P;\boldsymbol{\Lambda})=\mathbf{0}_N$ reaches the minimum. The solution being unique from Proposition 3, $\mathbf{0}_N$ is the unique solution and Assumption 4, *(ii)* is verified.
Further, it is reasonable to expect that the uniform Lipschitzianity with respect to hyperparameters results of Remark 5, extend to the estimator $\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$, defined in (60). Yet, to the best of our knowledge, no direct proof that Lipschitzianity Assumption 5 holds for general Penalized Least Squares exists. This issue is a scientific question in itself and will be addressed in future work. □

4.3 Practical computation of $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}$ and $\partial_{\boldsymbol{\Lambda}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}$

This section addresses all technical issues encountered in running Algorithm 2, in the context of texture segmentation described above.

4.3.1 Covariance structure of the observations

The additive noise $\zeta_{j,\underline{n}}$ appearing in Equation (58) being Gaussian, Gaussianity Assumption 1 holds. The covariance matrix $\boldsymbol{\mathcal{S}}$ of noise $\boldsymbol{\zeta}$ reads

$$\mathcal{S}^{j',\underline{n}'}_{j,\underline{n}} \triangleq \mathbb{E}\,\zeta_{j,\underline{n}}\zeta_{j',\underline{n}'} = \mathcal{C}^{j'}_{j}\,\Xi^{j'}_{j}(\underline{n}-\underline{n}'), \tag{81}$$

where

$$\mathcal{C}^{j'}_{j} \triangleq \mathbb{E}\,\zeta_{j,\underline{n}}\zeta_{j',\underline{n}}, \quad \mathcal{C}^{j'}_{j} \text{ independent of } \underline{n} \tag{82}$$

quantifies the inter-scale covariance, and $\boldsymbol{\Xi}^{j'}_{j}$ encapsulate the stationary spatial correlations, with correlation length proportional to $\max(2^j, 2^{j'})$.

4.3.2 Matrix product $\boldsymbol{\mathcal{S}\varepsilon}$

Following Remark 4, in general, the direct product $\boldsymbol{\mathcal{S}\varepsilon}$ required for the practical evaluation of Finite Difference Monte Carlo SURE (17) is intractable because of the large size of matrix $\boldsymbol{\mathcal{S}}$. Yet, in the case of log-*leaders*, the spatial correlations presenting the Toeplitz structure (81), the product $\boldsymbol{\mathcal{S}\varepsilon}$ can be computed efficiently, making (17) usable in practice.
Indeed, given $\boldsymbol{\varepsilon} \in \mathbb{R}^{JN_1N_2} = (\boldsymbol{\varepsilon}_j)_{j=j_1}^{j_2} J$, with $\boldsymbol{\varepsilon}_j \in \mathbb{R}^{N_1N_2}$,

$$(\boldsymbol{\mathcal{S}\varepsilon})_{j,\underline{n}} = \sum_{j'=j_1}^{j_2} \sum_{\underline{n}' \in \Omega} \mathcal{S}_{j,\underline{n}}^{j',\underline{n}'} \varepsilon_{j',\underline{n}'} = \sum_{j'=j_1}^{j_2} \mathcal{C}_j^{j'} \sum_{\underline{n}' \in \Omega} \Xi_j^{j'}(\underline{n} - \underline{n}') \varepsilon_{j',\underline{n}'} = \sum_{j'=j_1}^{j_2} \mathcal{C}_j^{j'} \boldsymbol{\Xi}_{j,j'} * \boldsymbol{\varepsilon}_j. \tag{83}$$

which is the sum of J convolution products, denoted $*$, of high dimensional vector $\boldsymbol{\varepsilon} \in \mathbb{R}^{N_1N_2}$ with "low dimensional" finite support window $\mathcal{C}_j^{j'} \boldsymbol{\Xi}_{j,j'}$. Hence evaluating $\boldsymbol{\mathcal{S}\varepsilon}$ appears to be far less costly than a general product of matrix of size $P \times P$ by a vector of size P.

4.3.3 Operator A

In the same vein, the matrices $\boldsymbol{\Phi}^*$ (67), $(\boldsymbol{\Phi}^*\boldsymbol{\Phi})^{-1}$ (68) and $\boldsymbol{\Pi}$ (71) turn out to be very sparse, since they act independently on each pixel. Thus, the same sparse (pixel-wise) structure follows for

$$\mathbf{A} = \boldsymbol{\Pi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^* = \frac{1}{F_0F_2 - F_1^2} \begin{pmatrix} (F_0 + j_1F_1)\boldsymbol{I}_{N/2} & \cdots & (F_0 + j_2F_1)\boldsymbol{I}_{N/2} \\ \boldsymbol{Z}_{N/2} & \cdots & \boldsymbol{Z}_{N/2} \end{pmatrix}. \tag{84}$$

Hence the products $\mathbf{A}^*\boldsymbol{\Pi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ and $\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$, appearing in the Finite Difference Monte Carlo risk (17) and gradient of the risk (19) estimators, are very cheap to compute, involving $\mathcal{O}(N)$ operations.

4.3.4 Evaluation of $\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*)$

The evaluation of risk estimate $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, at Step (40) of generalized SURE and SUGAR Algorithm 2 requires the computation of the trace of an $N \times N$ matrix, with N possibly of order 10^6, e.g. in image processing.
In the present application, combining the structure of covariance matrix $\boldsymbol{\mathcal{S}}$ (81) and the sparse expression of $\mathbf{A}$ (84), provides a compact expression of the third term of generalized SURE (17) detailed in Proposition 5, which can be evaluated with very little computational effort.

Proposition 5 (Third term of Stein Unbiased Risk Estimate)**.** *Consider texture's leader coefficients* (58)*, whose covariance matrix* $\boldsymbol{\mathcal{S}}$ *evidences the sparse structure described in* (81)*. Define the linear operator* $\boldsymbol{A}$ *from Formula* (9)*, using operator* $\boldsymbol{\Phi}$ (66) *and projector* $\boldsymbol{\Pi}$ (71)*. Then, the third term of Stein estimator of the risk* (17) *reads*

$$\mathrm{Tr}(\boldsymbol{A\mathcal{S}A}^*) = \frac{N/2}{\left(F_0F_2 - F_1^2\right)^2} \left(\sum_{j,j'} \left(F_1^2 \mathcal{C}_j^{j'} - 2F_0F_1 j' \mathcal{C}_j^{j'} + F_0^2 jj' \mathcal{C}_j^{j'} \right) \right), \tag{85}$$

where the quantities $\{F_\alpha,\, \alpha = 0, 1, 2\}$ *are defined in* (68) *and* $\mathcal{C}_j^{j'}$ *denotes the covariance between scales* 2^j *and* $2^{j'}$*, as defined in* (82).

Proof. Proof is postponed to Appendix D. □

5 Hyperparameter tuning performance assessment

The aim of this section is to assess quantitatively, by means of numerical simulations, the performance in the estimation of the optimal hyperparamaters. To that end, Section 5.1 will detail the numerical simulation set-up and Section 5.2 will concentrate on several algorithmic issues. Section 5.3 will show on the prominent role of covariance matrix $\boldsymbol{\mathcal{S}}$, evaluating the impact of partial vs. full covariance matrix in Section 5.3.2 and comparing true vs. estimated covariance matrix in Section 5.3.3. Section 5.4 will further assess quantitatively how well optimal hyperparameters are estimated in the absence of available ground truth, with respect to different quality metrics.

5.1 Numerical simulation set-up

5.1.1 Textures

For sake of simplicity, we consider the two-region case $M = 2$, with elliptic mask displayed in Figure 1a. Synthetic textures of resolution $N_1 \times N_2 = 256 \times 256$, characterized by two attributes configurations:

- Configuration "**D**", "difficult", one realization being displayed in Figure 1b

$$\begin{aligned}\left(\overline{H}_1, \overline{\Sigma}_1^2\right) &= (0.5, 0.6) \quad \text{(background)},\\ \left(\overline{H}_2, \overline{\Sigma}_2^2\right) &= (0.75, 0.7) \quad \text{(central ellipse)}.\end{aligned}$$

- Configuration "**E**", "easy", one realization being displayed in Figure 1c

$$\begin{aligned}\left(\overline{H}_1, \overline{\Sigma}_1^2\right) &= (0.5, 0.6) \quad \text{(background)},\\ \left(\overline{H}_2, \overline{\Sigma}_2^2\right) &= (0.9, 1.1) \quad \text{(central ellipse)}.\end{aligned}$$

are generated from a Matlab routine designed by ourselves (see [51]).

5.1.2 Multiscale analysis

A 2D undecimated wavelet transform of the textured image is computed at scale 2^j, with mother wavelet obtained as a tensor product of 1D least asymmetric Daubechies wavelets, with 3 vanishing moments, see [46] for more details.

5.1.3 Performance evaluation

Following [51, 52], for a given textured image X, and the derived log-*leaders* $(\boldsymbol{\ell}_j)_{j=j_1}^{j_2}$, two performance indices are used:

- The *one-sample quadratic risk* on local regularity, computed from *one sample* of log-*leaders* $\boldsymbol{\ell}$ computed on the single image X

$$\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda}) \triangleq \left\| \widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda}) - \bar{\boldsymbol{h}} \right\|_2^2, \tag{86}$$

  with estimator $\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ defined in (60) and ground truth $\bar{\boldsymbol{h}}$ defined in (57).

- The *segmentation error*, defined as the percentage of incorrectly classified pixels

$$\mathcal{P}(\boldsymbol{\ell};\boldsymbol{\Lambda}) \triangleq \left| \overline{\Omega}_1 \cap \widehat{\Omega}_2(\boldsymbol{\ell};\boldsymbol{\Lambda}) \right| + \left| \widehat{\Omega}_1(\boldsymbol{\ell};\boldsymbol{\Lambda}) \cap \overline{\Omega}_2 \right|, \tag{87}$$

  where $\cup_m \widehat{\Omega}_m(\boldsymbol{\ell}\boldsymbol{\Lambda})$ is the estimated partition (62), obtain from TV-based texture segmentation, as described in Section 4.1.3.

Remark 10. By definition of the quadratic risk (70) and one-sample quadratic risk (86), $\mathbb{E}_{\boldsymbol{\zeta}}\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda}) = R[\widehat{\boldsymbol{h}}](\boldsymbol{\Lambda})$. In practice however, only one realization of $\boldsymbol{\ell}$ is available, hence the quadratic risk $R[\widehat{\boldsymbol{h}}](\boldsymbol{\Lambda})$ is not accessible. Thus, in the following experiments, the *one-sample quadratic risk* $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$, defined in (86), is used as a reference to which Stein risk estimator $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ will be compared.

5.2 Algorithmic set-up

5.2.1 Primal dual with iterative differentiation

Problem (60) is solved using the accelerated primal-dual algorithm 1, with primal variable $\boldsymbol{x} \triangleq (\boldsymbol{h}, \boldsymbol{v})$, taking advantage of strong-convexity of the data fidelity term. The maximal number of iterations is set to $K_{\max} = 5\,10^5$, and a threshold on the normalized duality gap is set to 10^{-4} (see [51]).

5.2.2 Scaling range

The estimation of piecewise constant local attributes requires to focus on fine scales. Thus, ideally, the least square term (59) would involve the two finest scales of the multiscale representation, and range from $j_1 = 1$ to $j_2 = 2$. Yet, the efficiency of acceleration strategy of Algorithm 1 increases with the strong-convexity modulus γ (73), displayed in Table 1, which is observed to increase with j_2, as $j_1 = 1$ is fixed. Thus, a trade-off between locality and convergence speed leads to select $j_2 = 3$.

5.2.3 Finite Difference Monte Carlo parameters

The Monte Carlo vector $\boldsymbol{\varepsilon} \in \mathbb{R}^P$, $P = JN_1N_2$, is drawn randomly, according to a i.i.d. normalized Gaussian $\mathcal{N}(\mathbf{0}_P, \boldsymbol{I}_P)$. We adapt the heuristic of [24] or the Finite Difference step ν to the case of correlated noise as

$$\nu = \frac{2}{P^\alpha} \max\left(\sqrt{C_j^j},\, j \in \{1,\dots,J\} \right), \quad \alpha = 0.3, \tag{88}$$

	$j_2 = 2$	$j_2 = 3$	$j_2 = 4$	$j_2 = 5$	$j_2 = 6$
γ	0.29	**0.72**	1.20	1.69	2.20

Table 1: Strong-convexity modulus γ of data-fidelity term of (72), computed from Formula (73), for fixed $j_1 = 1$ and varied j_2. The bold entry correspond to the range of scales used in the experiments of Sections 5.

where C_j^j is the variance of the log-*leaders* $\boldsymbol{\ell}_j$ at scale 2^j.
The derivatives with respect to hyperparameters of the estimates, $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{h}}$, $\partial_{\boldsymbol{\Lambda}} \widehat{\boldsymbol{v}}$, are obtained by iterative differentiation of primal dual algorithm, customized to texture segmentation in Appendix 1.

5.2.4 BFGS quasi-Newton initialization and parameters

To perform the risk minimization sketched in Algorithm 3, we used the GRadient-based Algorithm for Non-Smooth Optimization, implemented in GRANSO toolbox[3], from the BFGS quasi-Newton algorithm proposed in [22]. It consists of a low memory BFGS algorithm with box constraints, enabling to enforce positive λ_h and λ_v. The maximal number of iterations of BFGS Algorithm 3 is set to $K_{\max} = 250$, while the stopping criterion on the gradient norm is set to 10^{-6}.
As mentioned in Section 3.3, the initialization of quasi-Newton algorithms might drastically impact their convergence. Hence, we propose a model-based strategy for initializing $\boldsymbol{\Lambda}$ and $\boldsymbol{H}$. The initialization of λ_h and λ_v is performed by balancing the data fidelity term and the penalization appearing of functional (72). The data fidelity term grows like the variance of the noise

$$\mathbb{E} \sum_{j=j_1}^{j_2} \| j\bar{\boldsymbol{h}} + \bar{\boldsymbol{v}} - \boldsymbol{\ell}_j \|_2^2 = \mathrm{tr}(\boldsymbol{\mathcal{S}}), \tag{89}$$

and the penalization term can be evaluated using $\left(\widehat{\boldsymbol{h}}_{\mathrm{LR}}, \widehat{\boldsymbol{v}}_{\mathrm{LR}}\right)$ introduced in (59). Thus, the initial hyperparameters $\boldsymbol{\Lambda}$ for BFGS Algorithm 3 are set to

$$\boldsymbol{\Lambda}^{[0]} = \left(\lambda_h^{[0]}, \lambda_v^{[0]}\right), \quad \text{where} \quad \lambda_h^{[0]} = \frac{\mathrm{tr}(\boldsymbol{\mathcal{S}})}{2\,\mathrm{TV}(\widehat{\boldsymbol{h}}_{\mathrm{LR}}(\boldsymbol{\ell}))}, \text{ and } \lambda_v^{[0]} = \frac{\mathrm{tr}(\boldsymbol{\mathcal{S}})}{2\,\mathrm{TV}(\widehat{\boldsymbol{v}}_{\mathrm{LR}}(\boldsymbol{\ell}))}. \tag{90}$$

The *inverse* Hessian matrix $\boldsymbol{H}^{[0]} \in \mathbb{R}^{2\times 2}$, is initialized to enforce $\boldsymbol{\Lambda}^{[1]} = (1 \pm \kappa)\boldsymbol{\Lambda}^{[0]}$. It is chosen diagonal with coefficients

$$\boldsymbol{H}^{[0]} = \mathrm{diag}\left(\left| \frac{\kappa \lambda_h^{[0]}}{\partial_{\lambda_h} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}; \boldsymbol{\Lambda}^{[0]} | \boldsymbol{\mathcal{S}})} \right|, \left| \frac{\kappa \lambda_v^{[0]}}{\partial_{\lambda_v} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}; \boldsymbol{\Lambda}^{[0]} | \boldsymbol{\mathcal{S}})} \right| \right). \tag{91}$$

In practice, we used $\kappa = 0.5$ for all experiments. It is observed that this choice of $\boldsymbol{H}^{[0]}$ avoids the first iteration falling away from natural hyperpamaters scaling (90), which would induce huge computational cost to reach to optimal hyperparameters.

[3] `http://www.timmitchell.com/software/GRANSO/`

5.3 Covariance of *leaders*

5.3.1 Covariance estimation procedure

No closed-form formula exists to compute exactly the covariance matrix $\boldsymbol{\mathcal{S}}$ from the texture's attributes. Hence, from *one* sample $\boldsymbol{\ell}$, computed from a single texture X, the *estimated* covariance matrix, denoted $\widehat{\boldsymbol{\mathcal{S}}}$, is computed using classic sample covariance estimator:

$$\widehat{\mathcal{S}}_{j,\underline{n}}^{j',\underline{n}'} \triangleq \frac{1}{|\Omega|}\sum_{\underline{n}\in\Omega} \ell_{j,\underline{n}}\ell_{j',\underline{n}+\delta\underline{n}} - \left(\frac{1}{|\Omega|}\sum_{\underline{n}\in\Omega} \ell_{j,\underline{n}}\right)\left(\frac{1}{|\Omega|}\sum_{\underline{n}\in\Omega} \ell_{j',\underline{n}}\right), \tag{92}$$

for spatial lag $\delta\underline{n} \triangleq \underline{n}' - \underline{n}$, leading to inter-scale covariance

$$\widehat{\mathcal{C}}_j^{j'} = \widehat{\mathcal{S}}_{j,\underline{n}}^{j',\underline{n}} \tag{93}$$

and spatial correlations

$$\widehat{\Xi}_j^{j'}(\delta\underline{n}) = \frac{\widehat{\mathcal{S}}_{j,\underline{n}}^{j',\underline{n}'}}{\widehat{\mathcal{C}}_j^{j'}}. \tag{94}$$

Then, for Textures "**D**" and "**E**", a *true* covariance matrix $\boldsymbol{\mathcal{S}}$ is obtained numerically by averaging the above estimated covariance matrix $\widehat{\boldsymbol{\mathcal{S}}}^{(q)}$ over $Q = 5000$ texture samples as:

$$\boldsymbol{\mathcal{S}} \triangleq \left\langle \widehat{\boldsymbol{\mathcal{S}}}^{(q)} \right\rangle_{q=1}^{Q}, \tag{95}$$

the samples being generated with the mathematical model of [51].

5.3.2 Impact of partial versus full covariance on estimated risk

We now assess the impact of using two partial versions of the *full true* covariance matrix $\boldsymbol{\mathcal{S}}$, described in (95):

1. *Variance* matrix $\boldsymbol{\mathcal{S}}_{\rm var}$ neglecting both inter-scale and spatial correlations, reduces to the variances $\mathcal{C}_j^j$ of the $\boldsymbol{\ell}_j$'s, and hence is diagonal

$$S_{\rm var}{}_{j,\underline{n}}^{j',\underline{n}'} = \mathcal{C}_j^j \delta_{j,j'}\delta_{\underline{n},\underline{n}'}. \tag{96}$$

2. *Inter-scale* covariance matrix $\boldsymbol{\mathcal{S}}_{\rm int}$, neglecting spatial correlations, reduces to cross-correlations $\mathcal{C}_j^{j'}$ between the $\boldsymbol{\ell}_j$'s and the $\boldsymbol{\ell}_{j'}$'s at same location

$$S_{\rm int}{}_{j,\underline{n}}^{j',\underline{n}'} = \mathcal{C}_j^{j'}\delta_{\underline{n},\underline{n}'}. \tag{97}$$

For texture "**D**", both $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}_{\rm var})$ (Fig. 3e) and $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}_{\rm int})$ (Fig. 3i) fail to reproduce $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ (Fig. 3a). Hence, the selected hyperparameters $\widehat{\boldsymbol{\Lambda}}_{\nu,\boldsymbol{\varepsilon}}^{\dagger}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}_{\rm var})$ ('$\square$') and $\widehat{\boldsymbol{\Lambda}}_{\nu,\boldsymbol{\varepsilon}}^{\dagger}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}_{\rm int})$ ('$\diamond$') do not coincide with the optimal $\boldsymbol{\Lambda}_{\mathcal{R}}$ ('$+$'). The corresponding segmentations, $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\widehat{\boldsymbol{\Lambda}}_{\nu,\boldsymbol{\varepsilon}}^{\dagger}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}_{\rm var}))$ (Fig. 3f) and $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\widehat{\boldsymbol{\Lambda}}_{\nu,\boldsymbol{\varepsilon}}^{\dagger}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}_{\rm int}))$

(Fig. 3j), differ significantly from the targeted $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda}_{\mathcal{R}})$ (Fig. 3b).
On the opposite, $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ (Fig. 3m) perfectly matches $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ (Fig. 3a). Thanks to the exact computation of the constant term $\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*)$ in Proposition 5, the order of magnitude $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ is well reproduced by $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$, as observed on the colorbars in Figure 3. Further, $\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}_{\mathrm{int}})$ ('$\triangle$') coincides with $\boldsymbol{\Lambda}_{\mathcal{R}}$ ('$+$'), leading to segmentation $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}))$ (Fig. 3n) similar to $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\boldsymbol{\Lambda}_{\mathcal{R}})$ (Fig. 3b).
Similar observations can be made for Texture "**E**" at columns 3, 4 of Figure 3.

Altogether, these two examples illustrate that the *full* covariance is necessary so that $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ provides an accurate estimate of $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$. Moreover, $\boldsymbol{\Lambda}_{\mathcal{R}}$ appears to be well approximated by the optimal hyperparameters $\boldsymbol{\Lambda}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}})$, obtained using *full* covariance.

5.3.3 Impact of estimating the covariance matrix

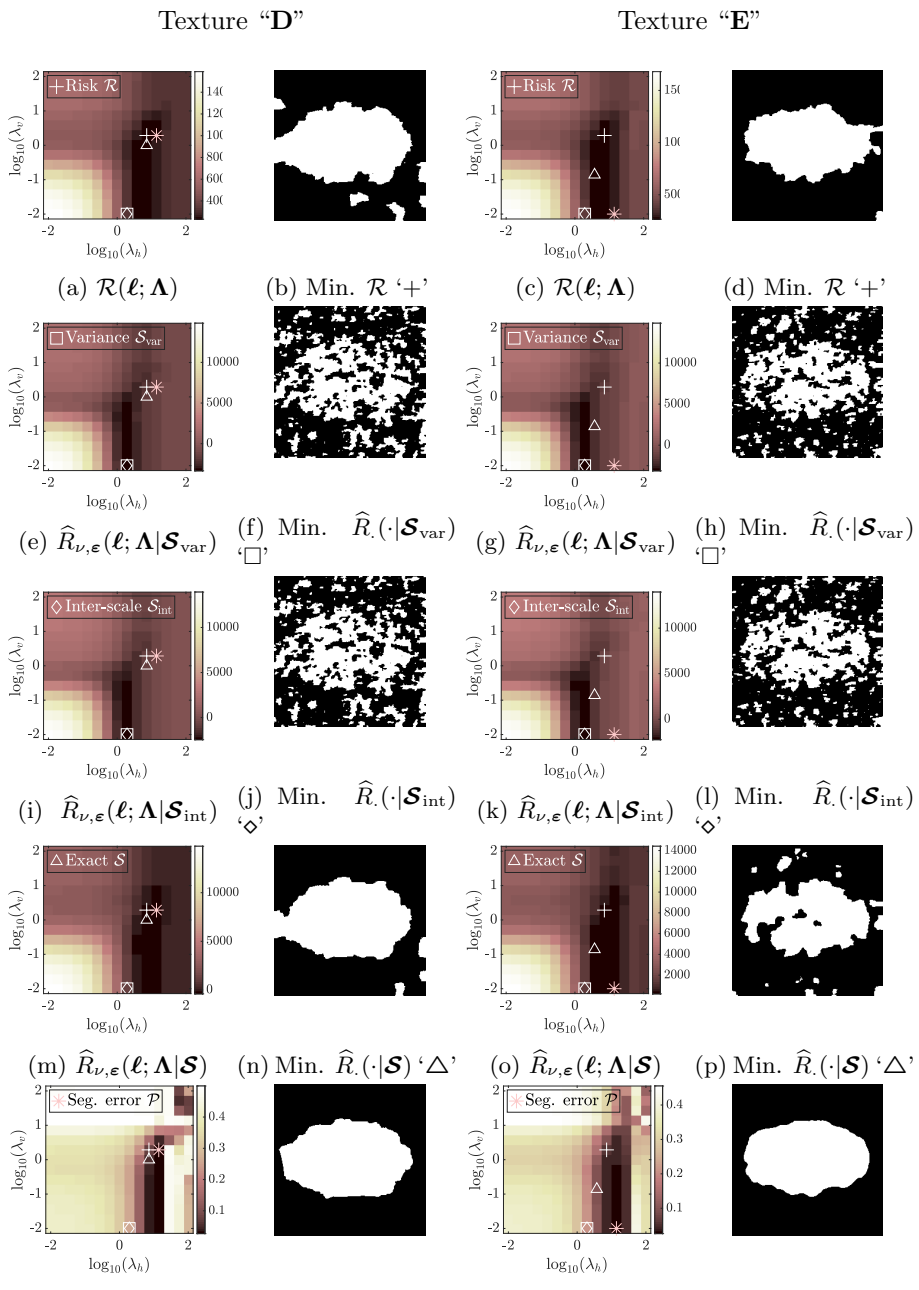
In practice, on has access to only the *estimated* covariance matrix $\widehat{\boldsymbol{\mathcal{S}}}$. This Section compares generalized SURE computed from *estimated* covariance $\widehat{\boldsymbol{\mathcal{S}}}$ to SURE computed assuming the knowledge of *true* covariance $\boldsymbol{\mathcal{S}}$.

For Texture "**D**", $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\widehat{\boldsymbol{\mathcal{S}}})$ (Figure 4b) is identical to $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ (Figure 4a). Further, optimal hyperparameters $\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\widehat{\boldsymbol{\mathcal{S}}})$ ('$\triangle$') perfectly matches $\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}})$ ('$\triangle$') and lead to similar segmentations, $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\widehat{\boldsymbol{\mathcal{S}}}))$ (Figure 4f) and $T\widehat{\boldsymbol{h}}(\boldsymbol{\ell};\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}))$ (Figure 4e). These observations are precisely quantified in Table 2 in term of values of $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ and percentage of misclassified pixels. The same observations can be made for Texture "**E**".

Altogether, Figure 4 and the quantitative results reported in Table 2 show that $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\widehat{\boldsymbol{\mathcal{S}}})$ provides an accurate estimate of $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$, and that $\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\widehat{\boldsymbol{\mathcal{S}}})$ is a good estimate of $\boldsymbol{\Lambda}_{\mathcal{R}}$.

	Texture "**D**"		Texture "**E**"	
Hyperparameter $\boldsymbol{\Lambda}$	$\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$	$\mathcal{P}(\boldsymbol{\ell};\boldsymbol{\Lambda})$	$\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$	$\mathcal{P}(\boldsymbol{\ell};\boldsymbol{\Lambda})$
$\boldsymbol{\Lambda}_{\mathcal{R}}$ '$+$'	$2.32\,10^3$	7.79%	$2.66\,10^3$	5.34%
$\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}})$ '$\triangle$'	$2.35\,10^3$	5.51%	$2.83\,10^3$	9.58%
$\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\widehat{\boldsymbol{\mathcal{S}}})$ '$\triangle$'	$2.35\,10^3$	5.51%	$2.96\,10^3$	4.61%
$\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}})$ '$\triangledown$'	$2.36\,10^3$	4.66%	$2.83\,10^3$	3.71%
$\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\widehat{\boldsymbol{\mathcal{S}}})$ '$\triangledown$'	$2.36\,10^3$	6.22%	$2.83\,10^3$	3.27%

Table 2: Grid search v.s. BFGS Algorithm 3 performance in term of quadratic error $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ and segmentation error $\mathcal{P}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ for the two different Textures "**D**" and "**E**".



(a) $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ (b) Min. $\mathcal{R}$ ‘+’ (c) $\mathcal{R}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ (d) Min. $\mathcal{R}$ ‘+’

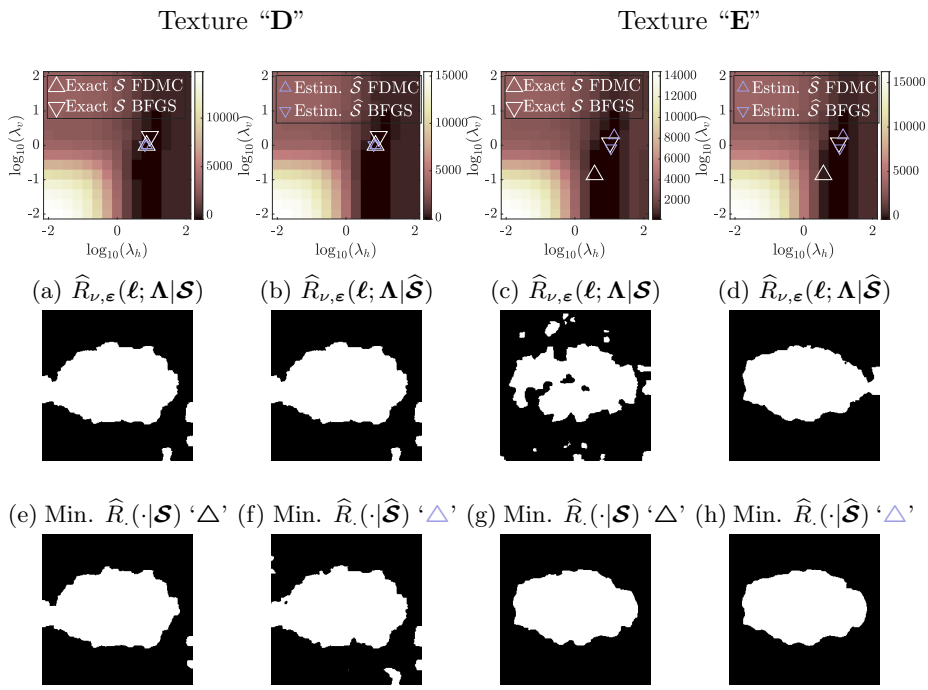
(e) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}_{\text{var}})$ (f) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}}_{\text{var}})$ ‘□’ (g) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}_{\text{var}})$ (h) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}}_{\text{var}})$ ‘□’

(i) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}_{\text{int}})$ (j) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}}_{\text{int}})$ ‘◇’ (k) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}_{\text{int}})$ (l) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}}_{\text{int}})$ ‘◇’

(m) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ (n) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}})$ ‘△’ (o) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ (p) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}})$ ‘△’

(q) $\mathcal{P}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ (r) Min. $\mathcal{P}$ ‘✳’ (s) $\mathcal{P}(\boldsymbol{\ell};\boldsymbol{\Lambda})$ (t) Min. $\mathcal{P}$ ‘✳’

Figure 3: Error maps for TV-based texture segmentation on a grid of $\boldsymbol{\Lambda} = (\lambda_h, \lambda_v)$, and segmentation obtained with associated optimal hyperparameters for piecewise Textures “**D**” (column 1, 2) and “**E**” (column 3, 4). Estimated risks $\widehat{R}_{\nu\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ computed either with variance matrix $\boldsymbol{\mathcal{S}}_{\text{var}}$ (second row), inter-scale covariance matrix $\boldsymbol{\mathcal{S}}_{\text{int}}$ (third row), or full covariance matrix $\boldsymbol{\mathcal{S}}$ (fourth row) are compared.



(a) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ (b) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\widehat{\boldsymbol{\mathcal{S}}})$ (c) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ (d) $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\widehat{\boldsymbol{\mathcal{S}}})$

(e) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}})$ '△' (f) Min. $\widehat{R}_{\cdot}(\cdot|\widehat{\boldsymbol{\mathcal{S}}})$ '△' (g) Min. $\widehat{R}_{\cdot}(\cdot|\boldsymbol{\mathcal{S}})$ '△' (h) Min. $\widehat{R}_{\cdot}(\cdot|\widehat{\boldsymbol{\mathcal{S}}})$ '△'

(i) Auto. selec. '▽' (j) Auto. selec. '▽' (k) Auto. selec. '▽' (l) Auto. selec. '▽'

Figure 4: Generalized SURE computed either from *true* covariance matrix $\boldsymbol{\mathcal{S}}$ (95), or from *estimated* covariance matrix (92) for Textures "**D**" and "**E**" (first row). Segmentations obtained minimizing the above generalized SURE (second row). Segmentations obtained with automated selection of hyperparameters from Algorithm 3, using generalized SUGAR with either *true* covariance matrix or *estimated* covariance matrix (third row).

5.4 Automated selection of hyperparameters

Section 5.3 has shown the relevance of Algorithm 3 by comparing its performance against those obtained from a grid search on hyperparameters Λ. Section 5.4 will now test the practical effectiveness of the proposed procedure by assessing the convergence of the quasi-Newton algorithm and corresponding performance in hyperparameter selection and segmentation, avoiding the recourse to any ground truth and hence to the greedy and unfeasible grid search.

5.4.1 Effective convergence of quasi-Newton Algorithm

The convergence of quasi-Newton Algorithm 3 is assessed empirically comparing automatically selected hyperparameters $\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}$ with optimal hyperparameters found from exhaustive grid search $\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}$.

Figures 4a and 4b illustrate that $\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}})$ ('▽') and $\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\widehat{\boldsymbol{\mathcal{S}}})$ ('▽') respectively match $\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}})$ ('△') and $\widehat{\boldsymbol{\Lambda}}^{\dagger}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\widehat{\boldsymbol{\mathcal{S}}})$ ('△') in the case of Tex-

ture "**D**". Similar conclusions can be drawn from Figures 4c and 4d for Texture "**E**".

Figure 4 and the quantitative results provided in Table 2 show the convergence of Algorithm 3 using $\boldsymbol{\mathcal{S}}$ (resp. $\widehat{\boldsymbol{\mathcal{S}}}$) toward the minimum of $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ (resp. $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell};\boldsymbol{\Lambda}|\widehat{\boldsymbol{\mathcal{S}}})$).

In term of computational cost, Algorithm 3 requires an average of 40 calls of Algorithm 1, compared to 225 calls needed to perform grid search at Section 5.3.2.

5.4.2 Automated selection of $\boldsymbol{\Lambda}$ and segmentation performance

Ten realizations of Textures "**D**" and "**E**" are generated following the procedure described in Section 5.1.1. For each of them, Algorithm 3 is run twice, first using $\boldsymbol{\mathcal{S}}$ and second using $\widehat{\boldsymbol{\mathcal{S}}}$.

Since here no grid search is performed, the minimum value of quadratic risk is unknown. The performance will hence be measured in terms of *normalized one-sample quadratic risk* $\widetilde{\mathcal{R}}$ defined as

$$\widetilde{\mathcal{R}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}) = \frac{\mathcal{R}(\boldsymbol{\ell};\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}}))}{\|\widehat{\boldsymbol{h}}_{\mathrm{LR}}(\boldsymbol{\ell})-\bar{\boldsymbol{h}}\|_2^2} = \frac{\|\widehat{\boldsymbol{h}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}|\boldsymbol{\mathcal{S}})-\bar{\boldsymbol{h}}\|_2^2}{\|\widehat{\boldsymbol{h}}_{\mathrm{LR}}(\boldsymbol{\ell})-\bar{\boldsymbol{h}}\|_2^2}, \tag{98}$$

measuring the improvement of the estimation achieved using TV-based texture segmentation (60) with hyperparameters automatically selected by Algorithm 3, compared to the classical least square estimate $\widehat{\boldsymbol{h}}_{\mathrm{LR}}$.

Averaged performance over ten realizations, presented in Table 3, show that the quadratic risk $\mathcal{R}$ obtained is decrease by a factor of 16 for Texture "**D**" and of 14 for Texture "**E**". The corresponding *segmentation error* is as low as 6% for Texture "**D**", and 3% for Texture "**E**". Further, the use of *estimated* covariance matrix does not degrade achieved performance compare to using *true* covariance matrix.

Hence, Algorithm 3, using the *estimated* covariance $\widehat{\boldsymbol{\mathcal{S}}}$, computed from (92), provides an efficient, parameter-free, automated and data-driven texture segmentation procedure.

	Texture "**D**"		Texture "**E**"	
Covariance matrix	$\boldsymbol{\mathcal{S}}$	$\widehat{\boldsymbol{\mathcal{S}}}$	$\boldsymbol{\mathcal{S}}$	$\widehat{\boldsymbol{\mathcal{S}}}$
$\widetilde{\mathcal{R}}(\boldsymbol{\ell}\|\cdot)$	0.060 ± 0.003	0.057 ± 0.002	0.071 ± 0.003	0.073 ± 0.004
$\mathcal{P}(\boldsymbol{\ell};\widehat{\boldsymbol{\Lambda}}^{\mathrm{BFGS}}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{\ell}\|\cdot))$ (%)	5.4 ± 0.7	6.8 ± 1.5	3.3 ± 0.7	2.8 ± 0.3

Table 3: Averaged performance of TV-based texture segmentation with automated selection of hyperparameters.

6 Conclusion

This work was focused on devising a procedure for the automated selection of the hyperparameters of parametric estimators, such as e.g., parametric linear filtering or penalized least squares. The main result obtained here consists of a theoretically grounded and practical operational fully-automated data driven procedure, that requires neither ground truth nor expert-based knowledge and work satisfactorily even when applied to a single observation of data.

To that end, Stein Unbiased Risk Estimator (SURE) was rewritten to account for additive correlated Gaussian noise, with any covariance structure. The main contribution compared to state-of-the-art procedure relies on including the covariance matrix of the noise only in SURE, rather than in the data fidelity term. The benefit is twofold: handling with a strongly convex function when Penalized Least Square is considered, and avoiding costly, if not intractable, inversion of the covariance matrix. Differentiating this Generalized SURE with respect to hyperparameters, an estimator for the risk gradient was designed, permitting to propose a Generalized Finite Difference Monte Carlo Stein Unbiased GrAdient Risk (SUGAR) estimate. The asymptotic unbiasedness of Generalized SUGAR was assessed theoretically, based on regularity assumptions on the parametric estimator.
Further, the case of sequential parametric estimators is discussed in depth in the case of primal-dual minimization scheme for Penalized Least Squares and a differentiated scheme is derived.

Embedding Generalized SURE and SUGAR into a quasi-Newton algorithm enabled to perform an automated risk minimization. An explicit algorithm permitting to implement the minimization was proposed.

To assess the performance of this automated hyperparameter selection procedure devised in a general setting, it has been customized to the specific problem of texture segmentation, based on multiscale descriptors (wavelet leaders) and nonsmooth Total-Variation based penalization. This problem is uneasy because observations are in nature multiscale, with inhomogeneous variance across scales and correlations both across scales and in space at each scale. Further, variances and correlations are unknown and need to be estimated directly from data.

Numerical simulations, conducted on ten realizations of synthetic piecewise fractal textures, permitted to show that the proposed strategy yield satisfactory performance in selecting automatically the penalization hyperparameter, leading to excellent texture segmentation, with no ad-hoc (or expert-based) tuning and without prior knowledge for ground truth, and using one-sample estimate of the covariance matrix.

The corresponding MATLAB routines, developed by ourselves and implementing these tools, ready for applications to real-world texture segmentation, where hyperparameter tuning constitutes an on-going hot topic, will be made publicly available to the research community in a documented toolbox at the time of publication.

A Proof of Theorem 1

Proof. For ease of computation we first define the *predictor* in Definition 4 and the ground truth *prediction* in Definition 5.

Definition 4 (*Predictor*)**.** From the estimator of underlying features $\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ one can equivalently consider a *prediction* estimator

$$\widehat{\boldsymbol{y}}(\boldsymbol{y};\boldsymbol{\Lambda}) \triangleq \boldsymbol{\Phi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}). \tag{99}$$

Indeed, from Assumption 2, $\boldsymbol{\Phi}^*\boldsymbol{\Phi}$ is invertible, and the relation (99) can be inverted computing

$$\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) = \left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^*\widehat{\boldsymbol{y}}(\boldsymbol{y};\boldsymbol{\Lambda}). \tag{100}$$

Definition 5 (*Prediction* ground truth)**.** The noise-free observation writes

$$\bar{\boldsymbol{y}} \triangleq \mathbb{E}_{\boldsymbol{\zeta}}\boldsymbol{y} = \boldsymbol{\Phi}\bar{\boldsymbol{x}}. \tag{101}$$

Thus, the quadratic risk defined in (10) can be expressed using operator $\mathbf{A}$ defined in (9) as

$$\begin{aligned} R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) = \mathbb{E}_{\boldsymbol{\zeta}}\left\|\boldsymbol{\Pi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \boldsymbol{\Pi}\boldsymbol{x}\right\|_2^2 &= \mathbb{E}_{\boldsymbol{\zeta}}\|\boldsymbol{\Pi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^*\left(\widehat{\boldsymbol{y}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \bar{\boldsymbol{y}}\right)\|_2^2, \\ &\overset{(9)}{=} \mathbb{E}_{\boldsymbol{\zeta}}\|\mathbf{A}\left(\widehat{\boldsymbol{y}}(\boldsymbol{y};\boldsymbol{\Lambda}) - \bar{\boldsymbol{y}}\right)\|_2^2 \end{aligned} \tag{102}$$

which will be easier to manipulate in the following when expressed in term of noise-free (or noisy) observations $\bar{\boldsymbol{y}}$ (or $\boldsymbol{y}$) and *prediction* $\widehat{\boldsymbol{y}}$.

By construction, the matrix $\mathbf{A}$, defined in (9), performs both:

- The projection on the interest subspace $\mathcal{I}$ of $\mathcal{H}$ via the linear operator $\boldsymbol{\Pi}$.
- The transition from predicted quantities $\widehat{\boldsymbol{y}}$ to estimated features $\widehat{\boldsymbol{x}}$, making use of relation (100).

From now, for sake of simplicity, we make implicit the dependency of $\widehat{\boldsymbol{x}}$ in $(\boldsymbol{y};\boldsymbol{\Lambda})$. From the model (1) and the Assumption 1 on the noise probability distribution, one directly derive two useful relations:

$$\mathbb{E}_{\boldsymbol{\zeta}}\left\|\mathbf{A}\left(\boldsymbol{y}-\bar{\boldsymbol{y}}\right)\right\|_2^2 \overset{(101)}{=} \mathbb{E}_{\boldsymbol{\zeta}}\left\|\mathbf{A}\left(\boldsymbol{y}-\boldsymbol{\Phi}\bar{\boldsymbol{x}}\right)\right\|_2^2 = \mathbb{E}_{\boldsymbol{\zeta}}\left\|\mathbf{A}\boldsymbol{\zeta}\right\|_2^2 \quad \overset{\text{Hyp. 1}}{=} \operatorname{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*), \tag{103}$$

$$\mathbb{E}_{\boldsymbol{\zeta}}\langle\mathbf{A}\boldsymbol{y}, \mathbf{A}\left(\boldsymbol{y}-\bar{\boldsymbol{y}}\right)\rangle \overset{\mathbb{E}(\boldsymbol{y}-\bar{\boldsymbol{y}})=0}{=} \mathbb{E}_{\boldsymbol{\zeta}}\langle\mathbf{A}\left(\boldsymbol{y}-\bar{\boldsymbol{y}}\right), \mathbf{A}\left(\boldsymbol{y}-\bar{\boldsymbol{y}}\right)\rangle \quad \overset{\text{Hyp. 1}}{=} \operatorname{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*). \tag{104}$$

Thus the risk can be expanded as

$$
\begin{aligned}
& R[\widehat{\boldsymbol{x}}](\boldsymbol{\Lambda}) \\
& \triangleq \ \mathbb{E}_{\boldsymbol{\zeta}} \left\| \mathbf{A} \left(\widehat{\boldsymbol{y}} - \bar{\boldsymbol{y}} \right) \right\|_2^2 \\
& = \ \mathbb{E}_{\boldsymbol{\zeta}} \left[\left\| \mathbf{A} \left(\widehat{\boldsymbol{y}} - \boldsymbol{y} \right) \right\|_2^2 + \left\| \mathbf{A} \left(\boldsymbol{y} - \bar{\boldsymbol{y}} \right) \right\|_2^2 + 2 \langle \mathbf{A} \left(\widehat{\boldsymbol{y}} - \boldsymbol{y} \right), \mathbf{A} \left(\boldsymbol{y} - \bar{\boldsymbol{y}} \right) \rangle \right] \\
& \overset{(103)}{=} \ \mathbb{E}_{\boldsymbol{\zeta}} \left[\left\| \mathbf{A} \left(\widehat{\boldsymbol{y}} - \boldsymbol{y} \right) \right\|_2^2 + 2 \langle \mathbf{A} \widehat{\boldsymbol{y}}, \mathbf{A} \left(\boldsymbol{y} - \bar{\boldsymbol{y}} \right) \rangle - 2 \langle \mathbf{A} \boldsymbol{y}, \mathbf{A} \left(\boldsymbol{y} - \bar{\boldsymbol{y}} \right) \rangle \right] + \mathrm{Tr}(\mathbf{A} \boldsymbol{\mathcal{S}} \mathbf{A}^*) \\
& \overset{(104)}{=} \ \mathbb{E}_{\boldsymbol{\zeta}} \left[\left\| \mathbf{A} \left(\widehat{\boldsymbol{y}} - \boldsymbol{y} \right) \right\|_2^2 + 2 \langle \mathbf{A} \widehat{\boldsymbol{y}}, \mathbf{A} \left(\boldsymbol{y} - \bar{\boldsymbol{y}} \right) \rangle \right] - 2 \mathrm{Tr}(\mathbf{A} \boldsymbol{\mathcal{S}} \mathbf{A}^*) + \mathrm{Tr}(\mathbf{A} \boldsymbol{\mathcal{S}} \mathbf{A}^*) \\
& = \ \mathbb{E}_{\boldsymbol{\zeta}} \left[\left\| \mathbf{A} \left(\widehat{\boldsymbol{y}} - \boldsymbol{y} \right) \right\|_2^2 + 2 \langle \mathbf{A} \widehat{\boldsymbol{y}}, \mathbf{A} \left(\boldsymbol{y} - \bar{\boldsymbol{y}} \right) \rangle \right] - \mathrm{Tr}(\mathbf{A} \boldsymbol{\mathcal{S}} \mathbf{A}^*) \\
& = \ \mathbb{E}_{\boldsymbol{\zeta}} \left[\left\| \mathbf{A} \left(\boldsymbol{\Phi} \widehat{\boldsymbol{x}} - \boldsymbol{y} \right) \right\|_2^2 + 2 \langle \mathbf{A}^* \mathbf{A} \boldsymbol{\Phi} \widehat{\boldsymbol{x}}, \left(\boldsymbol{y} - \boldsymbol{\Phi} \bar{\boldsymbol{x}} \right) \rangle \right] - \mathrm{Tr}(\mathbf{A} \boldsymbol{\mathcal{S}} \mathbf{A}^*) \\
& = \ \mathbb{E}_{\boldsymbol{\zeta}} \left\| \mathbf{A} \left(\boldsymbol{\Phi} \widehat{\boldsymbol{x}} - \boldsymbol{y} \right) \right\|_2^2 + 2 \mathbb{E}_{\boldsymbol{\zeta}} \langle \mathbf{A}^* \mathbf{A} \boldsymbol{\Phi} \widehat{\boldsymbol{x}}, \boldsymbol{\zeta} \rangle - \mathrm{Tr}(\mathbf{A} \boldsymbol{\mathcal{S}} \mathbf{A}^*) \\
& = \ \mathbb{E}_{\boldsymbol{\zeta}} \left\| \mathbf{A} \left(\boldsymbol{\Phi} \widehat{\boldsymbol{x}} - \boldsymbol{y} \right) \right\|_2^2 + 2 \mathbb{E}_{\boldsymbol{\zeta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}, \boldsymbol{\zeta} \rangle - \mathrm{Tr}(\mathbf{A} \boldsymbol{\mathcal{S}} \mathbf{A}^*),
\end{aligned}
$$

re-injecting the definition of $\mathbf{A}$ (9) in terms of $\boldsymbol{\Phi}$ and $\boldsymbol{\Pi}$.

The second term, $\mathbb{E}_{\boldsymbol{\zeta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\zeta} \rangle$, is called the *degrees of freedom* [29]. From Assumption 3 it is well-defined and writes

$$
\mathbb{E}_{\boldsymbol{\zeta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\zeta} \rangle = \tag{105}
$$

$$
\frac{1}{\sqrt{(2\pi)^P |\det(\boldsymbol{\mathcal{S}})|}} \int \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\zeta} \rangle \, \exp\left(-\frac{\boldsymbol{\zeta}^* \boldsymbol{\mathcal{S}}^{-1} \boldsymbol{\zeta}}{2} \right) \mathrm{d}\boldsymbol{\zeta}, \tag{106}
$$

hence requiring generalized Stein's lemma to be estimated[4].

Because of the off-diagonal terms in $\boldsymbol{\mathcal{S}}^{-1}$, the Integration by Parts (IP) required to transform (105) cannot be directly justified, thus Stein's lemma generalization to $\mathcal{G}$-valued random variable $\boldsymbol{\zeta}$ is not straightforward. Hence we propose to first diagonalize $\boldsymbol{\mathcal{S}}^{-1}$ (which is a symmetric matrix) in a orthonormal basis, obtaining

$$
\boldsymbol{\mathcal{S}}^{-1} = \boldsymbol{\mathcal{V}}^* \boldsymbol{\mathcal{D}} \boldsymbol{\mathcal{V}},
$$

with $\boldsymbol{\mathcal{V}}$ an orthonormal matrix (which columns are eigenvectors of $\boldsymbol{\mathcal{S}}^{-1}$) and $\boldsymbol{\mathcal{D}} = \mathrm{diag}(\beta_1, \ldots, \beta_P)$ containing (positive) eigenvalues of $\boldsymbol{\mathcal{S}}^{-1}$. Then, setting $\boldsymbol{\vartheta} = \boldsymbol{\mathcal{V}} \boldsymbol{\zeta}$

$$
\begin{aligned}
& \mathbb{E}_{\boldsymbol{\vartheta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\vartheta} \rangle = \\
& \frac{1}{\sqrt{(2\pi)^P |\det(\boldsymbol{\mathcal{S}})|}} \int \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\mathcal{V}}^{-1} \boldsymbol{\vartheta} \rangle \, \exp\left(-\frac{\boldsymbol{\vartheta}^* \boldsymbol{\mathcal{D}} \boldsymbol{\vartheta}}{2} \right) |\det(\boldsymbol{\mathcal{V}}^{-1})| \mathrm{d}\boldsymbol{\vartheta}.
\end{aligned}
$$

with $\boldsymbol{\vartheta}^* \boldsymbol{\mathcal{D}} \boldsymbol{\vartheta} = \sum_{p=1}^P \beta_p |\vartheta_p|^2$.

[4] Stein's lemma states that, for a real random variable $\zeta \sim \mathcal{N}(0, \sigma^2)$, if $f : \mathbb{R} \to \mathbb{R}$ is a function such that both $\mathbb{E}_\zeta[\zeta f(\zeta)]$ and $\mathbb{E}_\zeta[f'(\zeta)]$ exist, then $\mathbb{E}_\zeta[\zeta f(\zeta)] = \sigma^2 \mathbb{E}_\zeta[f'(\zeta)]$. Its demonstration relies on appropriate integration by parts.

Since $\boldsymbol{\mathcal{V}}$ is orthonormal: $\boldsymbol{\mathcal{V}}^{-1} = \boldsymbol{\mathcal{V}}^*$ and $|\det(\boldsymbol{\mathcal{V}}^{-1})| = 1$, leading to

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\vartheta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\vartheta} \rangle \\
&= \frac{1}{\sqrt{(2\pi)^P |\det(\boldsymbol{\mathcal{S}})|}} \int \langle \boldsymbol{\mathcal{V}} \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\vartheta} \rangle \exp\left(-\frac{\boldsymbol{\vartheta}^* \boldsymbol{\mathcal{D}} \boldsymbol{\vartheta}}{2} \right) \mathrm{d}\boldsymbol{\vartheta} \\
&= \frac{1}{\sqrt{(2\pi)^P |\det(\boldsymbol{\mathcal{S}})|}} \int \sum_{p=1}^{P} \left(\boldsymbol{\mathcal{V}} \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)_p \vartheta_p \exp\left(-\frac{\sum_{p=1}^{P} \beta_p |\vartheta_p|^2}{2} \right) \mathrm{d}\vartheta_1 \dots \mathrm{d}\vartheta_P \\
&\overset{\text{(IP)}}{=} \mathbb{E}_{\boldsymbol{\vartheta}} \left[\sum_{p=1}^{P} \frac{1}{\beta_p} \frac{\partial \left(\boldsymbol{\mathcal{V}} \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)_p}{\partial \vartheta_p} \right] \\
&= \mathbb{E}_{\boldsymbol{\vartheta}} \left[\mathrm{Tr}\left(\boldsymbol{\mathcal{D}}^{-1} \frac{\partial \left(\boldsymbol{\mathcal{V}} \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{\vartheta}} \right) \right],
\end{aligned}
$$

where $\dfrac{\partial \left(\boldsymbol{\mathcal{V}} \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{\vartheta}}$ denotes the Jacobian matrix of $\boldsymbol{\mathcal{V}} \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ with respect to the variable $\boldsymbol{\vartheta} \triangleq \boldsymbol{\mathcal{V}}^{-1} \boldsymbol{\zeta}$. In order to go back to variable $\boldsymbol{\zeta}$, we make use of (1) relating $\boldsymbol{y}$ and $\boldsymbol{\zeta}$, and apply the reverse change of variable $\boldsymbol{\zeta} \triangleq \boldsymbol{\mathcal{V}} \boldsymbol{\vartheta}$ and obtain

$$
\frac{\partial \left(\boldsymbol{\mathcal{V}} \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{\vartheta}} = \boldsymbol{\mathcal{V}} \frac{\partial \left(\mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{\zeta}} \boldsymbol{\mathcal{V}}^{-1} = \boldsymbol{\mathcal{V}} \frac{\partial \left(\mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{y}} \boldsymbol{\mathcal{V}}^{-1},
$$

because $\partial_{\boldsymbol{\zeta}} \boldsymbol{y} = \boldsymbol{I}_P$ (the identity matrix of size $P \times P$).

Using the cyclicality of trace and the fact that $\boldsymbol{\mathcal{V}}$ is orthonormal, we finally obtain a closed-form expression of the degrees of freedom:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\vartheta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}), \boldsymbol{\vartheta} \rangle &= \mathbb{E}_{\boldsymbol{\zeta}} \left[\mathrm{Tr}\left(\boldsymbol{\mathcal{D}}^{-1} \boldsymbol{\mathcal{V}} \frac{\partial \left(\mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{y}} \boldsymbol{\mathcal{V}}^{-1} \right) \right], \\
&= \mathbb{E}_{\boldsymbol{\zeta}} \left[\mathrm{Tr}\left(\boldsymbol{\mathcal{V}}^{-1} \boldsymbol{\mathcal{D}}^{-1} \boldsymbol{\mathcal{V}} \frac{\partial \left(\mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{y}} \right) \right], \\
&= \mathbb{E}_{\boldsymbol{\zeta}} \left[\mathrm{Tr}\left(\boldsymbol{\mathcal{S}} \frac{\partial \left(\mathbf{A}^* \boldsymbol{\Pi} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right)}{\partial \boldsymbol{y}} \right) \right], \\
&= \mathbb{E}_{\boldsymbol{\zeta}} \left[\mathrm{Tr}\left(\boldsymbol{\mathcal{S}} \mathbf{A}^* \boldsymbol{\Pi} \partial_{\boldsymbol{y}} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right) \right]
\end{aligned}
$$

□

B Finite Difference Monte Carlo SURE

Proof. First, remark that since $\boldsymbol{y} \mapsto \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})$ is Lipschitz continuous from Assumption 4, it is Lebesgue differentiable almost everywhere and its Lebesgue derivative equals its weak derivative almost everywhere. Then, based on Theorem 1, the only difficulty relies in dominating the degrees of freedom, since it is the only term depending on the Finite Difference step ν.
Applying successively both Monte Carlo and Finite Difference strategies presented in Section 2.4 we obtain

$$
\begin{aligned}
\mathrm{Tr}\left(\boldsymbol{\mathcal{S}} \mathbf{A}^* \boldsymbol{\Pi} \partial_{\boldsymbol{y}} \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}) \right) &\overset{\text{Monte Carlo}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \boldsymbol{\mathcal{S}} \mathbf{A}^* \boldsymbol{\Pi} \frac{\partial \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\partial \boldsymbol{y}} [\boldsymbol{\varepsilon}], \boldsymbol{\varepsilon} \right\rangle \qquad (107) \\
&\overset{\text{Finite Difference}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \lim_{\nu \to 0} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu \boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}} \boldsymbol{\varepsilon} \right\rangle.
\end{aligned}
$$

Making use of the centered normalized Gaussian probability density function of $\boldsymbol{\varepsilon}$, the above expectation writes

$$\mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \lim_{\nu \to 0} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle \tag{108}$$

$$(\text{p.d.f. of } \boldsymbol{\varepsilon}) = \int_{\mathbb{R}^P} \lim_{\nu \to 0} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle \frac{\mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \mathrm{d}\boldsymbol{\varepsilon}}{(2\pi)^{P/2}}$$

Then the following majorations hold

$$\left| \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{(\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}))}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle \right| \mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \tag{109}$$

$$(\text{Cauchy-Schwarz}) \leq \left\| \mathbf{A}^* \boldsymbol{\Pi} \frac{(\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda}))}{\nu} \right\| \|\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\| \, \mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}}$$

$$(\text{Bounded operators}) \leq \|\mathbf{A}^*\| \|\boldsymbol{\Pi}\| \left\| \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu} \right\| \|\boldsymbol{\mathcal{S}}\| \|\boldsymbol{\varepsilon}\| \mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}}$$

$$(\text{Hyp. 4: } L_1\text{-Lipschitz}) \leq \|\mathbf{A}^*\| \|\boldsymbol{\Pi}\| L_1 \|\boldsymbol{\varepsilon}\| \|\boldsymbol{\mathcal{S}}\| \|\boldsymbol{\varepsilon}\| \mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}},$$

with $\|\boldsymbol{\varepsilon}\|^2 \mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}}$ integrable over $\mathbb{R}^P$. Further, the domination being independent of ν the limit can be interchanged with the integral on variable $\boldsymbol{\varepsilon}$ which gives

$$\int_{\mathbb{R}^P} \lim_{\nu \to 0} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle \frac{\mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \mathrm{d}\boldsymbol{\varepsilon}}{(2\pi)^{P/2}} \tag{110}$$

$$= \lim_{\nu \to 0} \int_{\mathbb{R}^P} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle \frac{\mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \mathrm{d}\boldsymbol{\varepsilon}}{(2\pi)^{P/2}},$$

and

$$\left| \lim_{\nu \to 0} \int_{\mathbb{R}^P} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle \frac{\mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \mathrm{d}\boldsymbol{\varepsilon}}{(2\pi)^{P/2}} \right|$$

$$\leq \|\mathbf{A}^*\| \|\boldsymbol{\Pi}\| L_1 \|\boldsymbol{\mathcal{S}}\| \int_{\mathbb{R}^P} \|\boldsymbol{\varepsilon}\|^2 \frac{\mathrm{e}^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \mathrm{d}\boldsymbol{\varepsilon}}{(2\pi)^{P/2}} < \infty. \tag{111}$$

Then, Equation (110) means that

$$\mathrm{Tr}\left(\boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})\right) = \lim_{\nu \to 0} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle. \tag{112}$$

Further, the majoration obtained in Equation (111) not depending on $\boldsymbol{\zeta}$ (since L_1 does not depend on $\boldsymbol{y}$, as stated in Assumption 4), neither on ν, the limits on ν and the expected value with respect to Gaussian random noise $\boldsymbol{\zeta}$ can be interchanged so that

$$\mathbb{E}_{\boldsymbol{\zeta}} \mathrm{Tr}\left(\boldsymbol{\mathcal{S}}\mathbf{A}^*\boldsymbol{\Pi}\partial_{\boldsymbol{y}}\widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})\right) = \mathbb{E}_{\boldsymbol{\zeta}} \lim_{\nu \to 0} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle \tag{113}$$

$$= \lim_{\nu \to 0} \mathbb{E}_{\boldsymbol{\zeta}, \boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\boldsymbol{x}}(\boldsymbol{y} + \nu\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \widehat{\boldsymbol{x}}(\boldsymbol{y}; \boldsymbol{\Lambda})}{\nu}, \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon} \right\rangle.$$

giving the asymptotic unbiasedness of the Finite Difference Monte Carlo estimator of degrees of freedom and hence of the Finite Difference Monte Carlo SURE (17). □

C Finite Difference Monte Carlo SUGAR

Proof. We remind that Finite Difference Monte Carlo SUGAR is composed of two terms, denoted $(\partial \mathbf{1})$ and $(\partial \mathbf{2})$ in the following:

$$\partial_{\mathbf{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\epsilon}}(\boldsymbol{y}; \mathbf{\Lambda}|\boldsymbol{\mathcal{S}}) \triangleq \tag{114}$$

$$\underbrace{2\left(\mathbf{A}\mathbf{\Phi}\partial_{\mathbf{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda})\right)^{*}\mathbf{A}\left(\mathbf{\Phi}\widehat{\boldsymbol{x}}-\boldsymbol{y}\right)}_{(\partial \mathbf{1})} + \frac{2}{\nu}\underbrace{\left\langle \mathbf{A}^{*}\mathbf{\Pi}\left(\partial_{\mathbf{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\mathbf{\Lambda})-\partial_{\mathbf{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda})\right), \boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle}_{(\partial \mathbf{2})}.$$

Since the estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda})$ is weakly differentiable with respect to $\mathbf{\Lambda}$, so is the true risk $R[\widehat{\boldsymbol{x}}](\mathbf{\Lambda})$. Thus, for any continuously differentiable test function $\varphi : \mathbb{R}^L \to \mathbb{R} \in C^1(\mathbb{V})$ with compact support denoted $\mathbb{V} \subset \mathbb{R}^L$, and any component $l \in \{1,\dots,L\}$ of the gradient of the risk $\partial_{\mathbf{\Lambda}} R[\widehat{\boldsymbol{x}}](\mathbf{\Lambda})$

$$\begin{aligned}
\int_{\mathbb{R}^L} \left(\partial_{\mathbf{\Lambda}} R[\widehat{\boldsymbol{x}}](\mathbf{\Lambda})\right)_l \varphi(\mathbf{\Lambda})\,\mathrm{d}\mathbf{\Lambda} &= \int_{\mathbb{V}} \left(\partial_{\mathbf{\Lambda}} R[\widehat{\boldsymbol{x}}](\mathbf{\Lambda})\right)_l \varphi(\mathbf{\Lambda})\,\mathrm{d}\mathbf{\Lambda} \qquad (115)\\
\text{(Weak differentiability)} &= -\int_{\mathbb{V}} R[\widehat{\boldsymbol{x}}](\mathbf{\Lambda})\left(\partial_{\mathbf{\Lambda}}\varphi(\mathbf{\Lambda})\right)_l \mathrm{d}\mathbf{\Lambda}\\
\text{(Definition of the risk (10))} &= -\int_{\mathbb{V}} \mathbb{E}_{\boldsymbol{\zeta}} \left\|\mathbf{\Pi}\widehat{\boldsymbol{x}}(\boldsymbol{y};\mathbf{\Lambda}) - \mathbf{\Pi}\boldsymbol{x}\right\|_2^2 \left(\partial_{\mathbf{\Lambda}}\varphi(\mathbf{\Lambda})\right)_l \mathrm{d}\mathbf{\Lambda}\\
\text{(Theorem 1)} &= -\int_{\mathbb{V}} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \lim_{\nu\to 0} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\left(\partial_{\mathbf{\Lambda}}\varphi(\mathbf{\Lambda})\right)_l \mathrm{d}\mathbf{\Lambda}\\
\text{(Theorem 2)} &= -\int_{\mathbb{V}} \lim_{\nu\to 0} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\left(\partial_{\mathbf{\Lambda}}\varphi(\mathbf{\Lambda})\right)_l \mathrm{d}\mathbf{\Lambda}\\
\text{(Dominated convergence)} &\overset{\text{(DC 1)}}{=} -\lim_{\nu\to 0}\int_{\mathbb{V}} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\left(\partial_{\mathbf{\Lambda}}\varphi(\mathbf{\Lambda})\right)_l \mathrm{d}\mathbf{\Lambda}\\
\text{(Fubini)} &\overset{\text{(Fu 1)}}{=} -\lim_{\nu\to 0} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \int_{\mathbb{V}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\left(\partial_{\mathbf{\Lambda}}\varphi(\mathbf{\Lambda})\right)_l \mathrm{d}\mathbf{\Lambda}\\
\text{(Proposition 1)} &= \lim_{\nu\to 0} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \int_{\mathbb{V}} \left(\partial_{\mathbf{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\right)_l \varphi(\mathbf{\Lambda})\,\mathrm{d}\mathbf{\Lambda}\\
&\overset{\text{(Fu 2)}}{=} \lim_{\nu\to 0} \int_{\mathbb{V}} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \left(\partial_{\mathbf{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\right)_l \varphi(\mathbf{\Lambda})\,\mathrm{d}\mathbf{\Lambda}\\
&\overset{\text{(DC 2)}}{=} \int_{\mathbb{V}} \lim_{\nu\to 0} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \left(\partial_{\mathbf{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\right)_l \varphi(\mathbf{\Lambda})\,\mathrm{d}\mathbf{\Lambda}.
\end{aligned}$$

(DC 1) In order to apply dominated convergence theorem interchanging the limit on ν and the integral on $\mathbb{V}$, since φ is a test function with compact domain $\mathbb{V}$, we derive a bound of $\mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})$ which is independent of both ν and $\mathbf{\Lambda}$. Using the probability density functions we have

$$\mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}}) = \int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\mathbf{\Lambda}|\boldsymbol{\mathcal{S}})\,\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon} \tag{116}$$

where $\mathcal{G}_{\boldsymbol{\mathcal{S}}}$ (resp. $\mathcal{G}_{\boldsymbol{I}}$) denotes the Gaussian probability density function with covariance matrix $\boldsymbol{\mathcal{S}}$ (resp. $\boldsymbol{I}$)

$$\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta}) \triangleq \frac{\exp\left(-\|\boldsymbol{\zeta}\|^2_{\boldsymbol{\mathcal{S}}^{-1}}/2\right)}{\sqrt{(2\pi)^P |\det \boldsymbol{\mathcal{S}}|}} \quad \left(\text{resp. } \mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\zeta}) \triangleq \frac{\exp\left(-\|\boldsymbol{\zeta}\|_2^2/2\right)}{\sqrt{(2\pi)^P}}\right). \tag{117}$$

We remind that $\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})$ is decomposed of three terms

$$\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}}) \triangleq$$
$$\underbrace{\|\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}-\boldsymbol{y}\right)\|_2^2}_{\textbf{(1)}} + \underbrace{\frac{1}{\nu}\left\langle \mathbf{A}^*\boldsymbol{\Pi}\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})-\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})\right),\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle}_{\textbf{(2)}} - \underbrace{\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*)}_{\textbf{(3)}}. \tag{118}$$

which will be bounded separately.
(1) First, combining Assumptions *(i)* and *(ii)* of Assumption 4, we have

$$\|\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})-\widehat{\boldsymbol{x}}(\mathbf{0}_P;\boldsymbol{\Lambda})\| \leq L_1\|\boldsymbol{y}-\mathbf{0}_P\| \Longrightarrow \|\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\| \leq L_1\|\boldsymbol{y}\|, \tag{119}$$

which can be used to bound first term (1) of (118) as

$$\begin{aligned}\|\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}-\boldsymbol{y}\right)\| &\leq \|\mathbf{A}\|\,\|\boldsymbol{\Phi}\widehat{\boldsymbol{x}}-\boldsymbol{y}\| \\ &\leq \|\mathbf{A}\|\left(\|\boldsymbol{\Phi}\widehat{\boldsymbol{x}}\|+\|\boldsymbol{y}\|\right) \\ &\leq \|\mathbf{A}\|\left(\|\boldsymbol{\Phi}\|\|\widehat{\boldsymbol{x}}\|+\|\boldsymbol{y}\|\right) \\ &\leq \|\mathbf{A}\|\left(\|\boldsymbol{\Phi}\|L_1\|\boldsymbol{y}\|+\|\boldsymbol{y}\|\right) \\ &\leq \|\mathbf{A}\|\left(\|\boldsymbol{\Phi}\|L_1+1\right)\|\boldsymbol{y}\|. \end{aligned} \tag{120}$$

Since by definition $\boldsymbol{\zeta}=\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}$, $\boldsymbol{y}\mapsto\|\boldsymbol{y}\|$ is integrable against the Gaussian density $\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})$, and the above domination being independent of ν, it enable us to apply dominated convergence.
(2) Making use of the domination of Equation (109) we have

$$\begin{aligned}&\left|\left\langle \mathbf{A}^*\boldsymbol{\Pi}\frac{\left(\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right)}{\nu},\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle\right| \\ &\leq \|\mathbf{A}^*\|\|\boldsymbol{\Pi}\|L_1\|\boldsymbol{\varepsilon}\|\|\boldsymbol{\mathcal{S}}\|\|\boldsymbol{\varepsilon}\|,\end{aligned} \tag{121}$$

and $\|\boldsymbol{\varepsilon}\|^2$ being integrable against $\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})$, dominated convergence applied.
(3) The third term being constant, the domination is obvious.
Putting altogether the majoration of **(1)**, **(2)** and **(3)**

$$\begin{aligned}\left|\mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}}\widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y};\boldsymbol{\Lambda}|\boldsymbol{\mathcal{S}})\right| \leq &\int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}}\|\mathbf{A}\|\left(\|\boldsymbol{\Phi}\|L_1+1\right)\|\boldsymbol{y}\|\;\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon} \\ &+\int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}}\|\mathbf{A}^*\|\|\boldsymbol{\Pi}\|L_1\|\boldsymbol{\mathcal{S}}\|\|\boldsymbol{\varepsilon}\|^2\,\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon} \\ &+\int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}}\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*)\,\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon} \leq \infty,\end{aligned} \tag{122}$$

the majoration being independent of ν and $\boldsymbol{\Lambda}$ dominated convergence applies.

(Fu 1) The above domination of Equation (122) being independent of $\boldsymbol{\Lambda}$, then Fubini's theorem applies.

(Fu 2) The first term of (114), denoted as $(\partial\mathbf{1})$ can be easily dominated by an integrable function. Indeed, Assumption 5 implies that $\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is uniformly

bounded by L_2, independently of $\boldsymbol{y}$. Then it follows from Cauchy-Schwarz inequality and the domination of Equation (120)

$$\begin{aligned} 2\left\|\left(\mathbf{A}\boldsymbol{\Phi}\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right)^{*}\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}-\boldsymbol{y}\right)\right\| &\leq 2\|\mathbf{A}\boldsymbol{\Phi}\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\|\|\mathbf{A}\left(\boldsymbol{\Phi}\widehat{\boldsymbol{x}}-\boldsymbol{y}\right)\|, \\ &\leq 2\|\mathbf{A}\|\|\boldsymbol{\Phi}\|L_2\|\mathbf{A}\|\left(\boldsymbol{\Phi}L_1+1\right)\|\boldsymbol{y}\|. \end{aligned} \tag{123}$$

Hence, since $\|\boldsymbol{y}\|$ is integrable against $\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})$ and the domination being independent of ν, both Fubini and dominated convergence theorems apply.

The second term of (114), denoted as $(\partial\mathbf{2})$, corresponding to the derivative of the estimation of degrees of freedom, can be rewritten as

$$\frac{2}{\nu}\left\langle\mathbf{A}^{*}\boldsymbol{\Pi}\left(\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right),\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle \triangleq \frac{2}{\nu}\left\langle u(\boldsymbol{\zeta}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-u(\boldsymbol{\zeta};\boldsymbol{\Lambda}),\boldsymbol{\varepsilon}\right\rangle$$

where we set $u(\boldsymbol{z};\boldsymbol{\Lambda}) \triangleq \boldsymbol{\mathcal{S}}\mathbf{A}^{*}\boldsymbol{\Pi}\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{\Phi}\boldsymbol{x}+\boldsymbol{z};\boldsymbol{\Lambda})$. Since $\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})$ is uniformly bounded by L_2, independently of $\boldsymbol{y}$, and all the linear operators are assumed to be bounded, then $\boldsymbol{\Lambda}\mapsto u(\boldsymbol{z};\boldsymbol{\Lambda})$ is bounded by some $L_u>0$, independently of $\boldsymbol{z}$. Then

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}}\left[\frac{2}{\nu}\left\langle\mathbf{A}^{*}\boldsymbol{\Pi}\left(\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-\partial_{\boldsymbol{\Lambda}}\widehat{\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\Lambda})\right),\boldsymbol{\mathcal{S}}\boldsymbol{\varepsilon}\right\rangle\right] \\ &= \int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}}\frac{2}{\nu}\left\langle u(\boldsymbol{\zeta}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda})-u(\boldsymbol{\zeta};\boldsymbol{\Lambda}),\boldsymbol{\varepsilon}\right\rangle\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\,\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon} \\ &= \int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}}\frac{2}{\nu}\left\langle u(\boldsymbol{\zeta}+\nu\boldsymbol{\varepsilon};\boldsymbol{\Lambda}),\boldsymbol{\varepsilon}\right\rangle\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\,\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon}-\int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}}\frac{2}{\nu}\left\langle u(\boldsymbol{\zeta};\boldsymbol{\Lambda}),\boldsymbol{\varepsilon}\right\rangle\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\,\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon} \\ &= \int_{\boldsymbol{\zeta}}\int_{\boldsymbol{\varepsilon}}\frac{2}{\nu}\left\langle u(\boldsymbol{\zeta};\boldsymbol{\Lambda}),\boldsymbol{\varepsilon}\right\rangle\left(\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta}-\nu\boldsymbol{\varepsilon})-\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\right)\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\,\mathrm{d}\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\varepsilon}. \end{aligned} \tag{124}$$

Further, the following majoration holds

$$\left\|\frac{2}{\nu}\left\langle u(\boldsymbol{\zeta};\boldsymbol{\Lambda}),\boldsymbol{\varepsilon}\right\rangle\left(\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta}-\nu\boldsymbol{\varepsilon})-\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\right)\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})\right\| \leq \frac{2}{\nu}L_u\|\boldsymbol{\varepsilon}\|\left|\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta}-\nu\boldsymbol{\varepsilon})-\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta})\right|\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon}). \tag{125}$$

Up to a unitary variable change (see Appendix A, with $\boldsymbol{\vartheta}=\boldsymbol{\mathcal{V}}^{-1}\boldsymbol{\varepsilon}$), we can assume that the covariance matrix is diagonal, with diagonal terms $\left(s_i^2\right)_{i=1}^{P}$ so that

$$\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta}) = \prod_{i=1}^{P}\frac{\exp\left(-|\zeta_i|^2/2s_i^2\right)}{\sqrt{2\pi s_i^2}} \tag{126}$$

and we define the one-dimensional Gaussian densities as

$$g_{s_i^2}(\zeta_i) \triangleq \frac{\exp\left(-|\zeta_i|^2/2s_i^2\right)}{\sqrt{2\pi s_i^2}}. \tag{127}$$

From Taylor inequality,

$$\left|g_{s_i^2}(\zeta_i-\nu\varepsilon_i)-g_{s_i^2}(\zeta_i)\right| \leq \int_{(0,\nu\varepsilon_i)}\left|g'_{s_i^2}(\zeta_i-\tau)\right|\mathrm{d}\tau, \tag{128}$$

where $(0, \nu\varepsilon_i)$ denotes the ordered interval, taking into account that ε_i might be negative that is

$$(0, \nu\varepsilon_i) = \begin{cases} [0, \nu\varepsilon_i] & \text{if } \varepsilon_i \geq 0 \\ [\nu\varepsilon_i, 0] & \text{else} \end{cases} \tag{129}$$

then

$$\begin{aligned} \int_{\zeta_i} |g_{s_i^2}(\zeta_i - \nu\varepsilon_i) - g_{s_i^2}(\zeta_i)| \, \mathrm{d}\zeta_i &\leq \int_{\zeta_i} \int_{(0,\nu\varepsilon_i)} |g'_{s_i^2}(\zeta_i - \tau)| \, \mathrm{d}\tau \mathrm{d}\zeta_i \\ &\leq \int_{(0,\nu\varepsilon_i)} \left(\int_{\zeta_i} |g'_{s_i^2}(\zeta_i - \tau)| \, \mathrm{d}\zeta_i \right) \mathrm{d}\tau \\ &= \left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| \, \mathrm{d}t \right) \nu|\varepsilon_i| < +\infty \end{aligned}$$

since the derivative of the Gaussian density is integrable over $\mathbb{R}$.
Going back to the integrals over variables $\boldsymbol{\zeta}, \boldsymbol{\varepsilon} \in \mathbb{R}^P$ of Equation (124)

$$\begin{aligned} & \left\| \int_{\boldsymbol{\zeta}} \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} \left\langle u(\boldsymbol{\zeta}; \boldsymbol{\Lambda}), \boldsymbol{\varepsilon} \right\rangle \left(\mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta} - \nu\boldsymbol{\varepsilon}) - \mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta}) \right) \mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon}) \, \mathrm{d}\boldsymbol{\zeta} \mathrm{d}\boldsymbol{\varepsilon} \right\| \qquad (130) \\ & \leq \int_{\boldsymbol{\zeta}} \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} L_u \|\boldsymbol{\varepsilon}\| \left| \mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta} - \nu\boldsymbol{\varepsilon}) - \mathcal{G}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{\zeta}) \right| \mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon}) \, \mathrm{d}\boldsymbol{\zeta} \mathrm{d}\boldsymbol{\varepsilon} \\ & = \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} L_u \|\boldsymbol{\varepsilon}\| \prod_{i=1}^{P} \left(\int_{\zeta_i} |g_{s_i^2}(\zeta_i - \nu\varepsilon_i) - g_{s_i^2}(\zeta_i)| \, \mathrm{d}\zeta_i \right) \mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon}) \, \mathrm{d}\boldsymbol{\varepsilon} \\ & \leq \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} L_u \|\boldsymbol{\varepsilon}\| \prod_{i=1}^{P} \left(\left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| \, \mathrm{d}t \right) \nu|\varepsilon_i| \right) \mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon}) \, \mathrm{d}\boldsymbol{\varepsilon} \\ |\varepsilon_i| \leq \|\boldsymbol{\varepsilon}\| \quad & \leq \int_{\boldsymbol{\varepsilon}} 2\nu^{P-1} \|L_u \boldsymbol{\varepsilon}\| \|\boldsymbol{\varepsilon}^P\| \prod_{i=1}^{P} \left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| \, \mathrm{d}t \right) \mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon}) \, \mathrm{d}\boldsymbol{\varepsilon} \\ (0 < \nu \leq 1) \quad & \leq \int_{\boldsymbol{\varepsilon}} 2L_u \|\boldsymbol{\varepsilon}\|^{P+1} \prod_{i=1}^{P} \left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| \, \mathrm{d}t \right) \mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon}) \, \mathrm{d}\boldsymbol{\varepsilon} < +\infty \end{aligned}$$

Indeed, $\|\cdot\|$ being the euclidean norm $(\forall i) \, |\varepsilon_i| \leq \|\boldsymbol{\varepsilon}\|$. Moreover, since we are interested in the limit $\nu \to 0$, we can assume without loss of generality that $0 > \nu \leq 1$ and thus $\nu^{P-1} \leq 1$. We conclude using the fact that any power of $\|\boldsymbol{\varepsilon}\|$ is integrable against $\mathcal{G}_{\boldsymbol{I}}(\boldsymbol{\varepsilon})$, combined to the fact that the support $\mathbb{V}$ of φ is compact, which enable to apply Fubini's theorem to exchange $\int_{\mathbb{V}}$ and $\mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}}$.

(DC 2) The above majoration does not depends on ν. Further, the Lipschitzianity assumptions provides the existence P-*a.s.* of

$$\lim_{\nu \to 0} \mathbb{E}_{\boldsymbol{\zeta},\boldsymbol{\varepsilon}} \left(\partial_{\boldsymbol{\Lambda}} \widehat{R}_{\nu,\boldsymbol{\varepsilon}}(\boldsymbol{y}; \boldsymbol{\Lambda} | \boldsymbol{\mathcal{S}}) \right)_l$$

then dominated convergence theorem applies to invert $\lim_{\nu \to 0}$ and $\int_{\mathbb{V}}$ which completes the proof.

□

D Constant term of Stein Unbiased Risk Estimate

Proof. Because of the cyclicality of the trace, one has $\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*) = \mathrm{Tr}(\mathbf{A}^*\mathbf{A}\boldsymbol{\mathcal{S}})$, then using the definition of $\mathbf{A} \triangleq \boldsymbol{\Pi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^*$,

$$\mathbf{A}^*\mathbf{A} = \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Pi}^*\boldsymbol{\Pi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^*$$

then turning to the block-matrix form to perform the products

$$\begin{aligned}
&\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Pi}^*\boldsymbol{\Pi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1} \\
&= \frac{1}{\left(F_0F_2 - F_1^2\right)^2}\begin{pmatrix} F_0\boldsymbol{I}_{N/2} & -F_1\boldsymbol{I}_{N/2} \\ -F_1\boldsymbol{I}_{N/2} & F_2\boldsymbol{I}_{N/2}\end{pmatrix}\begin{pmatrix} \boldsymbol{I}_{N/2} & \boldsymbol{Z}_{N/2} \\ \boldsymbol{Z}_{N/2} & \boldsymbol{Z}_{N/2}\end{pmatrix}\begin{pmatrix} F_0\boldsymbol{I}_{N/2} & -F_1\boldsymbol{I}_{N/2} \\ -F_1\boldsymbol{I}_{N/2} & F_2\boldsymbol{I}_{N/2}\end{pmatrix} \\
&= \frac{1}{\left(F_0F_2 - F_1^2\right)^2}\begin{pmatrix} F_0\boldsymbol{I}_{N/2} & -F_1\boldsymbol{I}_{N/2} \\ -F_1\boldsymbol{I}_{N/2} & F_2\boldsymbol{I}_{N/2}\end{pmatrix}\begin{pmatrix} F_0\boldsymbol{Z}_{N/2} & -F_1\boldsymbol{I}_{N/2} \\ \boldsymbol{Z}_{N/2} & \boldsymbol{Z}_{N/2}\end{pmatrix} \\
&= \frac{1}{\left(F_0F_2 - F_1^2\right)^2}\begin{pmatrix} F_0^2\boldsymbol{I}_{N/2} & -F_0F_1\boldsymbol{I}_{N/2} \\ -F_0F_1\boldsymbol{I}_{N/2} & F_1^2\boldsymbol{I}_{N/2}\end{pmatrix}.
\end{aligned}$$

Using again cyclicality of the trace

$$\begin{aligned}
\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*) &= \mathrm{Tr}\left(\boldsymbol{\Phi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Pi}^*\boldsymbol{\Pi}\left(\boldsymbol{\Phi}^*\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^*\boldsymbol{\mathcal{S}}\right) \\
&= \mathrm{Tr}\left(\boldsymbol{\Phi}\frac{1}{\left(F_0F_2 - F_1^2\right)^2}\begin{pmatrix} F_0^2\boldsymbol{I}_{N/2} & -F_0F_1\boldsymbol{I}_{N/2} \\ -F_0F_1\boldsymbol{I}_{N/2} & F_1^2\boldsymbol{I}_{N/2}\end{pmatrix}\boldsymbol{\Phi}^*\boldsymbol{\mathcal{S}}\right) \\
&= \mathrm{Tr}\left(\frac{1}{\left(F_0F_2 - F_1^2\right)^2}\begin{pmatrix} F_0^2\boldsymbol{I}_{N/2} & -F_0F_1\boldsymbol{I}_{N/2} \\ -F_0F_1\boldsymbol{I}_{N/2} & F_1^2\boldsymbol{I}_{N/2}\end{pmatrix}\boldsymbol{\Phi}^*\boldsymbol{\mathcal{S}}\boldsymbol{\Phi}\right)
\end{aligned}$$

Then using the action of $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$, explicited in Formula (66) we have the matrix representation

$$\boldsymbol{\Phi} = \begin{pmatrix} 1\boldsymbol{I}_{N/2} & \boldsymbol{I}_{N/2} \\ 2\boldsymbol{I}_{N/2} & \boldsymbol{I}_{N/2} \\ \vdots & \vdots \\ J\boldsymbol{I}_{N/2} & \boldsymbol{I}_{N/2}\end{pmatrix} \in \mathbb{R}^{JN_1N_2 \times 2N_1N2}. \tag{131}$$

Using of the decomposition of $\boldsymbol{\mathcal{S}}$ into J^2 blocks $\boldsymbol{\mathcal{S}}_j^{j'} = \mathcal{C}_j^{j'}\boldsymbol{\Xi}_j^{j'} \in \mathbb{R}^{N/2\times N/2}$, we obtain

$$\boldsymbol{\Phi}^*\boldsymbol{\mathcal{S}}\boldsymbol{\Phi} = \begin{pmatrix} \sum_{j,j'} jj'\boldsymbol{\mathcal{S}}_j^{j'} & \sum_{j,j'} j'\boldsymbol{\mathcal{S}}_j^{j'} \\ \sum_{j,j'} j\boldsymbol{\mathcal{S}}_j^{j'} & \sum_{j,j'} \boldsymbol{\mathcal{S}}_j^{j'}\end{pmatrix}, \quad 1 \leq j, j' \leq J. \tag{132}$$

It follows

$$
\begin{aligned}
&\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*) \\
&= \mathrm{Tr}\left(\frac{1}{\left(F_0F_2 - F_1^2\right)^2} \begin{pmatrix} F_0^2\boldsymbol{I}_{N/2} & -F_0F_1\boldsymbol{I}_{N/2} \\ -F_0F_1\boldsymbol{I}_{N/2} & F_1^2\boldsymbol{I}_{N/2} \end{pmatrix} \begin{pmatrix} \sum_{j,j'} jj'\boldsymbol{\mathcal{S}}_j^{j'} & \sum_{j,j'} j'\boldsymbol{\mathcal{S}}_j^{j'} \\ \sum_{j,j'} j\boldsymbol{\mathcal{S}}_j^{j'} & \sum_{j,j'} \boldsymbol{\mathcal{S}}_j^{j'} \end{pmatrix} \right) \\
&= \frac{1}{\left(F_0F_2 - F_1^2\right)^2} \mathrm{Tr} \begin{pmatrix} \sum_{j,j'} F_0^2 jj'\boldsymbol{\mathcal{S}}_j^{j'} - F_0F_1 j'\boldsymbol{\mathcal{S}}_j^{j'} & \sum_{j,j'} F_0^2 j'\boldsymbol{\mathcal{S}}_j^{j'} - F_0F_1\boldsymbol{\mathcal{S}}_j^{j'} \\ \sum_{j,j'} F_1^2 j\boldsymbol{\mathcal{S}}_j^{j'} - F_0F_1 jj'\boldsymbol{\mathcal{S}}_j^{j'} & \sum_{j,j'} F_1^2\boldsymbol{\mathcal{S}}_j^{j'} - F_0F_1 j'\boldsymbol{\mathcal{S}}_j^{j'} \end{pmatrix} \\
&= \frac{1}{\left(F_0F_2 - F_1^2\right)^2} \mathrm{Tr} \left(\sum_{j,j'} \left(F_0^2 jj'\boldsymbol{\mathcal{S}}_j^{j'} - F_0F_1 j'\boldsymbol{\mathcal{S}}_j^{j'} + F_1^2\boldsymbol{\mathcal{S}}_j^{j'} - F_0F_1 j'\boldsymbol{\mathcal{S}}_j^{j'} \right) \right).
\end{aligned}
\tag{133}
$$

Then, one can remark that

$$
\mathrm{Tr}(\boldsymbol{\mathcal{S}}_j^{j'}) \triangleq \sum_{\underline{n}\in\Omega} \mathcal{S}_{j,\underline{n}}^{j',\underline{n}} = \sum_{\underline{n}\in\Omega} \mathcal{C}_j^{j'} = \frac{N}{2}\mathcal{C}_j^{j'}. \tag{134}
$$

since the filter $\boldsymbol{\Xi}_j^{j'}$ is supposed to be normalized, in the sense that its maximum value equals 1. Finally

$$
\mathrm{Tr}(\mathbf{A}\boldsymbol{\mathcal{S}}\mathbf{A}^*) = \frac{N/2}{\left(F_0F_2 - F_1^2\right)^2} \sum_{j,j'} \left(F_0^2 jj'\mathcal{C}_j^{j'} - F_0F_1 j'\mathcal{C}_j^{j'} + F_1^2\mathcal{C}_j^{j'} - F_0F_1 j'\mathcal{C}_j^{j'} \right). \tag{135}
$$

□

References

[1] A. C. Aitkin. On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1935.

[2] R. Ammanouil, A. Ferrari, D. Mary, C. Ferrari, and F. Loi. A parallel and automatically tuned algorithm for multispectral image deconvolution. *Monthly Notices of the Royal Astronomical Society*, 490(1):37–49, 2019.

[3] J.-F. Aujol, G. Gilboa, T.F. Chan, and S. Osher. Structure-texture image decomposition–modeling, algorithms, and parameter selection. *Int. J. Comp. Vis.*, 67(1):111–136, 2006.

[4] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[6] J. Berger. Minimax estimation of a multivariate normal mean under arbitrary quadratic loss. *J. Multivariate Anal.*, 6(2):256–264, 1976.

[7] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[8] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. Atlanta, USA, June 16–21 2013. Jmlr.

[9] Q. Bertrand, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of Lasso-type models for hyperparameter optimization, 2020.

[10] Å. Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[11] T. Blu and F. Luisier. The SURE-LET approach to image denoising. *IEEE Trans. Image Process.*, 16(11):2778–2786, 2007.

[12] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *J. Sci. Comput.*, 16(5):1190–1208, 1995.

[13] J.-F. Cai, B. Dong, S. Osher, and Z. Shen. Image restoration: Total variation, wavelet frames, and beyond. *J. Amer. Math. Soc.*, 25:1033–1089, May 2012.

[14] X. Cai, R. Chan, C.-B. Schonlieb, G. Steidl, and T. Zeng. Linkage between piecewise constant mumford-shah model and rof model and its virtue in image segmentation. *arXiv preprint arXiv:1807.10194*, 2018.

[15] X. Cai and G. Steidl. Multiclass segmentation by iterated rof thresholding. In *Int. Workshop on Energy Minimization Methods in Comp. Vis. and Pat. Rec.*, pages 237–250. Springer, 2013.

[16] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.*, 40(1):120–145, 2011.

[17] T.F. Chan and J.J. Shen. Variational image inpainting. *Comm. Pure Applied Math.*, 58(5), May 2005.

[18] C. Chaux and L. Blanc-Féraud. Estimation d'hyperparamètres pour la résolution de problèmes inverses à l'aide d'ondelettes. In *XXIIe colloque GRETSI (Traitement du Signal et des Images)*, Dijon, France, Sept. 8–11 2009. GRETSI, Groupe dEtudes du Traitement du Signal et des Images.

[19] C. Chaux, L. Duval, A. Benazza-Benyahia, and J.-C. Pesquet. A nonlinear stein-based estimator for multichannel image denoising. *IEEE Trans. Signal Process.*, 56(8):3855–3870, 2008.

[20] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, New York, 2011.

[21] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. Proximal splitting algorithms: Relax them all!, 2019.

[22] F. E. Curtis, T. Mitchell, and M. L. Overton. A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optim. Methods Softw.*, 32(1):148–181, 2017.

[23] C.-A. Deledalle, F. Tupin, and L. Denis. Poisson NL means: Unsupervised non local means for Poisson noise. In *Proc. Int. Conf. Image Process.*, pages 801–804. IEEE, 2010.

[24] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré. Stein unbiased gradient estimator of the risk (SUGAR) for multiple parameter selection. *SIAM J. Imaging Sci.*, 7(4):2448–2487, 2014.

[25] L. Desbat and D. Girard. The "minimum reconstruction error" choice of regularization parameters: Some more efficient methods and their application to deconvolution problems. *J. Sci. Comput.*, 16(6):1387–1403, 1995.

[26] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *J. American Statist. Ass.*, 90(432):1200–1224, 1995.

[27] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

[28] C. Dossal, M. Kachour, J. Fadili, G. Peyré, and C. Chesneau. The degrees of freedom of the Lasso for general design matrix. *Statistica Sinica*, pages 809–828, 2013.

[29] B. Efron. How biased is the apparent error rate of a prediction rule? *J. Am. Stat. Assoc.*, 81(394):461–470, 1986.

[30] Yonina C Eldar. Generalized SURE for exponential families: Applications to regularization. *IEEE Trans. Signal Process.*, 57(2):471–481, 2008.

[31] L. Elden. Algorithms for the regularization of ill-conditioned least squares problems. *BIT Numer. Math.*, 17(2):134–145, 1977.

[32] L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. Chapman and Hall/CRC, 2015.

[33] N. P. Galatsanos and A. K. Katsaggelos. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Trans. Image Process.*, 1(3):322–336, 1992.

[34] A. Girard. A fast Monte-Carlo cross-validationprocedure for large least squares problems with noisy data. *Numerische Mathematik*, 56(1):1–23, 1989.

[35] G. H. Golub, P. C. Hansen, and D. P. O'Leary. Tikhonov regularization and total least squares. *SIAM J. Matrix Anal. Appl.*, 21(1):185–194, 1999.

[36] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[37] P. C. Hansen and D. P. OLeary. The use of the L-curve in the regularization of discrete ill-posed problems. *J. Sci. Comput.*, 14(6):1487–1503, 1993.

[38] H. M. Hudson. A natural identity for exponential families with applications in multiparameter estimation. *Ann. Stat.*, 6(3):473–484, 1978.

[39] H. M. Hudson and T. C. M. Lee. Maximum likelihood restoration and choice of smoothing parameter in deconvolution of image data subject to Poisson noise. *Computational statistics & data analysis*, 26(4):393–410, 1998.

[40] K. Kato. On the degrees of freedom in shrinkage estimation. *J. Multivariate Anal.*, 100(7):1338–1352, 2009.

[41] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, volume 15. SIAM, 1995.

[42] Y. Le Montagner, E. D. Angelini, and J.-C. Olivo-Marin. An unbiased risk estimator for image denoising in the presence of mixed poisson–gaussian noise. *IEEE Trans. Image Process.*, 23(3):1255–1268, 2014.

[43] K. C. Li. From stein's unbiased risk estimates to the method of generalized cross validation. *Ann. Stat.*, pages 1352–1377, 1985.

[44] F. Luisier, T. Blu, and M. Unser. A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding. *IEEE Trans. Image Process.*, 16(3):593–606, 2007.

[45] F. Luisier, T. Blu, and M. Unser. Image denoising in mixed poisson–gaussian noise. *IEEE Trans. Image Process.*, 20(3):696–708, 2010.

[46] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, Inc., Orlando, FL, USA, 3rd edition, 2008.

[47] M. Meyer and M. Woodroofe. On the degrees of freedom in shape-restricted regression. *Ann. Stat.*, pages 1083–1104, 2000.

[48] J. D. B. Nelson, C. Nafornita, and A. Isar. Semi-local scaling exponent estimation with box-penalty constraints and total-variation regularization. *IEEE Trans. Image Process.*, 25(7):3167–3181, 2016.

[49] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[50] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[51] B. Pascal, N. Pustelnik, and P. Abry. Nonsmooth Convex Joint Estimation of Local Regularity and Local Variance for Fractal Texture Segmentation. *arXiv e-prints*, page arXiv:1910.05246, Oct 2019.

[52] Barbara Pascal, Nelly Pustelnik, Patrice Abry, Marion Serres, and Valérie Vidal. Joint estimation of local variance and local regularity for texture segmentation. application to multiphase flow characterization. In *Proc. Int. Conf. Image Process.*, pages 2092–2096. IEEE, 2018.

[53] J.-C. Pesquet, A. Benazza-Benyahia, and C. Chaux. A SURE approach for digital signal/image deconvolution problems. *IEEE Trans. Signal Process.*, 57(12):4616–4632, 2009.

[54] O. Pont, A. Turiel, and H. Yahia. An optimized algorithm for the evaluation of local singularity exponents in digital signals. In *International Workshop on Combinatorial Image Analysis*, pages 346–357. Springer, 2011.

[55] N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet. Wavelet-based image deconvolution and reconstruction. *Wiley Encyclopedia of Electrical and Electronics Engineering*, Feb. 2016.

[56] N. Pustelnik, H. Wendt, P. Abry, and N. Dobigeon. Combining local regularity estimation and total variation optimization for scale-free texture segmentation. *IEEE Trans. Computational Imaging*, 2(4):468–479, 2016.

[57] S. Ramani, T. Blu, and M. Unser. Monte-carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Trans. Image Process.*, 17(9):1540–1554, 2008.

[58] R. T. Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

[59] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[60] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Stat.*, pages 1135–1151, 1981.

[61] T. Strutz. *Data fitting and uncertainty: A practical introduction to weighted least squares and beyond*. Vieweg and Teubner, 2010.

[62] A. M. Thompson, J. C. Brown, J. W Kay, and D. M. Titterington. A study of methods of choosing the smoothing parameter in image restoration by regularization. *IEEE Trans. Pattern Anal. Match. Int.*, (4):326–339, 1991.

[63] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[64] R. Tibshirani and L. Wasserman. Steins unbiased risk estimate. *Course notes from "Statistical Machine Learning"*, pages 1–12, 2015.

[65] R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *Ann. Stat.*, 40(2):1198–1232, 2012.

[66] A. Tikhonov. Tikhonov regularization of incorrectly posed problems. *Soviet Mathematics Doklady*, 4:1624–1627, 1963.

[67] S. Vaiter, C. Deledalle, J. Fadili, G. Peyré, and C. Dossal. The degrees of freedom of partly smooth regularizers. *Ann. Inst. Stat. Math*, 69(4):791–832, 2017.

[68] Darryl Veitch and Patrice Abry. A wavelet-based joint estimator of the parameters of long-range dependence. *IEEE Trans. Inform. Theory*, 45(3):878–897, 1999.

[69] C. Vonesch, S. Ramani, and M. Unser. Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint. In *Proc. Int. Conf. Image Process.*, pages 665–668. IEEE, 2008.

[70] H. Wendt. *Contributions of Wavelet Leaders and Bootstrap to Multifractal Analysis: Images, Estimation Performance, Dependence Structure and Vanishing Moments. Confidence Intervals and Hypothesis Tests.* PhD thesis, Ecole Normale Suprieure de Lyon, 2008.

[71] H. Wendt, P. Abry, S. Jaffard, H. Ji, and Z. Shen. Wavelet leader multifractal analysis for texture classification. In *Proc. Int. Conf. Image Process.*, pages 3829–3832. IEEE, 2009.

[72] H. Wendt, S. G. Roux, P. Abry, and S. Jaffard. Wavelet leaders and bootstrap for multifractal analysis of images. *Signal Process.*, 89(6):1100–1114, 2009.

[73] X. Xie, S.C. Kou, and L. D. Brown. Sure estimates for a heteroscedastic hierarchical model. *J. American Statist. Ass.*, 107(500):1465–1479, 2012.