WARP: Wavelets with adaptive recursive partitioning for multi-dimensional data

Meng Li¹ and Li Ma²

¹Department of Statistics, Rice University
²Department of Statistical Science, Duke University

July 11, 2018

Abstract

Traditional statistical wavelet analysis carries out modeling and inference under a given, predetermined wavelet transform. This approach can quickly lose efficiency for multi-dimensional data (e.g., observations measured on a multi-dimensional grid), because a predetermined wavelet transform does not exploit the structure of the underlying function in a problem-specific manner. Too much is already lost at the wavelet transform stage before any statistical remedy takes place. A number of non-wavelet methods aim at addressing this difficulty, at the cost of some of the most desirable statistical and computational benefits of wavelet methods. This work aims to overcome this challenge within the wavelet framework by making the wavelet transform itself adaptive to the structure of the data. By exploiting a connection between permutations on the index space of multi-dimensional functions and recursive partitions on that space, we show that the desired adaptivity in the wavelet transform can be achieved through a layer of Bayesian hierarchical modeling on the space of such recursive partitions. When one applies this framework to Haar wavelets, exact Bayesian inference under the model can be achieved analytically through a recursive message passing algorithm with an efficient computational complexity linear in the sample size. We also provide recipes for incorporating block shrinkage into the framework as well as for applying it to other wavelet bases. We demonstrate via numerical experiments that with the enhancement under this framework even simple 1D Haar wavelets can achieve excellent performance in the context of 2D and 3D image reconstruction, outperforming state-of-the-art wavelet and non-wavelet methods especially in noisy, low signal-to-noise ratio settings at a fraction of their computational cost. Furthermore, we investigate the source of the gain by quantitatively comparing the efficacy of energy concentration under our adaptive wavelet transforms to that of classical fixed wavelet transforms.
1 Introduction

Multi-resolution statistical analyses based on wavelet transforms (Donoho and Johnstone, 1994, 1995) have been successfully applied in many applications; see Mallat (2008) for some examples. Wavelet analyses often enjoy both good statistical performance and high computational scalability. An important property of wavelet bases is “energy concentration”—under wavelet transforms the information content in a functional observation is often “condensed” into a small number of wavelet coefficients, thereby allowing “signals” to be more readily separated from “noise”. The statistical performance of wavelet analyses largely depends on the effectiveness of energy concentration.

Traditional statistical wavelet analyses, from both frequentist and Bayesian perspectives, typically start with a given wavelet transform of the original data, and then focus on effective modeling and inference on the resulting wavelet coefficients (Abramovich et al., 1998; Brown et al., 2001; Clyde and George, 2000; Crouse et al., 1998; Morris and Carroll, 2006; Willett and Nowak, 2004). However, when the data are multi-dimensional, a predetermined wavelet transform typically does not sufficiently exploit the structure of the data, and thus much has already been lost at the wavelet transform stage before any statistical efforts can come to rescue. This limitation of classical wavelet analysis limits its applicability in a wide range of applications involving multi-dimensional data. The detrimental effect of such lack of adaptivity has been noted previously by a number of authors (Fryzlewicz and Timmermans, 2016; Le Pennec and Mallat, 2005; Peyré, 2011; Polzehl and Spokoiny, 2000), who also presented alternatives to the traditional wavelet analyses to address the challenge.

Given the desirable performance of wavelets and its substantial computational efficiency, our objective herein is to restore the effectiveness of classical wavelet analyses in multi-dimensional problems without destroying its desirable features, especially in terms of computational efficiency. It turns out that this challenge can be addressed within statistical wavelet analysis by simply starting the modeling and inference “one level up”—that is, by incorporating the choice of the wavelet transform itself into the statistical analysis, which can then allow the wavelet transform to adapt to the data at hand thereby substantially improving the resulting “energy concentration”.

For example, suppose our observation is a noisy image of the American national flag. An effective way to represent this function, if known, would first split the image into the portion with the stripes and another with the stars and deal with them differently. The portion with the stripes is effectively a “one-dimensional” function, as its value is constant across any horizontal line, and to represent the information among the stripes, a one-dimensional function is sufficient. One can thus achieve effective “energy concentration” through a 1D wavelet transform along the vertical direction, applied on the average value of each horizontal line as the value of that 1D function. On the other hand, efficiently representing the portion containing the stars calls for a decomposition in both the horizontal and vertical directions. Most real applications involve observations with less clear-cut structure, but exploiting the local characteristics in multi-dimensional observations can generally improve the analysis.

To achieve this goal, we utilize a 1D wavelet basis and allow it to “turn and twist” (or “warp”) over the index space depending the structure of the underlying function. (In
contrast, a classical multidimensional wavelet basis would decompose the function symmetrically in the dimensions.) While there are different ways for incorporating such “warping”, we adopt a hierarchical Bayesian approach through placing a prior on the local directionality of the 1D transform. Specifically, we show that “warping” the 1D wavelet transform is equivalent to fixing the wavelet basis while shuffling points in the multivariate index space of the function—i.e., through applying a given 1D wavelet transform to a permuted version of the observation. This connection implies that priors on the permutations of the functional index induce priors on the “warped” wavelet transform.

The space of all permutations, however, is massive, and therefore the choice of priors on the permutations must strike a balance between flexibility (i.e., supported on a large enough class of permutations) and practicability (i.e., the availability of efficient inference algorithms). A particular class of priors on the permutations—induced by recursive partitioning over the index space of the functions—achieves this desired balance. With this choice, existing modeling and inference tools for recursive partitioning can be adopted to achieve effective inference. For this reason, we shall refer to our framework as WARP, or wavelets with adaptive recursive partitioning.

Interestingly, we show that when the Haar basis is adopted in the 1D wavelet transform, the resulting posterior of a number of well-known Bayesian wavelet regression models, when generalized with WARP, can be computed analytically through exact message passing. The computation is extremely efficient—scaling linearly with the size of the index space, which is the same computational complexity as carrying out a standard 1D wavelet transform through Mallat’s pyramid algorithm. Such scalability makes the framework suitable for applications such as high-resolution image analysis and other large-scale problems. Also, WARP provides a unified framework for general m-dimensional functions—the same model applies for any dimensionality \( m \geq 2 \), in contrast to existing approaches that are either dimensionality-specific or at least require substantial modification for different values of \( m \).

The rest of the paper is organized as follows. Section 2 introduces the WARP framework. In Section 2.1 we review the key components of Bayesian wavelet regression models, introduce permutation of the index space as a way to incorporate adaptivity into wavelet analysis, and construct a class of priors on the permutation induced by recursive dyadic partitioning on the index space. We derive the corresponding posterior distributions and provide computational recipes for exact Bayesian inference under the Haar basis in Section 2.2. In Section 3, we carry out an extensive numerical study and compare our method to existing state-of-the-art wavelet and non-wavelet methods using a variety of 2D and 3D images. Section 4 concludes with some brief remarks. The Supplementary Materials contain all proofs, a sequential Monte Carlo algorithm for applying WARP on wavelet bases other than the Haar basis, a sensitivity analysis on the prior specification, and additional numerical experiments using 3D images. The C++ source code along with a Matlab toolbox and R package to implement the proposed method is available online at https://github.com/MaStatLab/WARP.
2 Method

2.1 Permuted wavelet regression and recursive dyadic partitions

We shall use \( \Omega \) to denote a space of indices or locations (e.g., pixels in images) where we obtain numerical measurements (e.g., intensities of pixels). Throughout this work, we assume \( \Omega \) to be an \( m \)-dimensional rectangular tube consisting of \( n_i = 2^{J_i} \) grid points in the \( i \)th dimension for \( i = 1, 2, \ldots, m \), that is, the function values are observed on a multi-dimensional equidistant grid. To simplify notation, we shall use \([a,b]\) to represent the set \( \{a,a+1,\ldots,b\} \) for two integers \( a \) and \( b \) with \( a \leq b \). Then the index space \( \Omega \) is of the form

\[
\Omega = [0,2^{J_1}-1] \times [0,2^{J_2}-1] \times \cdots \times [0,2^{J_m}-1].
\]

The locations in \( \Omega \) can be placed into a vector of length \( n = 2^{J_1} \). For example, we can map the location \( s = (s_1, s_2, \ldots, s_m) \in \Omega \) to the \( t \)th element in the vector, where \( t = s_1 + \sum_{i=2}^{m} (\prod_{l=1}^{i-1} n_l) s_i \). Correspondingly, any function \( f : \Omega \to \mathbb{R} \) can be represented as a vector \( f \) of length \( n = 2^{J} \) whose \( t \)th element is \( f(s) \). It may not seem obvious at first why one would want to treat a multi-dimensional function as a one-dimensional vector. We will show later that adaptive wavelet transform on multi-dimensional functions can be achieved through 1D wavelet transforms applied to adaptively permuted vectors.

Now, we consider the regression model

\[
y = f + \epsilon \quad \text{with} \quad \epsilon \sim N(0, \Sigma),
\]

where \( y = (y_0, y_1, \ldots, y_{2^{J}-1})' \) are the observation values made on \( \Omega \), \( f = (f_0, f_1, \ldots, f_{2^{J}-1})' \) the underlying unknown function mean (or the signal), and \( \epsilon = (\epsilon_0, \epsilon_1, \ldots, \epsilon_{2^{J}-1})' \) the noise. For ease of illustration, we assume homogeneous white noise, i.e., \( \Sigma_\epsilon = \sigma^2 I_n \), though our method does not rely on this assumption and can readily apply to models with heterogeneous variance; see Section 4 for further discussion.

Wavelet analysis starts by applying a discrete wavelet transform (DWT) to \( y \). This can be done by multiplying a corresponding orthonormal matrix \( W \) to both sides of Eq. (1), obtaining \( w = z + u \) where \( w = W y \) is the vector of empirical wavelet coefficients, \( z = W f \) the mean vector for wavelet coefficients and \( u = W \epsilon \) the noise vector in the wavelet domain. This model can be rewritten in a location-scale form:

\[
w_{j,k} = z_{j,k} + u_{j,k} \quad \text{for} \quad j = 0, 1, \ldots, J-1 \quad \text{and} \quad k = 0, 1, \ldots, 2^j - 1,
\]

where \( w_{j,k}, z_{j,k}, u_{j,k} \) are the \( k \)th wavelet coefficient, signal, and noise at the \( j \)th scale in the wavelet (i.e., location-scale) domain, respectively.

Obviously, it would not be reasonable to simply treat multi-dimensional observations as a vector, and apply 1D wavelet regression analysis to an arbitrarily given fixed vectorization; see Ali et al. (2014); Donoho (1999); Jacques et al. (2011). Such a vectorization ignores the structure of the underlying function, and thus will result in less effective energy concentration, producing a wavelet decomposition of \( f \) that is not very sparse—with many non-zero \( z_{j,k} \)'s of small to moderate sizes, reducing the signal-to-noise ratio at those \((j,k)\) combinations.
2.1.1 Recursive dyadic partitioning on the location space

A partition of $\Omega$ is a collection of nonempty sets $\{A_1, A_2, \ldots, A_H\}$ such that $\Omega = \bigcup_{h=1}^{H} A_h$ and $A_{h_1} \cap A_{h_2} = \emptyset$ for any $h_1 \neq h_2$. Now let $\mathcal{T}^0, \mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^j, \ldots$ be a sequence of partitions of $\Omega$. We say that this sequence is a recursive dyadic partition (RDP) if it satisfies the following two conditions: (i) $\mathcal{T}^j$ consists of $2^j$ blocks: $\mathcal{T}^j = \{A_{j,k} : k = 0, 1, \ldots, 2^j - 1\}$; (ii) $\mathcal{T}^{j+1}$ is obtained by dividing each set in $\mathcal{T}^j$ into two pieces, i.e., $A_{j,k} = A_{j+1,2k} \cup A_{j+1,2k+1}$ for all $j \geq 0$ and $k = 0, 1, \ldots, 2^j - 1$.

We call an RDP canonical if the sequence of partitions satisfy two additional conditions: (iii) if the partition blocks $A_{j,k}$ are rectangles of the form

$$A_{j,k} = [a_{j,k}^{(1)}, b_{j,k}^{(1)}] \times [a_{j,k}^{(2)}, b_{j,k}^{(2)}] \times \cdots \times [a_{j,k}^{(m)}, b_{j,k}^{(m)}],$$

and (iv) $A_{j+1,2k}$ and $A_{j+1,2k+1}$ are produced by dividing $A_{j,k}$ into two halves at the middle of one of $A_{j,k}$’s divisible dimensions.

A rectangular partition block $A_{j,k}$ is divisible in dimension $d$ if $A_{j,k}$ is supported on at least two values in that dimension, i.e., $a_{j,k}^{(d)} < b_{j,k}^{(d)}$. In this case, if $A_{j,k}$ is divided in dimension $d$, then its children $A_{j+1,2k}$ and $A_{j+1,2k+1}$ are given by

$$[a_{j+1,2k}^{(d)}, b_{j+1,2k}^{(d)}] = [a_{j,k}^{(d)}, (a_{j,k}^{(d)} + b_{j,k}^{(d)})/2] \quad \text{and} \quad [a_{j+1,2k+1}^{(d)}, b_{j+1,2k+1}^{(d)}] = [(a_{j,k}^{(d)} + b_{j,k}^{(d)})/2 + 1, b_{j,k}^{(d)}]$$

while

$$[a_{j+1,2k}^{(d')}, b_{j+1,2k}^{(d')}] = [a_{j+1,2k+1}^{(d')}, b_{j+1,2k+1}^{(d')}],$$

for all $d' \neq d$. 

Fortunately, for each specific data set at hand, there typically do exist some permutations that effectively reorganize the data so that the resulting 1D wavelet coefficients provides an efficient representation of the underlying function. See Figure 1 for an illustration. Adopting a model choice viewpoint, one can think of the wavelet regression model under each index permutation as a competing generative model for the observed data. This perspective inspires us to incorporate a prior on the permutations, thereby allowing us to compute a posterior on the space of competing wavelet regression models, and then to either select some good models (Raftery, 1995) or average over the models (Hoeting et al., 1999).

This approach does incur a common challenge in high-dimensional Bayesian model choice—that the space of all permutations is massive, and brute-force enumeration of the space is computationally impractical. In the current context, effective exploration of the model space becomes practical, however, once we realize that the vast majority of the permutations will lead to wavelet regression models that ignore the spatial smoothness of the underlying function—i.e., close locations in $\Omega$ often correspond to similar values in $f$. As such, we shall focus attention on a subclass of permutations that to various extents preserve smoothness, and design a model space prior supported on this subclass. To this end, we appeal to a relationship between recursive dyadic partitioning (RDP) on $\Omega$ and permutations, and shall consider the collection of permutations induced by RDPs. Next we introduce some basic notions regarding RDPs on $\Omega$, which is then used to construct a prior on permutations.
Any canonical RDP on $\Omega$ will have exactly $J + 1$ levels, i.e., $\mathcal{T}^0, \mathcal{T}^1, \ldots, \mathcal{T}^J$. The $j$th level partition $\mathcal{T}^j$ consists of $2^j$ rectangular pieces of equal size, each covering $n/2^j$ locations in $\Omega$. From now on, we simply use RDP to refer to canonical ones when this causes no confusion.

2.1.2 RDPs and permutations

Each RDP can be represented by a $J$ level bifurcating tree with the partition blocks in $\mathcal{T}^j$ forming the $2^j$ nodes in the $j$th level of the tree. As such, we can use $\mathcal{T} = \bigcup_{j=0}^J \mathcal{T}^j$ to represent the RDP. Each node in the $J$th level corresponds to a unique location in $\Omega$. We shall interchangeably refer to an RDP as a tree, and to the partition blocks as nodes.

Given the RDP $\mathcal{T}$, each location $s \in \Omega$ falls into a unique branch of $\mathcal{T}$, that is, $\Omega = A_0, 0 \supset A_1, k_1(s) \supset A_2, k_2(s) \supset \cdots \supset A_J, k_J(s) = \{s\}$, with $A_{j, k_j}(s)$ being the node in the $j$th level to which $s$ belongs. Accordingly, the RDP $\mathcal{T}$ induces a unique vectorization of the locations in $\Omega$ such that $s$ corresponds to the $t(s)$th element of the vector where $t(s) = \sum_{l=1}^J 2^{J-l} e_l(s)$ with $e_l = k_l(s) \mod 2$, indicating the branch of the tree $s$ falls into at level $l$. As such, $\mathcal{T}$ induces a permutation of the $n$ locations, and we let $\pi_{\mathcal{T}}$ denote this permutation.

As an illustration, Figure 1 presents an RDP and the induced permutation using a toy $4 \times 4$ image (so $m = 2$ and $J_1 = J_2 = 2$). We index pixels in the true image from 0 to 15. In addition, we assume that the underlying function takes only two values—0 and 1—on the 16 locations, represented by the white and the red colors respectively. The demonstrated RDP corresponds well to the structure of the underlying signal, which would result in an effective 1D wavelet analysis on the vectorized observation.

We shall now utilize the relationship between RDPs and permutations to construct a prior on the latter. Before that, we shall simplify our notations a little. Note that while what the $(j, k)$th node $A_{j, k}$ is depends on the RDP $\mathcal{T}$, different RDP trees can share common nodes—the $(j, k)$th node in one $\mathcal{T}$ may be the same as the $(j, k')$th node in another. (Note that the level of the node must be the same in either RDP.) In the following, we will need to specify quantities that only depends on the node regardless of the RDP tree $\mathcal{T}$ it arises from. A succinct way for expressing such quantities is to write them as a mapping from $\mathcal{A}$ to $\mathbb{R}$, where $\mathcal{A}$ denotes the finite collection of all sets that could be nodes in some RDP, or equivalently, $\mathcal{A}$ is the totality of nodes in all RDPs. (This is to be distinguished from the collection of nodes in any particular RDP, which is denoted by $\mathcal{T}$.) For example, we may define $\rho_{j, k}$ in a way that its value only depends on what the set $A_{j, k}$ is, regardless of the RDP $\mathcal{T}$ to which it belongs. In this case we can let $\rho_{j, k} = \rho(A_{j, k})$, where $\rho(\cdot)$ is a mapping form $\mathcal{A}$ to $[0, 1]$.

The mapping-based notation such as $\rho(\cdot)$ allows various parameters to be specified in a node-specific (rather than RDP-specific) manner. This is critical as we show later that the space of nodes $\mathcal{A}$ for all canonical RDPs is of a cardinality linear in the size of $\Omega$, while that of canonical RDPs is exponential in $n$. (See Proposition 1 in the Supplementary Materials.) Therefore carrying out computation in a node-specific manner is key to achieving linear complexity. Moreover, this notation will also help elucidate derivations on the posterior.
First, $T_n$ to very efficient posterior inference algorithms that scales linearly in the following generative prior on the RDP (Ma, 2013; Wong and Ma, 2010), which will lead Bayesian approach, by placing a hyperprior on the RDP.

Our strategy of representing multi-dimensional functions using vectors will only pay off if the vectorization of $\Omega$ can result in efficient characterization of the data, thereby leading to stronger energy concentration under wavelet transforms. For example, the RDP illustrated in Figure 1 will lead to particularly efficient inference of the corresponding function. In general, the true optimal vectorization is unknown, and one shall rely on the data to learn the RDPs that induce “good” vectorizations. Next we shall achieve this in hierarchical Bayesian approach, by placing a hyperprior on the RDP.

Several priors on recursive partitions have been proposed in the literature. We consider the following generative prior on the RDP (Ma, 2013; Wong and Ma, 2010), which will lead to very efficient posterior inference algorithms that scales linearly in $n$, the size of $\Omega$.

We describe the prior as a very simple generative procedure in an inductive manner. First, $T^0 = \{\Omega\}$ by construction. Now suppose we have generated $T^0, T^1, \cdots, T^j$ for some
$0 \leq j \leq J - 1$, then $T^{j+1}$ is generated as follows. For each $A_{j,k} \in T^j$, let $D(A_{j,k}) \subset \{1,2,\ldots,m\}$ be the collection of its divisible dimensions. We randomly draw a dimension in $D(A_{j,k})$, and divide $A_{j,k}$ in that dimension to get $A_{j+1,2k}$ and $A_{j+1,2k+1}$. In particular, we let $\lambda_d(A_{j,k})$ be the probability for drawing the $d$th dimension, where $\sum_{d=1}^m \lambda_d(A_{j,k}) = 1$ and $\lambda_d(A_{j,k}) = 0$ for $d \notin D(A_{j,k})$. In many problems, a priori one has no reason to favor dividing any particular dimension over another, and a default specification is to set

$$\lambda_d(A_{j,k}) = 1/|D(A_{j,k})| \cdot 1_{\{d \in D(A_{j,k})\}}.$$  

This completes the inductive generation of $T^{j+1}$. The procedure will terminate after $T^J$ is generated as all nodes in $T^J$ are atomic with no divisible dimensions.

The above generative mechanism forms a probability distribution on the space of RDPs, which is called the random recursive dyadic partition (RRDP) distribution, and it is specified by the collection of selection probabilities defined on all potential nodes. We write

$$T \sim \text{RRDP}(\lambda),$$

where $\{\lambda(A) : A \in \mathcal{A}\}$, and $\lambda(A) = (\lambda_1(A), \lambda_2(A), \ldots, \lambda_m(A))'$, that is, $\lambda$ is a mapping from $\mathcal{A}$ to the $(m-1)$-dimensional simplex.

It is worth noting that the RRDP is nothing but a restrictive version of the Bayesian classification and regression tree (CART) prior (Chipman et al., 1998; Denison et al., 1998). The main constraint in RRDP compared to the general Bayesian CART is that the former is supported on canonical RDPs only—that is, each dyadic partition must be an even split, occurring at the middle of the range in one of the divisible dimensions. This additional restriction ensures the cardinality of $\mathcal{A}$ to be linear in $n$, thereby reducing the computational complexity required for inference to $O(n)$.

### 2.2 Recipes for Bayesian inference

Bayesian inference can proceed if we can derive the marginal posterior of $T$, because given $T$, we have a standard Bayesian wavelet regression model, for which classical inference strategies can apply. In this section, we present recipes for deriving and sampling from the posterior, and for evaluating posterior summaries such as the posterior mean of $\mathbf{f}$.

It turns out that when a Haar basis is adopted in the wavelet regression model, there exists a close-form generative expression for the marginal posterior of $T$, with which one can sample from the posterior directly through vanilla Monte Carlo (not Markov Chain Monte Carlo). This close-form posterior can be calculated through a recursive algorithm that are operationally similar to Mallat’s pyramid algorithm, achieving a linear computational complexity $O(n)$.

Before describing the results for Haar basis, we note that WARP can also be applied to other wavelet bases. In such cases, though the exact inference recipe as in the Haar basis is lost and there is no analytic expression for the posterior, an efficient sequential Monte Carlo (SMC) algorithm can be constructed for inference, using the analytic solution for the Haar basis as proposals. However, our experience in extensive numerical experiments suggests
that applying WARP to other bases generally results in no substantive performance gain at least in image analysis to justify the additional complexity and Monte Carlo variation involved in the SMC algorithm. As such we defer details on the SMC strategy for other bases to the Supplementary Materials, mainly for completeness.

### 2.2.1 Exact Bayesian inference under Haar basis

The Haar wavelet basis is unique in its very short support, which leads to the desirable property that under the vectorization induced by any RDP $T$, the $(j,k)$th wavelet coefficient are determined by only the locations inside the node $A_{j,k}$. We call this property of the Haar basis node-autonomy and say that inference under Haar basis is node-autonomous.

The node-autonomy of Haar wavelets has an important computational implication—inferece can be carried out in a self-similar fashion on each node, avoiding integration in the much larger space of RDPs. Consequently, exact inference can be completed in a computational complexity of the same scale as the total number of all nodes of all possible RDP trees, which is equal to $\prod_{i=1}^{m}(2n_i - 1) = O(2^m n)$.

Specifically, for all RDPs in which $A$ is a node and is divided in the $d$th direction, the corresponding Haar wavelet coefficient associated with the node $A$ is given by

$$w_d(A) = 1/\sqrt{|A|} \cdot \left( \sum_{x \in A_{l}^{(d)}} y(x) - \sum_{x \in A_{r}^{(d)}} y(x) \right)$$

where $A_{l}^{(d)}$ and $A_{r}^{(d)}$ represent the two children nodes if $A$ is divided in the $d$th dimension and $|A| = 2^{J-j}$ is the total number of locations in $A$. In contrast, wavelet coefficients from wavelet bases with longer support than Haar are not node-autonomous—not only does the coefficient associated with $A$ depends on the observations within $A$ but on those in other (often but not always adjacent) nodes in $T$ as well.

Next we lay out the general strategy for inference. We show through two theorems that generic inference recipes exist for two popular classes of Bayes wavelet regression models—(i) those that model each wavelet coefficient independently (Theorem 1); and (ii) those that induce a hidden Markov model (HMM) for incorporating dependency among the wavelet coefficients (Theorem 2).

**Theorem 1.** Suppose $T \sim \text{RRDP}(\lambda)$ and given the Haar DWT under $T$, one models the wavelet coefficients independently, i.e., $(w_{j,k}, z_{j,k}) \overset{\text{ind}}{\sim} p_{j,k}(w, z | \phi)$ for all $(j,k)$, where $\phi$ represents the hyperparameters of the Bayesian wavelet regression model. Then the marginal posterior of $T$ is still an RRDP. Specifically, $T | y \sim \text{RRDP}(\tilde{\lambda})$ where the posterior selection probability mapping $\tilde{\lambda}$ is given as

$$\tilde{\lambda}_d(A) = \lambda_d(A)M_d(A)\Phi(A_l^{(d)})\Phi(A_r^{(d)})/\Phi(A)$$

for any non-atomic $A \in \mathcal{A}$ where $M_d(A)$ is the marginal likelihood contribution from the wavelet coefficient on node $A$ if it is a node in $T$ and divided in dimension $d$, i.e., $M_d(A) =$
A depends on its values on $S$ where

Remark: $\Phi(\Omega)$ is the overall marginal likelihood. It is a function of the hyperparameters $\phi$, and can be used for specifying the hyperparameters $\phi$ in an empirical Bayes strategy using maximum marginal likelihood estimation (MMLE).

**Theorem 2.** Suppose $T \sim \text{RRDP}(\lambda)$ and given $T$ under a Haar DWT, one models the wavelet coefficients conditionally independently given a set of latent state variables $S = \{S_{j,k} : j = 0, 1, 2, \ldots, J, k = 0, 1, \ldots, 2^j - 1\}$

$$(w_{j,k}, z_{j,k}) \mid S_{j,k} = s \sim p_{j,k}(w, z \mid \phi) \quad \text{for all } (j, k)$$

where $S_{j,k} \in \{1, 2, \ldots, K\}$ is a latent state variable associate with $(j, k)$. Also, suppose the collection of all latent variables is modeled as a top-down Markov tree (MT) with transition kernel $\rho$, $S \sim \text{MT}(\rho)$, i.e.,

$$P(S_{j,k} = s' \mid S_{j-1,[k/2]} = s) = \rho_j(s, s')$$

where $\rho_j(\cdot, \cdot)$ is the transition kernel of the Markov model which is allowed to be different over $j$. Then the joint marginal posterior of $(T, S)$ can be specified fully as the following sequential generative process. Suppose $T^0, T^1, \ldots, T^j$ and the latent variables up to level $j - 1$ have been generated. (To begin, we have $j = 0$ and $T^0 = \{\Omega\}$.) Then the state variables in level $j$, are generated from the following posterior transition probabilities

$$P(S_{j,k} = s' \mid S_{j-1,[k/2]} = s, T^{(j)}, \mathbf{y}) = \rho_j(s, s') \sum_d \lambda_d(A) M_d^{(s')} (A) \Phi_{s'}(A_{i}^{(d)}) \Phi_{s'}(A_{v}^{(d)}) / \Phi_s(A)$$

where $A$ is the node $A_{j,k}$ in $T^j$. Given $S_{j,k} = s'$, suppose $j < J$, then $T^{j+1}$ is generated by drawing $D_{j,k}$ from a multinomial with probabilities $\tilde{\lambda}(A)$ such that

$$P(D_{j,k} = d \mid S_{j,k} = s', T^{(j)}, \mathbf{y}) = \frac{\lambda_d(A) M_d^{(s')} (A) \Phi_{s'}(A_{i}^{(d)}) \Phi_{s'}(A_{v}^{(d)})}{\sum_{d'} \lambda_{d'}(A) M_{d'}^{(s')} (A) \Phi_{s'}(A_{i}^{(d')}) \Phi_{s'}(A_{v}^{(d')})}$$

where $M_d^{(s)}(A)$ is the marginal likelihood contribution from the wavelet coefficient on node $A$ if it is a node in $T$, is divided in dimension $d$ in $T$, and its latent state is $s$. That is, $M_d^{(s)}(A) = \int p_{j,k}^{(s)}(w_{d}(A), z \mid \phi) dz$ and $\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_K) : A \to [0, \infty)^K$ is a vector-valued mapping defined recursively as follows

$$\Phi_s(A) = \begin{cases} \sum_{s'} \rho_j(s, s') \sum_{d \in \mathcal{D}(A)} \lambda_d(A) M_d^{(s')} (A) \Phi_{s'}(A_{i}^{(d)}) \Phi_{s'}(A_{v}^{(d)}) & \text{if } A \text{ is not atomic,} \\ 1 & \text{if } A \text{ is atomic.} \end{cases}$$

for all $s \in \{1, 2, \ldots, K\}$, where $j$ is the level of $A$. 

\[ \int p_{j,k}(w_{d}(A), z \mid \phi) dz \] and $\Phi : A \to [0, \infty)$ is a mapping defined recursively (i.e., its value on $A$ depends on its values on $A$’s children) as

$$\Phi(A) = \begin{cases} \sum_{d \in \mathcal{D}(A)} \lambda_d(A) M_d(A) \Phi(A_{i}^{(d)}) \Phi(A_{v}^{(d)}) & \text{if } A \text{ is not atomic,} \\ 1 & \text{if } A \text{ is atomic.} \end{cases}$$

Remark: $\Phi(\Omega)$ is the overall marginal likelihood. It is a function of the hyperparameters $\phi$, and can be used for specifying the hyperparameters $\phi$ in an empirical Bayes strategy using maximum marginal likelihood estimation (MMLE).
Theorem 1 and Theorem 2 provide recipes for posterior sampling on \((T, S)\). Given \((T, S)\), one can further sample \(z\) from the conditional posterior corresponding to the chosen wavelet regression model, and Bayesian inference can proceed in the usual manner. For example, function estimation can proceed through drawing \(B\) posterior samples 

\( (T^{(1)}, S^{(1)}, z^{(1)}), (T^{(2)}, S^{(2)}, z^{(2)}), \ldots, (T^{(B)}, S^{(B)}, z^{(B)}) \).

For the \(b\)th draw, we can compute the corresponding function \(f^{(b)}\) using the inverse DWT 

\[ f^{(b)} = \pi_T^{-1} \left( W^{-1} z^{(b)} \right), \]

where \(\pi_T^{-1}\) denotes the inverse permutation corresponding to an RDP \(T\). Based on the posterior samples of \(f\), we can estimate the posterior mean \(E(f|y)\) and construct pointwise credible bands. If a point estimate is of ultimate interest such as in image reconstruction, we can estimate the posterior function mean by a Rao-Blackwellized Monte Carlo on the posterior means of \(z\):

\[ E(f | y) \approx \frac{1}{B} \sum_{b=1}^{B} \pi_T^{-1} \left( W^{-1} E(z^{(b)}|T^{(b)}, y) \right). \]

For some Bayesian wavelet regression models, this posterior mean can actually be computed exactly, eliminating the need of posterior sampling whatsoever. We present an algorithm for a class of popular wavelet regression models in Section 2.2.3, after first reviewing these models in Section 2.2.2.

### 2.2.2 Examples of compatible Bayesian wavelet regression models

So far we have kept the description of the Bayesian wavelet regression model general, using generic notations such as \(p(w_{j,k}, z_{j,k} | \phi)\) and \(p(w_{j,k}, z_{j,k} | S_{j,k}, \phi)\) without spelling out the details. Next we describe some of the most popular Bayes wavelet regression models. They indeed take these general forms and therefore our framework is applicable to them.

A popular class of Bayesian models for achieving adaptive shrinkage of \(z\) utilize the so-called spike-and-slab prior, which introduces a latent binary random variable \(S_{j,k}\) for each \((j, k)\) such that

\[ z_{j,k} | S_{j,k} \overset{\text{ind}}{\sim} (1 - S_{j,k})\delta_0(z_{j,k}) + S_{j,k} \gamma(z_{j,k}|\tau_j, \sigma) \]

where \(\delta_0(\cdot)\) is a point mass at 0, and \(\gamma(\cdot|\tau_j, \sigma)\) is a fixed unimodal symmetric density that possibly depends on \(\sigma\) and another scale parameter \(\tau_j\). A common choice of \(\gamma(\cdot|\tau_j, \sigma)\) is the normal distribution with mean 0 and variance \(\tau_j\sigma^2\), denoted by \(\phi(\cdot|0, \sqrt{\tau_j}\sigma)\), while heavy-tailed priors including the Laplace and quasi-Cauchy distributions (Johnstone and Silverman, 2005) also enjoy desirable theoretical properties. Specifically, the function \(\gamma(x | \tau_j, \sigma)\) is

\[ \gamma(x | \tau_j, \sigma) = a \exp(-a|x/\sigma|)/(2\sigma) \]
for Laplacian priors where \( a = \sqrt{2/\tau_j} \), and
\[
\gamma(x \mid \tau_j, \sigma) = (2\pi)^{-1/2}\{1 - |x/\sigma| \cdot \Phi(|x/\sigma|)/\phi(x/\sigma)\}/\sigma
\]
for quasi-Cauchy priors with \( \tilde{\Phi}(x) = \int_0^\infty \phi(t \mid 0, 1)dt \).

Many authors (Brown et al., 2001; Chipman et al., 1997; Clyde and George, 2000; Morris and Carroll, 2006) adopt independent priors on the latent shrinkage state variable \( S_{j,k} \)
\[ S_{j,k} \overset{\text{iid}}{\sim} \text{Bern}(\rho_{j,k}). \]

One way to specify \( \rho = \{\rho_{j,k}, 0 \leq k < 2^j, 0 \leq j \leq J - 1\} \) that properly controls for multiplicity is \( \rho_{j,k} \propto 2^{-j} \). The specification of \( \tau = \{\tau_j, 0 \leq j \leq J - 1\} \) of course depends on the choice of \( \gamma(\cdot \mid \tau_j, \sigma) \). For instance, if one uses \( \tau_j = 2^{-\alpha j}\tau_0 \) for the normal and Laplace prior, this leads to the reduced parameter \( \tau = (\alpha, \tau_0) \). One can use \( \tau_j \equiv 1 \) for the quasi-Cauchy prior. Other authors (Crouse et al., 1998; Ma and Soriano, 2018) show that introducing Markov dependency into the latent shrinkage states can substantially improve inference by allowing effective borrowing of information across the location and scale.

Carrying out inference under WARP requires the conditional posterior of \( z_{j,k} \) given \( (T, S) \). For the above popular models, this posterior is given by
\[
z_{j,k} \mid S_{j,k}, \mathbf{y} \overset{\text{iid}}{\sim} (1 - S_{j,k})\delta_0(z_{j,k}) + S_{j,k}f_1(z_{j,k} \mid w_{j,k}, \tau_j, \sigma),
\]
where \( f_1(z_{j,k} \mid w_{j,k}, \tau_j, \sigma) \propto \phi(w_{j,k} \mid z_{j,k}, \sigma) \cdot \gamma(z_{j,k} \mid \tau_j, \sigma) \). The function \( f_1(z_{j,k} \mid w_{j,k}, \tau_j, \sigma) \) is analytically available if \( \gamma(\cdot \mid \tau_j, \sigma) \) is the density of normal, Laplace, or quasi-Cauchy distributions. For the normal prior where \( \gamma(\cdot \mid \tau_j, \sigma) = \phi(\cdot \mid 0, \sqrt{\tau_j}\sigma) \), \( f_1(\cdot \mid w_{j,k}, \tau_j, \sigma) \) is the density of \( N(w_{j,k}/(1 + \tau_j^{-1}), \sigma^2/(1 + \tau_j^{-1})) \). For Laplacian and quasi-Cauchy priors, analytical forms of \( f_1(\cdot \mid \tau_j, \sigma) \) are available in (Johnstone and Silverman, 2005, Sec. 2.3). As it is often the mean corresponding to \( f_1 \) that is needed for posterior estimation, we here give the closed forms of the means by integrating out \( z_{j,k} \) with respect to its posterior distribution. Let the corresponding mean function be \( \mu_1(w_{j,k}, \tau_j, \sigma) \), we then have
\[
\mu_1(w_{j,k}, \tau_j, \sigma) = \begin{cases} 
  w_{j,k}/(1 + \tau_j^{-1}) & \text{normal priors} \\
  w_{j,k} - \sigma & \text{Laplace priors} \\
  w_{j,k} \left\{1 - \exp\left(-\frac{w_{j,k}^2}{2\sigma^2}\right)\right\}^{-1} - 2(\frac{w_{j,k}}{\sigma})^{-1} & \text{quasi-Cauchy priors}.
\end{cases}
\]

### 2.2.3 WARP with local block shrinkage

Traditional wavelet analysis is done by fixing the maximum depth of the wavelet tree at \( J \). That is, one partitions the index space all the way down to the finest level of “atomic” blocks. In most practical problems, once the blocks are small enough, the function value within the block becomes essentially constant with respect to the noise level, and so further division within such homogeneous blocks will be wasteful and will reduce statistical efficiency. For example, in Figure 1 the partition in the upper left block (Level 3) along with its descendants
are not necessary. Thus it is often desirable to incorporate adaptivity in the depth of wavelet tree and allow it to be terminated earlier than reaching level $J$. In practice the optimal maximum depth varies across $\Omega$. For example, some parts of an image may contain many interesting details, while the rest does not—e.g., an image of a painting hung on a gray wall. A high resolution will be needed to capture the details in the painting, but would be unnecessary and introduce additional variability in the estimation for the wall.

This consideration is closely related to the idea of adaptive block shrinkage (Cai, 1999) in the frequentist wavelet regression analysis. Once there is little evidence for any interesting structure within a subset of the index space, then the function value within that subset can be shrunk to a constant. That is, the wavelet tree is “pruned” there. Next we show that such pruning can be achieved in a hierarchical modeling manner, and the resulting Bayesian wavelet regression model is again compatible with our WARP framework.

To achieve such pruning, we introduce another set of latent variables $R = \{R_{j,k} : j = 0, 1, \ldots, J - 1, k = 0, 1, \ldots, 2^j - 1\}$, where $R_{j,k} = 1$ indicates that the tree is pruned at node $(j, k)$. Next we describe a generative prior on $R$ that will blend well with the WARP framework. To start, let $R_{0,0} \sim \text{Bern}(\eta_{0,0})$ and for all $j \geq 1$, let

$$
R_{j,k} \mid R_{j-1,\lceil k/2 \rceil} \sim \begin{cases} 
\text{Bern}(\eta_{j,k}) & \text{if } R_{j-1,\lceil k/2 \rceil} = 0 \\
\text{Bern}(1) & \text{if } R_{j-1,\lceil k/2 \rceil} = 1
\end{cases}.
$$

That is, if a node’s parent has been pruned, then its children are also pruned by construction. From now on, we shall refer to this prior model on $R$ as an optional pruning (OP) model (Ma, 2013), which is specified by a set of pruning probabilities $\eta_{j,k} \in [0, 1]$. We write $R \sim \text{OP}(\eta)$.

Given $R$, we can modify our prior on $S$ to reflect the effect of pruning. For example, instead of an independent prior on $S$, we can now generate them as follows

$$
S_{j,k} \mid R \sim \begin{cases} 
\text{Bern}(\rho_{j,k}) & \text{if } R_{j,k} = 0 \\
\text{Bern}(0) & \text{if } R_{j,k} = 1
\end{cases}.
$$

That is, if the node has not been pruned, then we generate $S_{j,k}$ from the independent Bernoulli as in the standard spike-and-slab setup, but if the node has been pruned, then by construction, we must have $S_{j,k} = 0$ due to pruning.

It is often reasonable to specify the prior shrinkage and pruning probabilities as functions of the level in the RDP. That is, $\rho_{j,k} = \rho_j$ and $\eta_{j,k} = \eta_j$ for all $k$. In the node-specific notation, $\rho(A) = \rho_j$ and $\eta(A) = \eta_j$ for all $j$th node $A \in A$. In this case, one can show that this joint model on $(S, R)$ is equivalent to a Markov tree model with three states defined in terms of the combinations of $(S_{j,k}, R_{j,k}) = (1, 0), (0,0),$ or $(0,1)$, and with the corresponding transition matrix for $S_{j,k}$ given by

$$
\rho_j = \begin{bmatrix}
\rho_j (1 - \eta_j) & (1 - \rho_j)(1 - \eta_j) & \eta_j \\
\rho_j (1 - \eta_j) & (1 - \rho_j)(1 - \eta_j) & \eta_j \\
0 & 0 & 1
\end{bmatrix}.
$$

This allows us to derive the posterior from Theorem 2, and carry out inference accordingly.

13
For the Haar basis, the posterior mean $E(f \mid y)$ can be evaluated analytically through recursive message passing without any Monte Carlo sampling for Bayesian wavelet regression models that adopt the spike-and-slab setup along with optional pruning of the wavelet tree, which contains the models without optional pruning as special cases with zero pruning probabilities. We describe the strategy next and will use it to compute $E(f \mid y)$ in our numerical examples.

For each $A \in \mathcal{A}$, let $c(A)$ be the scale (father wavelet) coefficient on $A$ if $A \in \mathcal{T}$, and let $\varphi(A) = E(c(A)1_{\{A \in \mathcal{T}\}} \mid y)$. Note that $E(f \mid y)$ is given by $\varphi(A)$ for all atomic $A$. To compute the mapping $\varphi$, we introduce two auxiliary mappings $\psi_0(A) = P(A \in \mathcal{T}, R(A) = 0 \mid y)$ and $\varphi_0(A) = E(c(A)1_{\{A \in \mathcal{T}, R(A) = 0\}} \mid y)$. Let $A^{(d)}$ denote the parent of $A$ in $\mathcal{T}$ if $A$ is a child node after dividing its parent in the $d$th dimension, and let $\mathcal{P}(A) \subset \{1, 2, \ldots, m\}$ be the collection of dimensions of $A$ that do not have full support $[0, 2^h - 1]$, i.e., those that have been partitioned at least once in previous levels. Theorem 3 gives a recursive message passing algorithm for computing the tri-variate mapping $(\varphi_0, \psi_0, \varphi): \mathcal{A} \rightarrow \mathbb{R}^3$.

**Theorem 3.** To initiate the recursion, for $A = \Omega$, we let $\psi_0(A) = 1 - \tilde{\eta}(A), \varphi_0(A) = (1 - \tilde{\eta}(A))|A|/\sqrt{n}$ and $\varphi(A) = |A|/\sqrt{n}$. Suppose we have evaluated these mappings up to level $j - 1$, for level $j = 1, \ldots, J$, we have

\[
\begin{align*}
\psi_0(A) &= \sum_{d \in \mathcal{P}(A)} \psi_0(A^{(d)})\hat{\lambda}_d(A^{(d)})(1 - \tilde{\eta}(A)); \\
\varphi_0(A) &= \sum_{d \in \mathcal{P}(A)} \frac{\hat{\lambda}_d(A^{(d)})}{\sqrt{2}} \left[ \varphi_0(A^{(d)}) - \tilde{\rho}_d(A^{(d)})\mu_1(w_d(A^{(d)}))\psi_0(A^{(d)}) \cdot (-1)^{1(A \text{ is the left child of } A^{(d)})} \right] \\
&\quad \cdot (1 - \tilde{\eta}(A)); \\
\varphi(A) &= \varphi_0(A)/(1 - \tilde{\eta}(A)) + \frac{1}{\sqrt{2}} \sum_{d \in \mathcal{P}(A)} \{\varphi(A^{(d)}) - \varphi_0(A^{(d)})\}\lambda_d(A^{(d)}).
\end{align*}
\]

Remark: Note that this recursion is top-down (from low to high resolutions), whereas that for computing $\Phi$ is bottom-up (from high to low resolutions). The two-directional recursion shares the spirit of the forward-backward algorithm for HMMs.

Once we have computed the mapping $(\varphi_0, \psi_0, \varphi): \mathcal{A} \rightarrow \mathbb{R}^3$, the posterior mean $E(f \mid y)$ is then given by $\varphi$ applied on the atomic nodes. Note that this theorem applies to the special case with no pruning as well.

### 3 Experiments

In this section, we conduct extensive experiments to evaluate the performance of our proposed framework in the context of image reconstruction. In particular, we examine its estimation accuracy, computational scalability, and ability to adaptively concentrate the energy. We compare WARP to a number of state-of-the-art wavelet and non-wavelet methods available in the literature. For illustration, we apply WARP to the independent spike-and-slab Bayesian
wavelet regression model with Haar basis and optioning pruning to denoise both 2D and 3D images. Because the results for 2D and 3D images are similar, we report the results on 2D images in this section and defer most results on 3D images, except those for evaluating computational scalability, to Supplementary Materials.

Our prior specification is as follows: $\rho(A) = \min(1, 2^{-\beta_j} C)$ for $A$ in the $j$th resolution, $\tau_j = 2^{-\alpha_j} \tau_0$, and $\eta(A) = \eta_0$ for all $A$; we set $\sigma^2$ to an estimate based on the finest scale wavelet coefficients (Donoho and Johnstone, 1995); all other parameters in $\phi = (\alpha, \beta, \sigma^2, \tau, C, \eta_0)$ are estimated by maximizing the marginal likelihood (available in a closed form as $\Phi(\Omega)$ from our recursive message passing algorithm) at a set of grid points. Supplementary Materials contain sensitivity analyses showing that WARP is generally robust to the values of its hyperparameters, therefore we recommend a grid search on a small set rather than a full optimization as the default method. Gaussian noise with standard deviation $\sigma$ is added to the true images and we apply all methods to the noisy observations for image reconstruction. For WARP, we use the posterior mean as the reconstructed image, which is analytically attainable through Theorem 3.

3.1 Performance comparison in image reconstruction

We illustrate the work of our method in the application of image reconstruction. We use 100 test images randomly chosen from the ImageNet dataset (Deng et al., 2009) in addition to the famous Lena image. ImageNet is originally used for large-scale visual recognition in the community of computer vision, and we here use its Fall 2011 release (consisting of 14,197,122 urls) to evaluate selected methods in reconstructing images of various structures.

We compare our method with eight existing wavelet and non-wavelet approaches with available software: 1-dimensional Haar denoising operated on vectorized observation (Johnstone and Silverman, 2005) or 1D-Haar, translation-invariant 2D Haar estimation (Willett and Nowak, 2004) or TI-2D-Haar, shape-adaptive Haar wavelets (Fryzlewicz and Timmermans, 2016) or SHAH, adaptive weights smoothing (Polzehl and Spokoiny, 2000) or AWS, Bayesian smoothing method using the Chinese restaurant process (Li and Ghosal, 2014) or CRP, coarse-to-fine wedgelet (Castro et al., 2004) or Wedgelet, nonparametric Bayesian dictionary learning proposed by Zhou et al. (2012) or BPFA, and the conventional running median method or RM. We apply the cycle spinning technique to remove visual artifacts in image reconstruction (Coifman and Donoho, 1995; Li and Ghosal, 2015) for the methods of WARP, 1D-Haar, SHAH, AWS, CRP, Wedgelet and RM, by averaging 121 local shifts (a step size up to 5 pixels in each direction). 2D-Haar is translation invariant and BPFA includes cycle spinning based on patches, and thus no additional cycle spinning is needed for these two methods. For each method, we calculate the mean squared error (MSE) to measure its accuracy, and time each method based on one replication ran on MacBook Pro with 2.7 GHz Intel core i7 CPU and 16GB RAM.

Table 1 reports the average MSEs of all methods where $\sigma$ varies from 0.1 to 0.7. We can first see that the proposed hierarchical adaptive partition improved the basic wavelet regression significantly (compare the row of 1D-Haar vs. WARP) for all scenarios. For all scenarios, WARP is always among the top two approaches (the two smallest MSEs are in
Table 1: Average MSEs ($\times 10^{-2}$) of various methods based on 100 replications (Lena) and 100 randomly selected images from ImageNet. Running time is recorded in seconds when $\sigma = 0.3$ (Lena).

| Method   | Lena 0.1 | Lena 0.3 | Lena 0.5 | Lena 0.7 | ImageNet 0.1 | ImageNet 0.3 | ImageNet 0.5 | ImageNet 0.7 | Time |
|----------|----------|----------|----------|----------|--------------|--------------|--------------|--------------|------|
| WARP     | 0.07     | 0.18     | 0.28     | 0.38     | 0.02         | 0.06         | 0.12         | 0.17         | 7.2  |
| 1D-Haar  | 0.17     | 0.78     | 1.34     | 1.60     | 0.08         | 0.40         | 0.60         | 0.75         | < 1  |
| TI-2D-Haar | 0.08   | 0.23     | 0.36     | 0.49     | 0.03         | 0.12         | 0.22         | 0.31         | < 1  |
| SHAH     | 0.08     | 0.20     | 0.32     | 0.46     | 0.03         | 0.12         | 0.22         | 0.32         | 76.9 |
| AWS      | 0.11     | 0.21     | 0.31     | 0.41     | 0.04         | 0.11         | 0.21         | 0.31         | 7.9  |
| CRP      | 0.08     | 0.21     | 0.32     | 0.42     | 0.05         | 0.17         | 0.25         | 0.33         | 10.7 |
| Wedgelet | 0.08     | 0.22     | 0.35     | 0.48     | 0.03         | 0.10         | 0.19         | 0.32         | 8.7  |
| BPFA     | 0.06     | 0.19     | 0.35     | 0.53     | 0.02         | 0.13         | 0.29         | 0.50         | $2.1 \times 10^3$ |
| RM       | 0.16     | 0.89     | 2.32     | 4.47     | 0.17         | 0.88         | 2.31         | 4.43         | < 1  |

bold with possible ties). In fact, WARP leads to the smallest MSEs in all but one scenario (Lena with light noise $\sigma = 0.1$), with the performance lead over other methods widening as the noise level increases. The sensitivity analysis in the Supplementary Materials indicates that the method of WARP is robust to hyperparameters and choices of $\gamma$.

WARP is computationally efficient benefiting from the conjugacy of random recursive partition and closed form expression in Theorem 3. The last column of Table 1 clearly shows that WARP is the fastest approach among SHAH, AWS, CRP, Wedgelet, and BPFA, while uniformly more accurate than 1D-Haar, TI-2D-Haar, and RM.

Visual evaluation is also useful since it reveals how well a method reconstructs important features present in images. Figure 2 plots the denoised Lena image under two noise levels for the methods of Haar, WARP, 2D-Haar and the best other approach based on MSEs. We can observe that 1D Haar destroys the geometric structure in the image due to vectorization (column Haar, especially Row 2). Importantly, WARP is capable of recovering sharp edges (see for example Lena’s nose and mouth) and smooths out the noise well. The ability to recover local features such as the nose and mouth is generally not reflected in the global risk measures such as the MSE. In contrast, 2D-Haar smooths out the noise well but smooths out the sharp features in the observation while the best among other methods for Lena (BPFA) seems to under-smooth the noise at the lower signal-to-noise ratio (Row 2).

### 3.2 Assessment of energy concentration

To further understand the reason for the effectiveness of WARP, we use the 100 ImageNet images to illustrate quantitatively how much improvement in energy concentration WARP achieves through adaptively identify good permutations. To this end, we define a metric to quantify energy concentration using the number of wavelet coefficients needed to exceed a proportion of the sum of squares of the signal.
For each ImageNet image, we draw a sample from the posterior distribution of partition trees produced by WARP, and compute the number of coefficients required to attain a proportion of the total sum of squares corresponding to the resulting permutation on a noisy observation at $\sigma = 0.1$. We compute the numbers of wavelet coefficients required to attain a proportion of the total sum of squares for deterministic partitions via 2D Haar and for 1D Haar DWT applied to the vectorized true image by column. Figure 3 presents the numbers of wavelet coefficients required as a function of the proportion of sum of squares for three representative images.

Focusing on the range of proportion of sum of squares from 0.85 to 0.95, we can see that compared to a deterministic partition via either 2D or 1D discrete Haar wavelet transformation, the adaptive partition in WARP requires substantially fewer numbers of wavelet coefficients to attain the same proportion (see the red and blue line in each plot of Figure 3 corresponding to the right $y$ axis). For better visualization, we calculate the percentage of coefficient savings (black line in Figure 3) defined by $100\%$ less the ratio of the blue and red curves. We can see that WARP substantially improves the energy concentration for all three test images and energy levels. In Figure 3, the largest coefficient saving of WARP is $(80\%, 70\%, 70\%)$ compared to 2D DWT, and this saving becomes $(97\%, 99\%, 90\%)$ when compared to 1D DWT. In fact, we observe that WARP requires substantially fewer coefficients to attain the same energy level compared to 2D and 1D DWT in a wide range of test images in the database, and how much WARP improves the energy concentration depends on the image structure.

### 3.3 Scalability

Next we verify the linear complexity of the WARP framework using both 2D and 3D images. Figure 4 demonstrates the scalability of WARP for both 2D and 3D images by plotting the time consumed by the estimation step versus the number of locations. We can see that the
Figure 3: Comparison of energy concentration for three methods—WARP, 1D Haar, and 2D Haar—on ImageNet images. Column (a) plots the true image, Column (b) compares WARP versus 1D DWT, and Column (c) compares WARP versus 2D DWT. In Columns (b) and (c), the red and blue lines correspond to the right $y$ axis, plotting the number of coefficients to attain a specific energy level ($x$ axis) by deterministic DWT and WARP, respectively. The black curve corresponds to the left $y$ axis and is 100% less the ratio of the blue and red curves, indicating the percentage reduction in the number of wavelet coefficients to achieve the same sum of squares by WARP.
empirical running time approximately follows a linear function of the number of pixels (2D) or voxels (3D). It is worth mentioning that WARP takes only about 2 minutes for a large image of $4096 \times 4096$ that contains 17 million pixels, and 5.3 seconds for an image of 1024 by 1024. Given the hyperparameters, WARP scales linearly in the number of locations for fixed $m \geq 2$.

It is worth noting that while many state-of-the-art methods designed for 2D images such as Wedgelet, TI-2D-Haar, and BPFA require substantial modifications for a new dimensional setting, such as 3D images, the proposed WARP framework is directly applicable to $m$-dimensional data without modification.

4 Discussion

We have introduced the WARP framework that uses random recursive partitioning to induce a prior on the permutations of the index space, thereby achieving efficient inference on multi-dimensional functions by converting it into a Bayesian model choice problem involving one-dimensional competitive generative models. While our approach is Bayesian, one may consider other methods such as frequentist adaptive partitioning and shrinkage methods that incorporate the same idea. We do find satisfying the fully principled probabilistic inferential recipes based on recursive message passing that arise under our approach.

The proposed framework WARP is applicable to a wider range of Bayes wavelet regression models, including those low heterogeneous noise variances. If the error $\epsilon$ in Model (1) has general covariance matrix $\Sigma_\epsilon$, it often still makes sense to assume that the covariance of the error $u$ in the wavelet domain, i.e. $W\Sigma_\epsilon W'$, is diagonal, due to the so-called whitening property of wavelet transforms discussed in Johnstone and Silverman (1997). In this case, let $\sigma_j^2 = \text{Var}(u_{j,k})$ for each $j$, then one may estimate $\sigma_j^2$ using a robust estimator of the scale based on $\{w_{j,k}, 0 \leq k \leq 2^j - 1\}$ given a tree, for example, using the median absolute deviation.
of \{w_{j,k}, 0 \leq k \leq 2^j - 1\} rescaled by 0.6745. Alternatively, one can adopt a hyperprior on location-based unknown variance \(\sigma^2_j \sim IG(\nu + 1, \nu \sigma^2_0)\) which is an inverse gamma prior with shape \(\nu + 1\) and scale \(\nu \sigma^2_0\) (thus the prior mean is \(\sigma^2_0\)). The hyperparameters \((\nu, \sigma^2_0)\) are either specified by users or estimated using data, for instance, we may estimate \(\sigma^2_j\) by the median estimate based on the finest scale wavelet coefficients (Donoho and Johnstone, 1995).

While we introduce the WARP framework in the context of image denoising, the adaptive wavelet transform is applicable to other domains involving multi-dimensional function processing. In particular, one area of current investigation is data compression—the posterior distribution on the permutations can be utilized to compress multi-dimensional signals to one or more one-dimensional signals, while preserving most of the information in the data. For example, in Figure 3 shows that even just using a random sample from the posterior of the partitions can help reduce the number of wavelet coefficients needed to retain information in the data by over 50% in 2D images, in comparison to traditional 2D wavelets. We are currently studying strategies to using a representative permutation from the posterior, such as the posterior mode partition to achieve data compression.

Acknowledgment

Meng Li’s research is partly supported by ORAU Ralph E. Powe Junior Faculty Enhancement Award. Li Ma’s research is partly supported by NSF grants DMS-1612889 and DMS-1749789. We thank Daniel Bourgeois for his help in porting our C++ code to R.

Supplementary Materials

Supplementary materials contain Proposition 1 and its proof; proofs of Theorems (Theorem 1, Theorem 2, and Theorem 3) and Lemma 2; a sequential Monte Carlo algorithm for applying WARP to non-Haar wavelets along with a numerical example; a sensitivity analysis for the proposed framework; and comparison of WARP and selected methods using experiments of 3D image reconstruction.

References

Abramovich, F., Sapatinas, T., and Silverman, B. W. (1998). Wavelet thresholding via a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(4):725–749.

Aho, A. V. and Sloane, N. J. (1973). Some doubly exponential sequences. *Fibonacci Quart*, 11(4):429–437.

Ali, S. T., Antoine, J.-P., and Gazeau, J.-P. (2014). *Coherent states, wavelets and their generalizations*. Springer, 2nd edition.
Brown, P. J., Fearn, T., and Vannucci, M. (2001). Bayesian Wavelet Regression on Curves With Application to a Spectroscopic Calibration Problem. *Journal of the American Statistical Association*, 96(454):398–408.

Cai, T. T. (1999). Adaptive wavelet estimation: A block thresholding and oracle inequality approach. *Annals of Statistics*, 27(3):898–924.

Castro, R., Willett, R., and Nowak, R. (2004). Coarse-to-fine manifold learning. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. (ICASSP’04)*.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART Model Search. *Journal of the American Statistical Association*, 93(443):935–948.

Chipman, H. A., Kolaczyk, E. D., and McCulloch, R. E. (1997). Adaptive Bayesian Wavelet Shrinkage. *Journal of the American Statistical Association*, 92(440):1413–1421.

Clyde, M. and George, E. I. (2000). Flexible empirical Bayes estimation for wavelets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):681–698.

Coifman, R. and Donoho, D. (1995). Translation-invariant de-noising. In Antoniadis, A. and Oppenheim, G., editors, *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 125–150. Springer New York.

Coupé, P., Yger, P., Prima, S., Hellier, P., Kervrann, C., and Barillot, C. (2008). An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images. *IEEE Transactions on Medical Imaging*, 27(4):425–441.

Crouse, M. S., Nowak, R. D., and Baraniuk, R. G. (1998). Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902.

Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Denison, D. G. T., Mallick, B. K., and Smith, A. F. M. (1998). A Bayesian CART Algorithm. *Biometrika*, 85(2):363–377.

Donoho, D. L. (1999). Wedgelets: Nearly minimax estimation of edges. *The Annals of Statistics*, 27(3):859–897.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adapation by wavelet shrinkage. *Biometrika*, 81(3):425–455.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224.
Fryzlewicz, P. and Timmermans, C. (2016). Shah: Shape-adaptive haar wavelets for image processing. *Journal of Computational and Graphical Statistics*, 25(3):879–898.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian Data Analysis*. CRC press Boca Raton, FL.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial. *Statist. Sci.*, 14(4):382–417.

Jacques, L., Duval, L., Chaux, C., and Peyré, G. (2011). A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity. *Signal Processing*, 91(12):2699–2730.

Jain, A. (1989). *Fundamentals of Digital Image Processing*. Prentice-Hall Information and System Sciences Series. Prentice Hall.

Johnstone, I. M. and Silverman, B. W. (1997). Wavelet Threshold Estimators for Data with Correlated Noise. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(2):319–351.

Johnstone, I. M. and Silverman, B. W. (2005). Empirical Bayes selection of wavelet thresholds. *Annals of Statistics*, 33(4):1700–1752.

Lakshminarayanan, B., Roy, D., and Whye Teh, Y. (2013). Top-down particle filtering for Bayesian decision trees. *Proceedings of The 30th International Conference on Machine Learning*, 28:280–288.

Le Pennec, E. and Mallat, S. (2005). Sparse geometric image representations with bandelets. *IEEE transactions on image processing*, 14(4):423–438.

Li, M. and Ghosal, S. (2014). Bayesian multiscale smoothing of Gaussian noised images. *Bayesian Analysis*, 9(3):733–758.

Li, M. and Ghosal, S. (2015). Fast translation invariant multiscale image denoising. *IEEE Transactions on Image Processing*, 24(12):4876–4887.

Lin, M., Chen, R., Liu, J. S., et al. (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science*, 28(1):69–94.

Liu, J. S. (2004). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer New York, New York, NY.

Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044.

Ma, L. (2013). Adaptive testing of conditional association through recursive mixture modeling. *Journal of the American Statistical Association*, 108(504):1493–1505.
Ma, L. and Soriano, J. (2018). Efficient functional ANOVA through wavelet-domain Markov groves. *Journal of the American Statistical Association*.

Mallat, S. (2008). *A Wavelet Tour of Signal Processing: the sparse way*. Academic press, third edition.

Morris, J. S. and Carroll, R. J. (2006). Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):179–199.

Mukherjee, P. and Qiu, P. (2011). 3-d image denoising by local smoothing and nonparametric regression. *Technometrics*, 53(2):196–208.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Peyré, G. (2011). A review of adaptive image representations. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):896–911.

Polzehl, J. and Spokoiny, V. G. (2000). Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(2):335–354.

Raftery, A. E. (1995). Bayesian model selection in social research. *Sociological Methodology*, 25:111–163.

Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268.

Willet, R. and Nowak, R. (2004). Fast multiresolution photon-limited image reconstruction. In *IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2004*, pages 1192–1195.

Wong, W. H. and Ma, L. (2010). Optional Pólya tree and Bayesian inference. *The Annals of Statistics*, 38(3):1433–1459.

Zhou, M., Chen, H., Paisley, J., Ren, L., Li, L., Xing, Z., Dunson, D., Sapiro, G., and Carin, L. (2012). Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, 21(1):130–144.
Supplementary Materials to “Partition mixture of Bayesian wavelet regression for multidimensional data”

Supplementary materials contain (A) Proposition 1 and its proof, (B) proofs of all theorems and Lemma 2, (C) a sequential Monte Carlo algorithm for applying WARP to non-Haar wavelets along with a numerical example, (D) a sensitivity analysis for the proposed framework, and (E) comparison of WARP and selected methods using experiments of 3D image reconstruction.

A Cardinality of the space of RDPs

Proposition 1. The log cardinality of the tree space induced by RDPs is $O(n)$ when $m = 2$.

Proof of Proposition 1. Let $c(a, b)$ be the cardinality of the tree space induced by RDPs for an $2^a$ by $2^b$ image. We can obtain the following recursive formula

$$c(a, b) = \begin{cases} c^2(a - 1, b) + c^2(a, b - 1), & \text{if } a, b \geq 1 \\ 1 & \text{if } a = 0 \text{ or } b = 0. \end{cases}$$

We assert that there exist two constants $(k_1, k_2)$ such that $k_2 \geq k_1 > 1$ and

$$c(a, b) \in \left[\frac{1}{2}k_1^{2a+b}, \frac{1}{2}k_2^{2a+b}\right],$$

for any $a \geq 1$ and $b \geq 1$.

First consider $a = 1$ and $b \geq 1$. We have $c(1, b) = c^2(1, b - 1) + 1$ when $b \geq 1$ and $c(1, 0) = 1$ when $b = 0$. The quantity $c(1, b)$ is actually the number of “strongly” binary trees of height $\leq b$, which possesses an analytical form

$$c(1, b) = \lfloor k^{2b} \rfloor,$$

according to Aho and Sloane (1973), where

$$k = \exp \left\{ \sum_{j=0}^{\infty} 2^{-j-1} \log(1 + c^{-2}(1, j)) \right\} \approx 1.503.$$

Letting $k_1 = \sqrt{k}$ and $k_2 = k$ and noting $k^{2b} \geq 2$ for all $b \geq 1$, we obtain that

$$\frac{1}{2}k_1^{2a+b} = \frac{1}{2}k^{2b} \leq k^{2b} - 1 \leq \lfloor k^{2b} \rfloor \leq k^{2b} \leq \frac{1}{2}k_2^{2a+b},$$

for all $b \geq 1$. Therefore, the assertion holds for all $a = 1$ and $b \geq 1$. Since $c(a, b) = c(b, a)$, the assertion also holds for all $a \geq 1$ and $b = 1$.

For any $a \geq 1$ and $b \geq 1$, it is easy to verify that if the assertion holds for $(a, b - 1)$ and $(a - 1, b)$, then it holds for $(a, b)$. We then complete the proof by induction. \qed
B Proofs of Theorems and Lemmas

Proof of Theorem 1. Because Theorem 2 can be considered a special case with a single latent state, its proof follows immediately from the latter theorem, which we prove below.

Proof of Theorem 2. First we verify that the mapping $\Phi_s(A)$ is the marginal likelihood contributed from data with locations in $A$, given that $A \in \mathcal{T}$ and that the latent state variable associated with parent of $A$ in $\mathcal{T}$ is $s$. We show this by induction. First note that if $A$ is atomic, then

$$\Phi_s(A) = P(y(A) \mid A \in \mathcal{T}, S(A_p) = s) = 1$$

by design as there are no wavelet coefficients associated with atomic nodes. Now, suppose we have shown that $\Phi_s(A) = P(y(A) \mid A \in \mathcal{T}, S(A_p) = s)$ for all $A$ with level higher than $j$. Then if $A$ is of level $j$, it follows that

$$P(y(A) \mid A \in \mathcal{T}, S(A_p) = s)$$

$$= \sum_{s'} \sum_d P(y(A) \mid A \in \mathcal{T}, S(A) = s', S(A_p) = s, D(A) = d) P(S(A) = s' \mid A \in \mathcal{T}, S(A_p) = s)$$

$$\times P(D(A) = d \mid A \in \mathcal{T}, S(A_p) = s)$$

$$= \sum_{s'} \rho_j(s, s') \sum_{d \in D(A)} \lambda_d M^{(s')}_{d} (A) \Psi_{s'}(A^d_l) \Psi_{s'}(A^d_r),$$

which leads to the definition of $\Phi_s(A)$ in Theorem 2.

Next let us derive the joint marginal posterior of $(\mathcal{T}, S)$. Note that

$$P(S_{j,k} = s' \mid S_{j-1,\lfloor k/2 \rfloor} = s, \mathcal{T}^{(j)}, y) = \frac{P(S_{j,k} = s', S_{j-1,\lfloor k/2 \rfloor} = s, y(A) \mid \mathcal{T}^{(j)})}{P(S_{j-1,\lfloor k/2 \rfloor} = s, y(A) \mid \mathcal{T}^{(j)})}.$$

Now we have

$$P(S_{j,k} = s', D_{j,k} = d, y(A) \mid \mathcal{T}^{(j)}, S_{j-1,\lfloor k/2 \rfloor} = s) = \rho_j(s, s') \lambda_d(A) M^{(s')}_{d} (A) \Phi_{s'}(A^d_l) \Phi_{s'}(A^d_r),$$

which leads to

$$P(S_{j,k} = s', y(A) \mid \mathcal{T}^{(j)}, S_{j-1,\lfloor k/2 \rfloor} = s) = \rho_j(s, s') \sum_d \lambda_d(A) M^{(s')}_{d} (A) \Phi_{s'}(A^d_l) \Phi_{s'}(A^d_r)$$

and furthermore,

$$P(S_{j,k} = s' \mid S_{j-1,\lfloor k/2 \rfloor} = s, \mathcal{T}^{(j)}, y) = \frac{\rho_j(s, s') \sum_d \lambda_d(A) M^{(s')}_{d} (A) \Phi_{s'}(A^d_l) \Phi_{s'}(A^d_r)}{\sum_{s''} \rho_j(s, s'') \sum_d \lambda_d(A) M^{(s'')}_{d} (A) \Phi_{s''}(A^d_l) \Phi_{s''}(A^d_r)},$$

where the denominator is just $\Phi_s(A)$. 

S-2
Finally,

\[
P(D_{j,k} = d \mid S_{j,k} = s', T^{(j)}, y) = \frac{P(D_{j,k} = d, y(A) \mid S_{j,k} = s', T^{(j)})}{P(y(A) \mid S_{j,k} = s', T^{(j)})} = \frac{\lambda_d(A)M_{d}^{(s')} (A) \Phi_{s'} (A_1^{(d)}) \Phi_{s'} (A_i^{(d)})}{\sum_{d'} \lambda_{d'}(A)M_{d'}^{(s')} (A) \Phi_{s'} (A_1^{(d)}) \Phi_{s'} (A_i^{(d)})}.
\]

This completes the proof.

Proof of Theorem 3. We first obtain the recursive recipe for computing the maps \((\psi_0, \varphi_0)\) following Theorem 1:

\[
\psi_0(A) = \sum_{d \in P(A)} P(\bar{A}^{(d)} \in T, R(\bar{A}^{(d)}) = 0 \mid y) \bar{\lambda}_d(\bar{A}^{(d)})(1 - \bar{\eta}(A))
\]
\[
= \sum_{d \in P(A)} \psi_0(\bar{A}^{(d)}) \bar{\lambda}_d(\bar{A}^{(d)})(1 - \bar{\eta}(A)),
\]

and

\[
\varphi_0(A) = E\left(c(A)1_{\{A \in T, R(A) = 0\}} \mid y\right) = \sum_{d \in P(A)} E\left(c(A)1_{\{\bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d, R(A) = 0\}} \mid y\right)
\]
\[
= \sum_{d \in P(A)} E\left(c(A) \mid \bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d, R(A) = 0 \right) \cdot P\left(\bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d, R(A) = 0 \mid y\right)
\]
\[
= \sum_{d \in P(A)} \frac{\bar{\lambda}_d(\bar{A}^{(d)})}{\sqrt{2}} \left[ \varphi_0(\bar{A}^{(d)}) - \bar{\rho}_d(\bar{A}^{(d)}) \mu_1 (w_d(\bar{A}^{(d)})) \psi_0(\bar{A}^{(d)}) \cdot (-1)^{1(A \text{ is the left child of } \bar{A}^{(d)})} \right] \cdot (1 - \bar{\eta}(A)).
\] (2)

We next derive the recursive formula for \(\varphi(A)\). Let \(\varphi_1(A) = E(c(A)1_{\{A \in T, R(A) = 1\}} \mid y)\), then we have \(\varphi(A) = E(c(A)1_{\{A \in T\}} \mid y) = \varphi_0(A) + \varphi_1(A)\). Note that

\[
\varphi(A) = \sum_{d \in P(A)} E\left(c(A)1_{\{\bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d\}} \mid y\right),
\]

(3)

and for each \(d \in P(A)\), we have

\[
E\left(c(A)1_{\{\bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d\}} \mid y\right) = \sum_{r=0,1} E\left(c(A)1_{\{\bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d, R(\bar{A}^{(d)}) = r\}} \mid y\right)
\]
\[
= \sum_{r=0,1} E\left(c(A) \mid \bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d, R(\bar{A}^{(d)}) = r, y\right) \times P(\bar{A}^{(d)} \in T, D(\bar{A}^{(d)}) = d, R(\bar{A}^{(d)}) = r \mid y).
\] (4)
For the second term in (4), we have
\[
P(\bar{A}^{(d)} \in \mathcal{T}, D(\bar{A}^{(d)}) = d, R(\bar{A}^{(d)}) = r \mid y) = P(D(\bar{A}^{(d)}) = d \mid \bar{A}^{(d)} \in \mathcal{T}, R(\bar{A}^{(d)}) = r, y) \cdot P(\bar{A}^{(d)} \in \mathcal{T}, R(\bar{A}^{(d)}) = r \mid y)
\]

\[= \tilde{\lambda}_d(\bar{A}^{(d)})^{1-r} \lambda_d(\bar{A}^{(d)})^r \psi_r(\bar{A}^{(d)}) \]

For the first term in (4), it is easy to check that
\[
E(c(A) \mid \bar{A}^{(d)} \in \mathcal{T}, D(\bar{A}^{(d)}) = d, R(\bar{A}^{(d)}) = r, y) = \begin{cases} \\
\frac{1}{\sqrt{2}} \left[ \frac{\varphi_0(\bar{A}^{(d)}) - \tilde{\rho}_d(\bar{A}^{(d)}) \mu_1(w_d(\bar{A}^{(d)}))}{\tau_j^{-1}} \cdot (-1)^{1(A \text{ is the left child of } \bar{A}^{(d)})} \right] & \text{if } r = 0 \\
\frac{1}{\sqrt{2}} \varphi_1(\bar{A}^{(d)}) / \psi_1(\bar{A}^{(d)}) & \text{if } r = 1,
\end{cases}
\]

where we use the independence between \(c(A)\) and \(D(A)\) given \(A \in \mathcal{T}\). Plugging the two terms into (4), we obtain that
\[
E(c(A) \mathbbm{1}_{\{\bar{A}^{(d)} \in \mathcal{T}, D(\bar{A}^{(d)}) = d\}} \mid y) = \frac{1}{\sqrt{2}} \varphi_0(\bar{A}^{(d)}) - \tilde{\rho}_d(\bar{A}^{(d)}) w_d(\bar{A}^{(d)}) / (1 + \tau_j^{-1}) \cdot (-1)^{1(A \text{ is the left child of } \bar{A}^{(d)})} \cdot \psi_0(\bar{A}^{(d)}) \tilde{\lambda}_d(\bar{A}^{(d)})
\]

\[+ \frac{1}{\sqrt{2}} \varphi_1(\bar{A}^{(d)}) \lambda_d(\bar{A}^{(d)}). \tag{5}\]

Combining the result in (3) and (5), and comparing it with \(\varphi_0(A)\) in (2), we obtain that
\[
\varphi(A) = \frac{\varphi_0(A)}{1 - \bar{\eta}(A)} + \frac{1}{\sqrt{2}} \sum_{d \in \mathcal{P}(A)} \varphi_1(\bar{A}^{(d)}) \lambda_d(\bar{A}^{(d)}),
\]

which concludes the proof by plugging in \(\varphi_1(\cdot) = \varphi(\cdot) - \varphi_0(\cdot)\). \qed

Proof of Lemma 2. The conclusion obviously holds for the last level of the move as only \(l\) wavelet coefficients are affected and \(l \geq 1\). Suppose the conclusion holds at \(j\)’th level, then for the \((j' - 1)\)th level, the number of affected size-2 blocks is at most \(l\), thus the number of affected wavelet coefficients is at most \(l + l - 1 = 2l - 1\). The proof concludes by induction. \qed

C Sequential Monte Carlo for other wavelet bases

We present a sequential Monte Carlo (Liu, 2004) algorithm for estimating the posterior mean when applying WARP to Bayesian wavelet regression with bases other than Haar. In particular, we shall use the posterior under the Haar basis to construct the proposal. In what follows, we refer to the Bayes wavelet regression model with the Haar basis as \(\mathbb{B}_0\) and one with a user-specified general wavelet basis as \(\mathbb{B}\).

Suppose that we have drawn \(I\) samples from the marginal posterior on the RDP \(\mathcal{T}\) under \(\mathbb{B}_0\), denoted as \(\mathcal{T}_i\) for \(i = 1, \ldots, I\). Let \(f_{\mathbb{B}}^{(i)} = \pi_{\mathcal{T}_i}^{-1}(W_{\mathbb{B}}^{-1} E(\mathcal{T}^{(i)} \mid \mathcal{T}^{(i)}, y, \mathbb{B}))\) be the
estimated mean vector under a Bayesian wavelet regression model given the partition $T_i$. Now let $\Psi(\Omega | T_i, B_0)$ and $\Psi(\Omega | T_i, B)$ be the marginal likelihoods given the partition $T_i$ under the operators $B_0$ and $B$, respectively. Then an importance sampling (Gelman et al., 2014, Ch.10) estimate of $E(f | y, B)$ is given by

$$E(f | y, B) \approx \frac{w_i f_B^{(i)}}{\sum_{i=1}^{T} w_i},$$

where $w_i = p(T_i) \Psi(\Omega | T_i, B_0)/p(T_i | y, B_0)$ is the unnormalized importance weight; here $p(T_i)$ is the prior of $T_i$ and the posterior of $T_i$ under the Haar basis $p(T_i | y, B_0)$ is used as the proposal.

However, importance sampling is not useful if the importance weight $w_i$ is degenerate and concentrates all the probabilities on one or a small number of draws (also referred to as particles). This occurs very commonly due to the massive size of the space of RDPs, and is indeed what we observed in our experiment running on an image of typical size, say 128 by 128 or 512 by 512, which is expected as the deviation between the target and the proposal distribution accumulates over the entire RDP and the number of nodes is as large as $n - 1$.

Sequential Monte Carlo (SMC) that updates the weight sequentially with a possible resampling step is a powerful tool for such complex dynamic systems (Lin et al., 2013; Liu and Chen, 1998). Next we shall adapt the SMC algorithm into the framework of WARP, show that the implementation of SMC has a nearly linear complexity, and demonstrate its performance using numerical experiments.

**SMC algorithm**

Consider a general wavelet basis with $2l$ support where the high pass filters are $(h_0, \ldots, h_{2l-1})$ and low pass filters are $(g_0, \ldots, g_{2l-1})$. For example, if one uses Daubechies D4 wavelet transform, the high-pass filter is

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}},$$

and low-pass filter is

$$g_0 = h_3, g_1 = -h_1, g_2 = h_1, g_3 = -h_0.$$

Suppose at the beginning of the current stage of particle propagation, the $i$th particle is a partially generally RDP tree $T_i^{(j,k)}$ such that $j < J - 1$ and all nodes $A_{j,k'}$ for $k' \leq k$ have been expanded, i.e., have children, whereas for all $k' > k$, $A_{j,k'}$ are leafs in the RDP. Thus the leafs of the RDP, denoted as $\partial T_i^{(j,k)}$, is

$$\partial T_i^{(j,k)} = \{A_{j,k'} : k' > k\} \cup \{A_{j+1,k'} : k' \leq 2k - 1\}.$$

In the current stage of particle propagation, we shall expand the node $A_{j,k+1}$ if $k < 2^j - 1$; otherwise we expand on $A_{j+1,0}$ if $j < J$. The particle propagation terminates as when $j = J - 1$ and $k = 2^{J-1} - 1$, in which case $T_i^{(J-1,2^{J-1}-1)} = T_i$ denotes an entire particle path as it reaches the very last scale of a tree.
Let \((w_{j',k'}(\mathcal{T}_{i}^{(j,k)}), c_{j',k'}(\mathcal{T}_{i}^{(j,k)}))\) be the \((j', k')\)th pair of mother and father wavelet coefficients, which are calculated iteratively as follows:

\[
c_{j',k'} = \begin{cases} 
\sum_{x \in A_{j',k'}} g(x)/\sqrt{|A_{j',k'}|} & \text{if } A_{j',k'} \in \partial \mathcal{T}_{i}^{j,k} \\
\sum_{t=0}^{2l-1} h_{t} c_{j'+1,2k'+t} & \text{if } A_{j',k'} \notin \partial \mathcal{T}_{i}^{j,k}, 
\end{cases}
\]

and

\[
w_{j',k'} = \begin{cases} 
0 & \text{if } A_{j',k'} \in \partial \mathcal{T}_{i}^{j,k} \\
\sum_{t=0}^{2l-1} g_{t} c_{j'+1,2k'+t} & \text{if } A_{j',k'} \notin \partial \mathcal{T}_{i}^{j,k}. 
\end{cases}
\tag{6}
\]

We here use periodic padding for the boundary condition, namely,

\[
c_{j',k'} = \begin{cases} 
c_{j',k'-2^{r}} & \text{if } j' \leq j \text{ and } k' \geq 2^{r} \\
c_{j',k'-2k} & \text{if } j' = j+1 \text{ and } k' \geq 2k.
\end{cases}
\]

Now let \(\Psi(\Omega|\mathcal{T}_{i}^{(j,k)}, \mathbb{B})\) denote the overall marginal likelihood contributed from the wavelet coefficients in the partial tree \(\mathcal{T}_{i}^{(j,k)}\) under \(\mathbb{B}\), and \(w_{i}^{(j,k)} = p(\mathcal{T}_{i}^{(j,k)}) \Psi(\Omega|\mathcal{T}_{i}^{(j,k)}, \mathbb{B})/p(\mathcal{T}_{i}^{(j,k)}|\mathbf{y}, \mathbb{B}_0)\) be the current weight for the \(i\)th particle. We have

\[
w_{i}^{(j,k+1)} = w_{i}^{(j,k)} \cdot \frac{p(\mathcal{T}_{i}^{(j,k+1)}|\mathcal{T}_{i}^{(j,k)}, \mathbb{B})/p(\mathcal{T}_{i}^{(j,k+1)}|\mathbf{y}, \mathbb{B}_0)}{p(\mathcal{T}_{i}^{(j,k)}|\mathcal{T}_{i}^{(j,k)}, \mathbb{B})/p(\mathcal{T}_{i}^{(j,k)}|\mathbf{y}, \mathbb{B}_0)} \cdot \frac{\Psi(\Omega|\mathcal{T}_{i}^{(j,k+1)}, \mathbb{B})}{\Psi(\Omega|\mathcal{T}_{i}^{(j,k)}, \mathbb{B})},
\]

where \(d\) is the partition dimension of \(A_{j,k}\) at the move of the particle. Algorithm 1 presents a complete implementation of SMC for general wavelet basis.

For a level \(j\) node \(A\), the likelihood corresponding to node \(A\) is \(M_{d}(A; \mathcal{T}, \mathbb{B}) = \rho(A) M_{d}^{(1)}(A) + (1-\rho(A)) M_{d}^{(0)}(A)\), where \(M_{d}^{(0)}(A) = \phi(w_{d}(A)|0, \sigma)\), \(M_{d}^{(1)}(A) = g(w_{d}(A)|\tau_{j}, \sigma)\), and \(w_{d}(A)\) is obtained by Eq. (6) under tree \(\mathcal{T}\). Therefore, the likelihood ratio in Algorithm 1 is

\[
\frac{\Psi(\Omega|\mathcal{T}_{i}^{(j,k+1)}, \mathbb{B})}{\Psi(\Omega|\mathcal{T}_{i}^{(j,k)}, \mathbb{B})} = \prod_{j'=0}^{j} \prod_{A \in \mathcal{A}_{j'}^{(j,k)}} \frac{M_{d}(A; \mathcal{T}_{i}^{(j,k+1)}, \mathbb{B})}{M_{d}(A; \mathcal{T}_{i}^{(j,k)}, \mathbb{B})},
\]

where \(\mathcal{A}_{j'}^{(j,k)}\) is the collection of nodes at level \(j'\) where the wavelet coefficients require updates at the move \((\mathcal{T}_{i}^{(j,k-1)} \rightarrow \mathcal{T}_{i}^{(j,k)})\).

**Complexity of SMC**

The complexity of SMC is mostly determined by the step of updating wavelet coefficients at each move which boils down to the cardinality of \(\bigcup_{j'=0}^{j} \mathcal{A}_{j'}^{(j,k)}\). A wavelet basis other than the
Algorithm 1: SMC

Data: Prior partition probability: $\lambda$; Posterior partition probability based on $B_0$: $\tilde{\lambda}$; Number of particles: $I$; Effective sample size threshold: $\text{ESS}$.

for $i = 1, 2, \ldots, I$; /* Initialization */
    do
        Initialize $T_i^{(0,0)} = \Omega$, $w_i^{(0,0)} = 1$, and $\tilde{w}_i^{(0,0)} = 1/I$.
    end

for $j = 1, 2, \ldots, J - 1$ do
    for $k = 0, 1, \ldots, 2^j - 1$ do
        for $i = 1 : I$ do
            Particle Propagation: $T_i^{(j,k+1)} | T_i^{(j,k)} \sim \text{Multinomial}(\tilde{\lambda}(A_{j,k}))$
            Update wavelet coefficients at nodes $\bigcup_{j'=0}^j A_{j,k}^{j+1}$
            Weight update:
            
            \[
            w_i^{(j,k+1)} = w_i^{(j,k)} \cdot \frac{\lambda(A_{j,k}) \cdot \Psi(\Omega | T_i^{(j,k+1)}, B)}{\Psi(\Omega | T_i^{(j,k)}, B)}.
            \]
        end
    Compute normalization $W_i = \sum_{i=1}^I w_i^{(j,k+1)}$
    Normalize weights: $\tilde{w}_i^{(j,k+1)} = w_i^{(j,k+1)}/W_i$
    if $\{\sum_{i=1}^I (\tilde{w}_i^{(j,k+1)})^2\}^{-1} < \text{ESS}$; /* Resampling */
        then
            Resample $T_i^{(j,k+1)}$ from $\sum_{i=1}^I \tilde{w}_i^{(j,k+1)} \delta_i$.
        Reset $w_i^{(j,k+1)} = W_i/I$.
    end
end

Result: $T_i$ and the associated weight $w_i$; estimated marginal probability $W_i/I$.

Haar basis is not node-autonomous and thus the calculation of all quantities corresponding to each node often depends on other nodes. Hence each move of a particle affects more than a change at the new node. Specifically, let us consider a wavelet basis whose support length is $2l$ (such as the Daubechies wavelet (Daubechies, 1992)) where $l \geq 2$. A particle moving from $T^{(j,k-1)}$ to $T^{(j,k)}$ brings two new nodes to the particle, i.e., $A_{j+1,2k}$ and $A_{j+1,2k+1}$. As a consequence, the wavelet coefficients associated with nodes $(A_{j,k-l+1}, \ldots, A_{j,k})$ at level $j$ all need to be updated, and that in turn affects wavelet coefficients at all previous levels $0, 1, \ldots, j - 1$. The following proposition shows that the cardinality of $A_{j,k}^j$ is at most $(2l - 1)j^l$. It follows that the total calculation to update the likelihood before a particle terminates is $O((2l - 1) \sum_{j=1}^{L-1} j \cdot 2^j) = O((2l - 1)L \cdot 2^L) = O((2l - 1)n \log n)$, that is, with a nearly linear complexity.
Proposition 2 (Computational complexity of SMC). If $D-2l$ is used in $\mathcal{B}$, then each move of a particle affects at most $2l - 1$ wavelet coefficients at each parent level.

Remark: This proposition confirms that the induced changes in all previous wavelet coefficients are sparse thus efficient update is possible.

Experiment

We next illustrate the proposed SMC algorithm with D4 wavelet basis, and compare its performance to the exact WARP on Haar basis. We use 121 cycle shifts for both methods. We draw 10 particles per cycle spinning for SMC, which leads to $I = 10 \times 121 = 1210$ particles per run in total. We specify the effective sample size as $0.1I$. Combined with the technique of cycle spinning, our proposed SMC algorithm inherently incorporates the structure of islanding (Lakshminarayanan et al., 2013) through averaging the results of multiple independent particle filters (or islands) rather than drawing a single but larger filter.

Figure 5 plots the ratio of MSEs obtained by a WARPed D4 via SMC and WARPed Haar at the phantom (Jain, 1989) and Lena test images. It shows that the two types of wavelet lead to close performance, but D4 wavelet tends to outperform the Haar basis when the noise level increases. Even when the noise level is light, the performance of WARPed D4 wavelet may be viewed as satisfactory since the WARPed Haar has been shown to be constantly among the best approaches in a wide range of selected methods.

Figure 5: The true image of Phantom and Ratios of MSEs (solid blue line) between D4 and Haar wavelets using the phantom and Lena data at various $\sigma$. The dashed black lines are (the average of ratios $\pm 1.96 \times$ standard error) based on 10 replications.

D Sensitivity analysis

In this section, we conduct a sensitivity analysis for the proposed WARP framework at various choices of hyperparameters.
We first implement the method of “WARP-full” which chooses $\phi$ by a full optimization of the marginal likelihood using Lena in Table 1 and $(f_1, f_2)$ in Table 5. Recall that the row of WARP selects $\phi$ at a limited set of grid points. Table 2 and Table 3 show that the MSEs of WARP-full are almost identical to the row of WARP. This observation is consistent across many scenarios we have tested. Therefore, the method of WARP seems robust in terms of hyperparameters, and we shall recommend a maximization over a small set of grid points as default. In addition, we investigate the performances of WARP at various choices of $\gamma$ in $B_0$ including Laplace and quasi-Cauchy priors. We find out these $B_0$ lead to almost exactly the same MSEs as normal priors (results not shown here).

We further investigate the sensitivity of WARP by considering the following ways to select hyperparameters $\tau$ and $\eta$:

- $\tau$: “function” (we use $\tau_j = 2^{-\alpha_j} \tau_0$ as in Section 3); “mix” (we use separate $\tau_j$ only for the last two levels and a constant for other levels, therefore we have three free parameters for $\tau$); “full” (we use separate $\tau_j$’s for all levels $j$)

- $\eta$: “constant” (we use $\eta(A) = \eta_0$ for all $A$ as in Section 3); “mix” (we use $\eta_j$ for the last two levels and a constant for other levels, therefore we have three free parameters for $\eta$); “full” (we use separate $\eta_j$’s for all levels $j$).

Table 4 show that the MSEs only exhibit minimal differences across various combinations of tuning approaches. This confirms the previous findings that the proposed framework is not sensitive to hyperparameters.

E 3D images

Unlike WARP which is directly applicable to $m$-dimensional data for $m > 2$, other methods in Table 1 such as Wedgelet, TI-2D-Haar, and BPFA may require substantial modifications for a new dimensional setting. SHAH is conceptually applicable for 3D data, but the existing software takes hours to days in the tuning step for 3D images of intermediate size while its performance in 2D settings is not among top two. Therefore, we compare WARP with AWS, CRP, and RM along with a collection of other approaches, including a 3D image denoising method via local smoothing and nonparametric regression (LSNR) proposed by Mukherjee and Qiu (2011), anisotropic diffusion (AD) method (Perona and Malik, 1990), total variation minimization (TV) method (Rudin et al., 1992) and optimized non-local means (ONLM) method (Coupé et al., 2008). The TV method is modified by Mukherjee and Qiu (2011) by minimizing a 3D-version of the TV criterion. We adopt simulation settings in Mukherjee and Qiu (2011), which uses two artificial 3D images with the following true intensity functions:

$$f_1(x, y, z) = -(x - \frac{1}{2})^2 - (y - \frac{1}{2})^2 - (z - \frac{1}{2})^2 + 1 \mathbb{I}\{(x, y, z) \in R_1 \cup R_2\},$$

where $R_1 = \{(x, y, z) : |x - \frac{1}{2}| \leq \frac{1}{4}, |y - \frac{1}{2}| \leq \frac{1}{4}, |z - \frac{1}{2}| \leq \frac{1}{4}\}$ and $R_2 = \{(x, y, z) : (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2 \leq 0.15^2, |z - \frac{1}{2}| \leq 0.35\}$; $f_2(x, y, z) = \frac{1}{4} \sin(2\pi(x + y + z) + 1) + \frac{1}{4} + 1 \mathbb{I}\{(x, y, z) \in S_1 \cup S_2\},$
Table 2: Average MSEs ($\times 10^{-2}$) of WARP-full and WARP based on 100 replications using Lena test image as in Table 1.

|         | Lena  |
|---------|-------|
| $\sigma$ | 0.1   | 0.3   | 0.5 | 0.7 |
| WARP-full | 0.07  | 0.18  | 0.28 | 0.38 |
| WARP     | 0.07  | 0.18  | 0.28 | 0.38 |

Table 3: Average MSEs ($\times 10^{-2}$) of WARP-full and WARP based on 100 replications under the setting of Table 5.

| Method   | $n = 64$ | $n = 128$ |
|----------|----------|-----------|
|          | $f = f_1$ | $f = f_2$ | $f = f_1$ | $f = f_2$ |
|          | $\sigma = 0.1$ | 0.2 | $\sigma = 0.1$ | 0.2 | $\sigma = 0.1$ | 0.2 | $\sigma = 0.1$ | 0.2 |
| WARP-full | 0.02 | 0.04 | 0.04 | 0.12 | 0.01 | 0.02 | 0.02 | 0.05 |
| WARP     | 0.02 | 0.04 | 0.04 | 0.11 | 0.01 | 0.02 | 0.02 | 0.05 |

Table 4: Sensitivity analysis of WARP when hyperparameters are selected differently. The average MSEs ($\times 10^{-2}$) are reported based on 5 replications.

| $\tau$    | $\eta$ | Lena  | Phantom |
|-----------|--------|-------|---------|
|          | $\sigma$ | 0.1 | 0.3 | 0.5 | 0.7 | 0.1 | 0.3 | 0.5 | 0.7 |
| function constant | 0.07 | 0.18 | 0.28 | 0.37 | 0.03 | 0.27 | 0.57 | 0.89 |
| function mix | 0.07 | 0.18 | 0.28 | 0.37 | 0.03 | 0.27 | 0.58 | 0.88 |
| function full | 0.07 | 0.19 | 0.28 | 0.37 | 0.03 | 0.27 | 0.57 | 0.87 |
| mix constant | 0.07 | 0.20 | 0.30 | 0.40 | 0.03 | 0.26 | 0.57 | 0.94 |
| mix mix | 0.07 | 0.20 | 0.30 | 0.40 | 0.03 | 0.26 | 0.57 | 0.91 |
| mix full | 0.07 | 0.20 | 0.30 | 0.40 | 0.03 | 0.27 | 0.57 | 0.91 |
| full constant | 0.07 | 0.18 | 0.28 | 0.38 | 0.03 | 0.27 | 0.58 | 0.86 |
| full mix | 0.07 | 0.18 | 0.29 | 0.38 | 0.03 | 0.27 | 0.56 | 0.87 |
| full full | 0.07 | 0.18 | 0.28 | 0.37 | 0.03 | 0.27 | 0.57 | 0.88 |

where $S_1 = \{(x, y, z): (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2 \leq \frac{1}{4}(z - \frac{1}{2})^2, 0.2 \leq z \leq 0.5\}$ and $S_2 = \{(x, y, z): 0.2^2 \leq (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2 + (z - \frac{1}{2})^2 \leq 0.4^2, z < 0.45\}$; here 1 l is the indicator function.

Table 5 shows the comparison of various methods using MSE. It is worth mentioning that the numerical records for the other five methods to estimate $f_1$ and $f_2$ are from Mukherjee and Qiu (2011) as the code is not immediately available and the running time for some method such as LSNR can take hours to days (including the tuning step). WARP is uniformly the best approach among all the selected methods at least under the simulation setting.

In view of that CRP and AWS produced state-of-the-art performance (Li and Ghosal, 2014; Polzehl and Spokoiny, 2000) and the code is publicly available, we further compare
WARP with these two methods using a wide range of noise level in Table 6. While WARP slightly underperforms CRP at light noise (such as $\sigma = 0.1$), it becomes substantially superior to CRP at higher noise levels for both test images. The running time in the caption of Table 6 shows that WARP is much faster than CRP in both tuning and estimation steps, which is crucial for large 3D images. WARP outperforms AWS for almost all cases; exceptions are a tie at light noise ($f = f_1; \sigma = 0.1$) and a slightly larger MSE ($f = f_2; \sigma = 0.3$).

Table 5: 3D denoising for two images $f_1$, $f_2$ in terms of MSE ($\times 10^{-2}$). WARP uses $5 \times 5 \times 5$ local shifts and are based on 100 replications. The mean of 100 MSEs is reported. The maximum standard error for each column is 0.00 for each row.

| Method | $n = 64$ | $n = 128$ |
|--------|----------|-----------|
|        | $f = f_1$ | $f = f_2$ | $f = f_1$ | $f = f_2$ |
|        | $\sigma = 0.1$ | $\sigma = 0.2$ | $\sigma = 0.1$ | $\sigma = 0.2$ |
| WARP   | 0.02 | 0.04 | 0.04 | 0.11 | 0.01 | 0.02 | 0.02 | 0.05 |
| LSNR   | 0.03 | 0.08 | 0.06 | 0.13 | **0.01** | **0.03** | **0.02** | 0.06 |
| TV     | 0.03 | 0.09 | 0.06 | 0.15 | **0.01** | 0.04 | 0.03 | 0.06 |
| AD     | 0.06 | 0.35 | 0.07 | 0.38 | 0.03 | 0.20 | 0.04 | 0.22 |
| ONLM   | 0.03 | 0.12 | 0.06 | 0.14 | **0.01** | 0.06 | 0.03 | 0.06 |
| RM     | 0.22 | 0.33 | 0.11 | 0.26 | 0.08 | 0.19 | 0.06 | 0.14 |

Table 6: Comparison of (WARP, CRP, AWS) for $128 \times 128 \times 128$ images. We report MSE($\times 10^{-3}$) which is averaged over 100 replications. The smallest MSE in each scenario is in bold. The running times for tuning and estimation are (173s, 13.3s) for WARP and (1.4 $\times 10^4$s, 86s) for CRP, recorded at $\sigma = 0.3$.

| $\sigma$ | $f = f_1$ | $f = f_2$ |
|-----------|-----------|-----------|
|           | WARP | CRP | AWS | WARP | CRP | AWS |
| 0.1       | 0.07 | **0.06** | 0.07 | 0.18 | **0.11** | 0.19 |
| 0.3       | 0.27 | 1.03 | 0.35 | 1.03 | 1.60 | **0.93** |
| 0.5       | 0.51 | 4.04 | 1.22 | **2.13** | 5.65 | 2.14 |
| 0.7       | **0.78** | 7.41 | 2.63 | **3.14** | 15.30 | 3.43 |