Information and Inference: A Journal of the IMA (2016) **5**, 105–133
doi:10.1093/imaiai/iaw007
Advance Access publication on 19 April 2016

Deep Haar scattering networks

XIUYUAN CHENG[†]
Applied Mathematics Program, Yale University, New Haven, CT, USA
[†]Corresponding author: xiuyuan.cheng@yale.edu

XU CHEN
Electrical Engineering Department, Princeton University, Princeton, NJ, USA
Email: xuchen@princeton.edu

AND

STÉPHANE MALLAT
Computer Science Department, École Normale Supérieure, Paris, France
Email: stephane.mallat@ens.fr

[Received on 3 February 2015; revised on 22 December 2015; accepted on 1 March 2016]

An orthogonal Haar scattering transform is a deep network computed with a hierarchy of additions, subtractions and absolute values over pairs of coefficients. Unsupervised learning optimizes Haar pairs to obtain sparse representations of training data with an algorithm of polynomial complexity. For signals defined on a graph, a Haar scattering is computed by cascading orthogonal Haar wavelet transforms on the graph, with Haar wavelets having connected supports. It defines a representation which is invariant to local displacements of signal values on the graph. When the graph connectivity is unknown, unsupervised Haar learning can provide a consistent estimation of connected wavelet supports. Classification results are given on image data bases, defined on regular grids or graphs, with a connectivity which may be known or unknown.

Keywords: deep learning; neural network; scattering transform; Haar wavelet; classification; images; graphs.

1. Introduction

Deep neural networks provide scalable learning architectures for high-dimensional data, with impressive results on many different types of data and signals [3]. The networks alternate linear operators, whose coefficients are optimized with training samples, with point-wise nonlinearities. To obtain good classification results, strong constraints are imposed on the network architecture on the support of these linear operators [27]. Despite their efficiency, there are many open issues to understand the properties of these architectures [31].

Over the past few years, much of the efforts in deep learning has been devoted to supervised learning. Supervised deep neural networks have achieved great successes in the classifications of images, video, speech, audio and texts. Convolutional neural networks [27] usually provide the most efficient architectures among supervised deep neural networks. They implement a cascade of linear filtering based on convolutions, followed by pointwise nonlinearities and subsampling or max pooling operators. Deep convolutional networks get state-of-the-art results for almost all classification and detection problems in computer vision, with performances comparable to humans in some tasks [18,44].

Despite the phenomenal success of supervised deep neural networks, a fundamental challenge to deep learning is the lack of sufficient labeled training data in many practical situations. The availability of huge amounts of unlabeled examples motivates unsupervised learning. Unsupervised learning is about discovering regularities, features or structures from unlabeled data. Many unsupervised methods are designed to maximize entropy-related objectives or to generate distributed and sparse representations of the input signals. Unsupervised layer-wise pre-training is useful for training deep networks such as DBNs [19] and Stacked Auto-encoders [14,21]. It can help prevent overfitting when the data set is small [4]. Unsupervised deep learning is also used to estimate probability distributions and generate new samples from these distributions [5,41].

This paper studies unsupervised deep learning by introducing a simple deep Haar scattering architecture, which only computes the sum of pairs of coefficients, and the absolute value of their difference. Inspired by scattering networks [6,30], the architecture preserves some important properties of deep networks, while reducing the computational complexity and simplifying their mathematical analysis. Most deep neural networks are fighting against the curse of dimensionality by reducing the variance of the input data with contractive nonlinearities [3,38]. The danger of such contractions is to nearly collapse together vectors which belong to different classes. We will show that unsupervised Haar scattering can optimize an average discriminability by computing sparse features. Sparse unsupervised learning, which is usually NP hard, is reduced to a pair-matching problem for Haar scattering. It can thus be computed with a polynomial complexity algorithm. Under appropriate assumptions, we prove that pairing problems avoid the curse of dimensionality.

In social, sensor or transportation networks, high-dimensional data vectors are supported on a graph [42]. In most cases, propagation phenomena require to define translation invariant representations for classification. We will show that an appropriate configuration of an orthogonal Haar scattering defines such a translation invariant representation on a graph. When the connectivity of the graph is unknown, building invariant representations requires to estimate the graph connectivity. Such information can be inferred from unlabeled data by analyzing the joint variability of signals defined on the unknown graph. Despite its simplicity, a Haar scattering gives good classification results compared to other deep learning networks, to classify scrambled images (so that the pixel grid information is removed), or other datasets defined on unknown graphs.

Haar wavelets on graphs have already been studied for machine learning. A tree representation of graph data defines a multi-resolution analysis associated to a Haar wavelet basis [10,16]. For unknown graphs, a hierarchical tree can be computed with iterative algorithms based on diffusion maps and spectral clustering [1,10]. There are many possible constructions of wavelets on graphs [11,43], providing sparse representation of graph signals [40]. These techniques have mostly been applied to image and network data reconstructions, as opposed to classification problems.

Section 2 introduces the learning pipeline of a Haar scattering transform. It optimizes an orthogonal Haar scattering representation from unsupervised data, to which is applied a supervised classification including feature selection. Section 3 studies locally invariant representations of signals defined on a graph, with a Haar scattering transform. Section 4 gives numerical results on several classification problems. All computations can be reproduced with a software available at *www.di.ens.fr/data/scattering/haar*.

2. Orthogonal Haar scattering

2.1 *Haar scattering transform*

We progressively introduce an orthogonal Haar scattering by specializing a general deep neural network. The input layer is a positive d-dimensional signal $x \in (\mathbb{R}^+)^d$. The positivity assumption simplifies the model, and positive data are widely encountered in image data and other applications. When the data takes negative values, the algorithm is adapted in [8], by not applying an absolute value to additions, which play the role of low-pass filters in scattering transforms [6,30].

We denote by $S_j x$ the network layer at the depth j, and $S_0 x = x$. A deep neural network computes $S_{j+1}x$ by applying a linear operator H_j to $S_j x$, followed by a nonlinear point-wise operator. Particular deep network architectures impose that H_j preserves distances, up to a constant normalization factor λ [34]:

$$\|H_j y - H_j y'\| = \lambda \|y - y'\|.$$

The network then applies a pointwise contraction ρ to each value of the output vector $H_j S_j x$. If $|\rho(a) - \rho(b)| \leqslant |a - b|$ for any $(a,b) \in \mathbb{R}^2$, then the network is contractive. Examples include rectifications $\rho(a) = \max(0,a)$ and sigmoids. In a Haar scattering network, we use an absolute value $\rho(a) = |a|$. It preserves amplitude and gives a permutation invariance which is studied. For any vector $y = (y(n))_n$, the pointwise absolute value is written $|y| = (|y(n)|)_n$. The next network layer is thus:

$$S_{j+1}x = |H_j S_j x|. \tag{2.1}$$

This transform is iterated up to a maximum depth $J \leqslant \log_2(d)$ to compute the network output $S_J x$.

We shall further impose that each layer $S_j x$ has the same dimension as x, and hence that H_j is an orthogonal operator in $\mathbb{R}^d$, up to the scaling factor λ. Geometrically, $S_{j+1}x$ is thus obtained by rotating $S_j x$ with H_j, and by contracting each of its coordinate with the absolute value. The geometry of this contraction is defined by the choice of the operator H_j, which adjusts the one-dimensional directions along which the contraction is performed.

An orthogonal Haar scattering is implemented with an orthogonal Haar filter H_j at each layer. The vector $H_j y$ regroups the coefficients of $y \in \mathbb{R}^d$ into $d/2$ pairs and computes their sums and differences. The rotation H_j is thus factorized into $d/2$ rotations by $\pi/4$ in $\mathbb{R}^2$, and multiplications by $2^{1/2}$. The transformation of each coordinate pair $(\alpha,\beta) \in \mathbb{R}^2$ is

$$(\alpha,\beta) \longrightarrow (\alpha+\beta,\ \alpha-\beta).$$

The operator $|H_j|$ applies an absolute value to each output coordinate, which has no effect on $\alpha+\beta$ since $\alpha \geqslant 0$ and $\beta \geqslant 0$, while it removes the sign of their difference:

$$(\alpha,\beta) \longrightarrow (\alpha+\beta, |\alpha-\beta|). \tag{2.2}$$

Observe that this nonlinear operator defines a permutation invariant representation of (α,β). Indeed, the output values are not modified by a permutation of α and β, and the two values of α, β are recovered without order, by

$$\max(\alpha,\beta) = \tfrac{1}{2}(\alpha+\beta+|\alpha-\beta|) \quad \text{and} \quad \min(\alpha,\beta) = \tfrac{1}{2}(\alpha+\beta-|\alpha-\beta|). \tag{2.3}$$

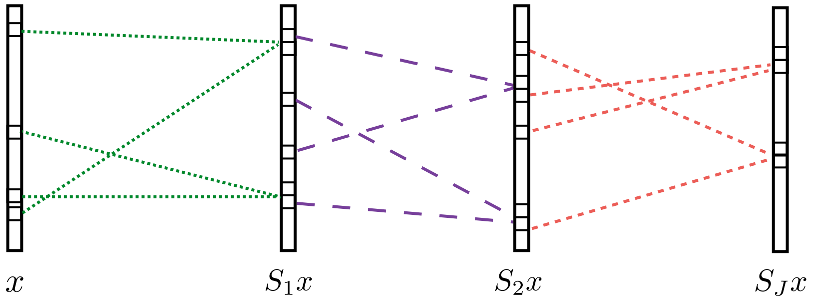

FIG. 1. A free Haar scattering network computes a layer $S_{j+1}x$ by pairing the coefficients of the previous layer S_jx, and storing the sum of coefficients and the amplitude of their difference in each pair.

The operator $|H_j|$ can thus also be interpreted as a calculation of $d/2$ permutation invariant representations of pairs of coefficients.

Applying $|H_j|$ to S_jx computes the next layer $S_{j+1}x = |H_jS_jx|$, obtained by regrouping the coefficients of $S_jx \in \mathbb{R}^d$ into $d/2$ pairs of indices written $\pi_j = \{\pi_j(2n), \pi_j(2n+1)\}_{0\leqslant n<d/2}$:

$$S_{j+1}x(2n) = S_jx(\pi_j(2n)) + S_jx(\pi_j(2n+1)), \tag{2.4}$$

$$S_{j+1}x(2n+1) = |S_jx(\pi_j(2n)) - S_jx(\pi_j(2n+1))|. \tag{2.5}$$

The pairing π_j specifies which index $\pi_j(2n+1)$ is paired with $\pi_j(2n)$, but the ordering index n is not important, as n specifies the storing position in $S_{j+1}x$ of the transformed values. The network output S_Jx is calculated with $Jd/2$ additions, subtractions and absolute values. Each coefficient of S_Jx is calculated by cascading J permutation invariant operators over pairs, and thus defines an invariant over a group of 2^J coefficients. The network depth J thus corresponds to an invariance scale 2^J. This deep network computation is illustrated in Fig. 1. The Haar transforms H_j are specified by the multi-layer pairings π_j. Section 2.4 studies an unsupervised optimization of these pairings.

2.2 *Contractions and orthogonal transforms*

Since the network is computed by iterating orthogonal linear operators, up to a normalization, and a contractive absolute value, the following theorem proves that it defines a contractive transform, which preserves the norm up to a normalization. It also proves that an orthogonal Haar scattering S_Jx applies an orthogonal matrix to x, which depends upon x and J.

THEOREM 2.1 For any $J \geqslant 0$ and any $(x, x') \in \mathbb{R}^{2d}$

$$\|S_Jx - S_Jx'\| \leqslant 2^{J/2}\|x - x'\|. \tag{2.6}$$

Moreover, $S_Jx = 2^{J/2}M_{x,J}x$, where $M_{x,J}$ is an orthogonal matrix which depends on x and J, and

$$\|S_Jx\| = 2^{J/2}\|x\|. \tag{2.7}$$

Proof. Since $S_{j+1}x = |H_jS_jx|$, where H_j is an orthogonal operator multiplied by $2^{1/2}$,

$$\|S_{j+1}x - S_{j+1}x'\| \leqslant \|H_jS_jx - H_jS_jx'\| = 2^{1/2}\|S_jx - S_jx'\|.$$

Since $S_0 x = x$, equation (2.6) is verified by induction on j. We can also rewrite

$$S_{j+1}x = |H_j S_j x| = E_{j,x} H_j x,$$

where $E_{j,x}$ is a diagonal matrix where the diagonal entries are ± 1, with a sign which depend on $S_j x$. Since $2^{-1/2} H_j$ is orthogonal, $2^{-1/2} E_{j,x} H_j$ is also orthogonal, so $M_{x,J} = 2^{-J/2} \prod_{j=1}^{J} E_{j,x} H_j$ is orthogonal, and depends on x and J. It results that $\|S_J x\| = 2^{J/2} \|x\|$. □

2.3 *Completeness with bagging*

A single Haar scattering loses information, since it applies orthogonal operators followed by an absolute value which removes the sign information. However, the following theorem proves that x can be recovered from 2^J distinct orthogonal Haar scattering transforms, computed with different pairings π_j at each layer.

Theorem 2.2 There exist 2^J different orthogonal Haar scattering transforms such that almost all $x \in \mathbb{R}^d$ can be reconstructed from the coefficients of these 2^J transforms.

This theorem is proved by observing that a Haar scattering transform is computed with permutation invariants operators over pairs. Inverting these operators recovers values of signal pairs, but not their locations. However, recombining these values on enough overlapping sets can recover their locations, and hence the original signal x. This is proved by the following lemma applied to *interlaced pairings*. We say that two pairings $\pi^0 = \{\pi^0(2n), \pi^0(2n+1)\}_{0 \leqslant n < d/2}$ and $\pi^1 = \{\pi^1(2n), \pi^1(2n+1)\}_{0 \leqslant n < d/2}$ are interlaced if there exists no strict subset Ω of $\{1, \ldots, d\}$ such that π^0 and π^1 are pairing elements within Ω. The following lemma shows that a single-layer scattering is invertible with two interlaced pairings.

Lemma 2.1 If two pairings π^0 and π^1 of $\{1, \ldots, d\}$ are interlaced, then any $x \in \mathbb{R}^d$ whose coordinates have more than two different values can be recovered from the values of $S_1 x$ computed with π^0 and the values of $S_1 x$ computed with π^1.

Proof. Let us consider a triplet n_1, n_2, n_3, where (n_1, n_2) is a pair in π^0 and (n_1, n_3) is a pair in π^1. From $S_1 x$ computed with π^0, we get

$$x(n_1) + x(n_2), \quad |x(n_1) - x(n_2)|,$$

and we saw in (2.3) that it determines the values of $\{x(n_1), x(n_2)\}$ up to a permutation. Similarly, $\{x(n_1), x(n_3)\}$ are determined up to a permutation by $S_1 x$ computed with π^1. Then unless $x(n_1) \neq x(n_2)$ and $x(n_2) = x(n_3)$, the three values $x(n_1), x(n_2), x(n_3)$ are recovered. The interlacing condition implies that π^1 pairs n_2 to an index n_4 which cannot be n_3 or n_1. Thus, the four values of $x(n_1), x(n_2), x(n_3), x(x_4)$ are specified unless $x(n_4) = x(n_1) \neq x(n_2) = x(n_3)$. This interlacing argument can be used to extend to $\{1, \ldots, d\}$ the set of all indices n_i for which $x(n_i)$ is specified, unless x takes only two values. □

Proof of Theorem 2.2. Suppose that the 2^J Haar scatterings are associated with the J hierarchical pairings $(\pi_1^{\epsilon_1}, \ldots, \pi_J^{\epsilon_J})$ where $\epsilon_j \in \{0, 1\}$, where for each j, π_j^0 and π_j^1 are two interlaced pairings of d elements. The sequence $(\epsilon_1, \ldots, \epsilon_J)$ is a binary vector taking 2^J different values.

The constraint on the signal x is that each of the intermediate scattering coefficients takes more than two distinct values, which holds for $x \in \mathbb{R}^d$ except for a union of hyperplanes, which has zero measure. Thus for almost every $x \in \mathbb{R}^d$, the theorem follows from applying Lemma 2.1 recursively to the jth level scattering coefficients for $J - 1 \geqslant j \geqslant 0$. □

Lemma 2.1 proves that only two pairings are sufficient to invert one Haar scattering layer. The argument proving that 2^J pairings are sufficient to invert J layers is quite brute-force. It is conjectured that the number of pairings needed to obtain a complete representation for almost all $x \in \mathbb{R}^d$ does not need to grow exponentially in J, but rather linearly.

Theorem 2.2 also suggests to define a signal representation by aggregating different Haar orthogonal scattering transforms. Numerical results in Section 4 show that bagging is important to improve classification accuracy.

2.4 *Unsupervised optimization of free pairing*

A Haar scattering can be combined with any Haar pairing strategies, including random pairings. However, classifications with random pairings gives poor classification results because it is not adapted to signal properties. As previously explained, an orthogonal Haar scattering is contractive. The pairing optimization amounts to find the best directions along which to perform the space compression. Contractions reduce the space volume, and hence the variance of scattering vectors, but it may also collapse together examples which belong to different classes. To maximize the 'average discriminability' among signal examples, we study an unsupervised optimization, which maximizes the variance of the scattering transform over the training set. Following [**32**], we show that it yields a representation whose coefficients are sparsely excited. This section studies 'free pairings,' as opposed to constrained pairings for data defined on graphs, studied in Section 3.1.

A Haar pairing sequence $\{\pi_j\}_{0\leqslant j<J}$ is optimized from N unlabeled data samples $\{x_i\}_{1\leqslant i\leqslant N}$. The algorithm follows a greedy layer-wise strategy, similar to many deep unsupervised learning algorithms [**3**,**20**]. It computes progressively each π_j as the depth j increases. We learn T different Haar pairings by dividing the training set $\{x_i\}_i$ into T non-overlapping subsets, and by optimizing one Haar pairing sequence per subset.

Let us suppose that Haar scattering operators H_ℓ are already computed for $1 \leqslant \ell < j$. One can thus compute S_jx for any $x \in \mathbb{R}^d$. We explain how to optimize H_j to maximize the variance of the next layer $S_{j+1}x$. The non-normalized empirical variance of S_j over N samples $\{x_i\}_{i=1}^N$ is

$$\sigma^2(S_jx) = \frac{1}{N}\sum_i \|S_jx_i\|^2 - \left\|\frac{1}{N}\sum_i S_jx_i\right\|^2.$$

The following proposition, adapted from [**32**], proves that the scattering variance decreases as the depth increases, up to a factor 2. It gives a condition on H_j to maximize the variance of the next layer.

Proposition 2.1 For any $j \geqslant 0$ and $x \in \mathbb{R}^d$, $\sigma^2(2^{-(j+1)/2}S_{j+1}x) \leqslant \sigma^2(2^{-j/2}S_jx)$. Maximizing $\sigma^2(S_{j+1}x)$ given S_jx is equivalent to finding H_j, which minimizes

$$\left\|\sum_i H_jS_jx_i\right\|^2 = \sum_n \left(\sum_i |H_jS_jx_i(n)|\right)^2. \tag{2.8}$$

Proof. Since $S_{j+1}x = |H_j S_j x|$ and $\|H_j S_j x\| = 2^{1/2}\|S_j x\|$, we have

$$\sigma^2(S_{j+1}x) = \frac{1}{N}\sum_i \|S_{j+1}x_i\|^2 - \left\|\frac{1}{N}\sum_i S_{j+1}x_i\right\|^2$$
$$= 2\frac{1}{N}\sum_{i=1}^N \|S_j x_i\|^2 - \left\|\frac{1}{N}\sum_{i=1}^N |H_j S_j x_i|\right\|^2.$$

Optimizing $\sigma^2(S_{j+1}x)$ is thus equivalent to minimizing (2.8). Moreover,

$$\sigma^2(S_{j+1}x) = 2\frac{1}{N}\sum_{i=1}^N \|S_j x_i\|^2 - \left\|H_j\frac{1}{N}\sum_{i=1}^N S_j x_i\right\|^2 + \left\|\frac{1}{N}\sum_{i=1}^N H_j S_j x_i\right\|^2 - \left\|\frac{1}{N}\sum_{i=1}^N |H_j S_j x_i|\right\|^2$$
$$= 2\frac{1}{N}\sum_{i=1}^N \|S_j x_i\|^2 - 2\left\|\frac{1}{N}\sum_{i=1}^N S_j x_i\right\|^2 + \left(\left\|\frac{1}{N}\sum_{i=1}^N H_j S_j x_i\right\|^2 - \left\|\frac{1}{N}\sum_{i=1}^N |H_j S_j x_i|\right\|^2\right)$$
$$\leqslant 2\frac{1}{N}\sum_{i=1}^N \|S_j x_i\|^2 - 2\left\|\frac{1}{N}\sum_{i=1}^N S_j x_i\right\|^2 = 2\sigma^2(S_j x),$$

which proves the first claim of the proposition. □

This proposition relies on the energy conservation $\|H_j y\| = 2^{1/2}\|y\|$. Because of the contraction of the absolute value, it proves that the variance of the normalized scattering $2^{-j/2}S_j x$ decreases as j increases. Moreover, the maximization of $\sigma^2(S_{j+1}x)$ amounts to minimize a mixed $\mathbf{l}^1$- and $\mathbf{l}^2$-norm on $H_j S_j x_i(n)$, where the sparsity $\mathbf{l}^1$-norm is along the realization index i, where as the $\mathbf{l}^2$-norm is along the feature index n of the scattering vector.

Minimizing the first $\mathbf{l}^1$-norm for n fixed tends to produce a coefficient indexed by n, which is sparsely excited across the examples indexed by i. It implies that this feature is discriminative among all examples. On the contrary, the $\mathbf{l}^2$-norm along the index n has a tendency to produce $\mathbf{l}^1$-sparsity norms, which have a uniformly small amplitude. The resulting 'features' indexed by n are thus uniformly sparse.

Because H_j preserves the norm, the total energy of coefficients is conserved:

$$\sum_n \sum_i |H_j S_j x_i(n)|^2 = 2\sum_i \|S_j x_i\|^2.$$

It results that a sparse representation along the index i implies that $H_j S_j x_i(n)$ is also sparse along n. The same type of result is thus obtained by replacing the mixed $\mathbf{l}^1$- and $\mathbf{l}^2$-norm (2.8) by a simpler $\mathbf{l}^1$-sparsity norm along both the i and n variables

$$\sum_n \sum_i |H_j S_j x_i(n)|. \tag{2.9}$$

This sparsity norm is often used by sparse auto-encoders for unsupervised learning of deep networks [3]. Numerical results in Section 4 verify that both norms have very close classification performances.

For Haar operators H_j, the $\mathbf{l}^1$-norm leads to a simpler interpretation of the result. Indeed, a Haar filtering is defined by a pairing π_j of d integers $\{1,\ldots,d\}$. Optimizing H_j amounts to optimize π_j, and hence minimize

$$\sum_n \sum_i |H_j S_j x_i(n)| = \sum_n \sum_i (S_j x_i(\pi_j(2n)) + S_j x_i(\pi_j(2n+1)) + |S_j x_i(\pi_j(2n)) - S_j x_i(\pi_j(2n+1))|).$$

But $\sum_n (S_j x(\pi_j(2n)) + S_j x(\pi_j(2n+1))) = \sum_n S_j x(n)$ does not depend upon the pairing π_j. Minimizing the $\mathbf{l}^1$-norm (2.9) is thus equivalent to minimizing

$$\sum_n \sum_i |S_j x_i(\pi_j(2n)) - S_j x_i(\pi_j(2n+1))|. \tag{2.10}$$

It minimizes the average variation within pairs, and thus tries to regroup pairs having close values.

Finding a linear operator H_j which minimizes (2.8) or (2.9) is a 'dictionary learning' problem which is in general an NP hard problem. For a Haar dictionary, we show that it is equivalent to a pair-matching problem, and can thus be solved with $O(d^3)$ operations. For both optimization norms, it amounts to finding a pairing π_j which minimizes an additive cost

$$C(\pi_j) = \sum_n C(\pi_j(2n), \pi_j(2n+1)), \tag{2.11}$$

where $C(\pi_j(2n), \pi_j(2n+1)) = \sum_i |H_j S_j x_i(n)|$ for (2.9) and $C(\pi_j(2n), \pi_j(2n+1)) = (\sum_i |H_j S_j x_i(n)|)^2$ for (2.8). This optimization can be converted to a *maximum matching* problem on a weighted complete graph, and thus can be solved exactly by the Edmonds' 1960 Blossom algorithm [13], which is of $O(d^3)$ complexity in the worst case. The program used in this paper is based on an implementation by E. Rothberg of the algorithm as in [15], and the source code is available at http://www.zib.de/en/services/web-services/mathprog/matching.html. The Greedy method obtains a $\frac{1}{2}$-approximation in $O(d^2)$ time [37]. Randomized approximation similar to [24] could also be adapted to achieve a complexity of $O(d \log d)$ for very large size problems.

2.5 *Supervised feature selection and classification*

A bagged Haar scattering transform $\Phi x = \{S_J^{(t)} x\}_{t=1}^T$ aggregates T Haar scatterings $S_J^{(t)}$, computed with different pairing sequences $\{\pi_j^{(t)}\}_{0 \leqslant j \leqslant J-1}$ for $1 \leqslant t \leqslant T$. It may be calculated with the free pairing optimization of Section 2.4 or with the graph pairing of Section 3.1. The classification algorithm reduces the dimensionality of Φx with a supervised feature selection, and it computes a supervised classification by applying a Gaussian SVM to this reduced representation.

The supervised feature selection is computed with an orthogonal least square (OLS) forward selection [9]. It selects K coefficients in Φx to discriminate each class c from all other classes, and decorrelates these features. Discriminating a class c from all other classes amounts to approximating the indicator function

$$f_c(x) = \begin{cases} 1 & \text{if } x \text{ belongs to class } c \\ 0 & \text{otherwise.} \end{cases}$$

An orthogonal least square linearly approximates $f_c(x)$ with a K-scattering coefficients $\{\phi_{p_k}\}_{k \leqslant K}$, which are selected one at a time. To avoid correlations between selected features, it includes a Gram-Schmidt orthogonalization, which de-correlates the scattering dictionary relatively to previously selected features. We denote by $\Phi^k x = \{\tilde{\phi}^k_p x\}_p$ the scattering dictionary, which was orthogonalized, and hence de-correlated relatively to the first k selected scattering features. For $k = 0$, we have $\Phi^0 x = \Phi x$. At the $k + 1$ iteration, we select $\phi^k_{p_k} x \in \Phi^k x$, which minimizes the mean-square error over training samples:

$$\sum_i \left(f_c(x_i) - \sum_{\ell=0}^{k} \alpha_\ell \phi^\ell_{p_\ell} x_i \right)^2 . \tag{2.12}$$

Because of the orthonormalization step, the linear regression coefficients are

$$\alpha_\ell = \sum_i f_c(x_i) \phi^\ell_{p_\ell} x_i,$$

and

$$\sum_i \left(f_c(x_i) - \sum_{\ell=0}^{k} \alpha_\ell \phi^\ell_{p_\ell} x_i \right)^2 = \sum_i |f_c(x_i)|^2 - \sum_{\ell=0}^{k} \alpha_\ell^2 .$$

The error (2.12) is thus minimized by choosing $\phi^k_{p_{k+1}} x$ having a maximum correlation:

$$\alpha_k = \sum_i f_c(x_i) \phi^k_{p_k} x_i = \arg\max_p \left(\sum_i f_c(x_i) \phi^k_p x_i \right).$$

The scattering dictionary is then updated by orthogonalizing each of its element relatively to the selected scattering feature $\phi^k_{p_k x}$:

$$\phi^{k+1}_p x = \phi^k_p x - \left(\sum_i \phi^k_p x_i \phi^k_{p_k} x_i \right) \phi^k_{p_k} x.$$

This orthogonal least square regression greedily selects the K de-correlated scattering features $\{\phi^k_{p_k} x\}_{0 \leqslant k < K}$ for each class c. For a total of C classes, the union of all these features defines a dictionary of size $M = KC$. They are linear combinations of the original Haar scattering coefficients $\{\phi_p x\}_p$. In the context of a deep neural network, this dimension reduction can be interpreted as a last fully connected network layer, which takes in input the bagged scattering coefficients and outputs a vector of size M. The parameter M optimizes the bias versus variance trade-off. It may be set *a priori* or adjusted by cross validation in order to yield a minimum classification error, at the output of the Gaussian kernel SVM classifier.

A Gaussian kernel SVM classifier is applied to the M-dimensional orthogonalized scattering feature vectors. The Euclidean norm of this vector is normalized to 1. In all applications of Section 4, M is set to 10^3, and hence remains large. Since the feature vectors lie on a high-dimensional unit sphere, the standard deviation σ of the Gaussian kernel SVM must be of the order of 1. Indeed, a Gaussian kernel SVM performs its classification by fitting a separating hyperplane over different balls of radius of radius σ. If $\sigma \ll 1$, then the number of balls covering the unit sphere grow like σ^{-M}. Since M is large, σ must remain in the order of 1 to insure that there are enough training samples to fit a hyperplane in each ball.

3. Haar scattering on graphs

When data samples lie on a graph, we introduce a pairing optimization with constraints imposed by graph structures. The resulting graph Haar scattering is a particular case of orthogonal Haar scattering.

This Haar pairing learns the graph connectivity from data variability. The consistency of this inference is studied. The decay of scattering coefficients is characterized as a function of their nonlinearity order.

3.1 *Unsupervised optimization of pairings on graphs*

The free pairing introduced in Section 2.4 associates any two elements of an internal network layer S_jx. In contrast, graph Haar scattering is constructed by pairing elements according to their position in the graph. Section 2.4 introduces two criteria to optimize the pairing π_j. We concentrate on the $\mathbf{l}^1$-norm minimization, which has a simpler expression.

We denote by V the set of d vertices of this graph, and assume that d is a power of 2. The input network layer is $S_0x(n,0) = x(n)$ as before. The pairing of π_0 is optimized as in the free pairing. The resulting $S_1x(n,q)$ is an array of size $d/2 \times 2$. The index $n \in \{1,\dots,d/2\}$ specifies each pair of nodes paired by π_0, and $q \in \{0,1\}$ indicates if the coefficient is computed with an addition or a subtraction and an absolute value. The next π_1 is pairing $d/2$ pairs, namely

$$\pi_1 = \{(\pi_1(2n), \pi_1(2n+1))\}_{0\leqslant n<d/4},$$

and is optimized by minimizing

$$\sum_{i=1}^{N} \sum_{n=0}^{d/4} \sum_{q=0\text{or}1} |S_1x_i(\pi_1(2n), q) - S_1x_i(\pi_1(2n+1), q)|.$$

Coefficients $\{S_1x(n,q)\}_n$ are thus paired for a fixed $q = 0$ or $q = 1$, and the pairing is the same for $q = 0$ and $q = 1$.

We show recursively that S_jx is an array $S_jx(n,q)$ of size $2^{-j}d \times 2^j$. For each $j \geqslant 0$, the row index $n \in \{1,\dots,2^{-j}d\}$ is a 'spatial' index of a set of $V_{j,n}$ of 2^j graph vertices. The column index $q \in \{0,\dots,2^j - 1\}$ indicates different Haar scattering coefficients computed from the values of x in $V_{j,n}$. As j increases from 0 to $J \leqslant \log_2(d)$, the family $\{V_{j,n}\}_{1\leqslant n\leqslant 2^{-j}d}$ is a hierarchical dyadic partition of the graph. Each $S_{j+1}x$ is computed by calculating a pairing π_j of the $2^{-j}d$ rows of S_jx, by minimizing

$$\sum_{i=1}^{N} \sum_{n=0}^{d2^{-(j+1)}} \sum_{q=0}^{2^j-1} |S_jx_i(\pi_j(2n), q) - S_jx_i(\pi_j(2n+1), q)|. \tag{3.1}$$

The row pairing

$$\pi_j = \{(\pi_j(2n), \pi_j(2n+1))\}_{0\leqslant n<2^{-(j+1)}d}, \tag{3.2}$$

is pairing each $(\pi_j(2n), q)$ with $(\pi_j(2n+1), q)$ for $0 \leqslant q < 2^j$. Applying (2.2) to each pair gives

$$S_{j+1}x(n, 2q) = S_jx(\pi_j(2n), q) + S_jx(\pi_j(2n+1), q) \tag{3.3}$$

and

$$S_{j+1}x(n, 2q+1) = |S_jx(\pi_j(2n), q) - S_jx(\pi_j(2n+1), q)|. \tag{3.4}$$

The graph pairing strategy gives a Haar scattering network illustrated in Fig. 2. Equation (3.1) means that the optimal pairing regroups vertex sets $V_{j,\pi_j(2n)}$ and $V_{j,\pi_j(2n+1)}$, whose scattering coefficients have a minimum total variation.

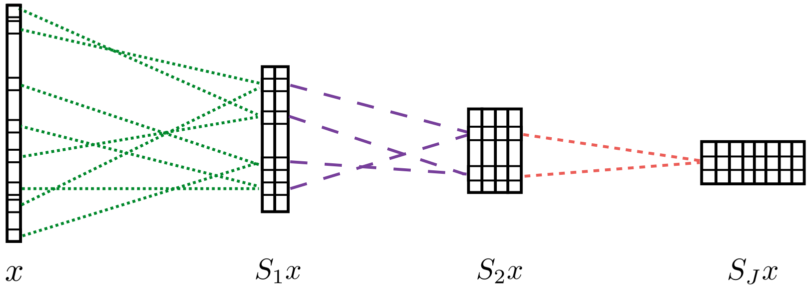

FIG. 2. A graph Haar scattering computes $S_{j+1}x$ by pairing the rows of the previous layer S_jx. Each row corresponds to a neighborhood having 2^j nodes in the graph, resulting from graph pairings at previous layers. For each pair of rows, the sum and the absolute value of their difference is stored twice in a bigger row.

Let $V_{0,n} = \{n\}$ for $n \in V$. For any $j \geqslant 0$ and $n \in \{1, \dots, 2^{-j-1}d\}$, we define

$$V_{j+1,n} = V_{j,\pi_j(2n)} \cup V_{j,\pi_j(2n+1)}. \tag{3.5}$$

We verify by induction on j that for each j, $V = \cup_n V_{j,n}$ defines a partition, and each $V_{j,n}$ is a set of 2^j vertices. We say that two non-overlapping subsets V_1 and V_2 of V are connected if at least one element of V_1 is connected to one element of V_2. The induction (3.5) defines sets $V_{j,n}$ with connected nodes in the graph if for all j and n, each pair $(\pi_j(2n), \pi_j(2n+1))$ regroups two sets $V_{j,\pi_j(2n)}$ and $V_{j,\pi_j(2n+1)}$ which are connected. There are many possible connected dyadic partitions of any given graph. Fig. 3(a,b) shows two different examples of connected graph partitions.

When the graph is known in advance, one can use the graph connectivity to define a sequence of graph pairings by only pairing connected neighborhoods of nodes on the graph. There exist many graph pairings which satisfy such a condition. For example, in images sampled on a square grid, a pixel is connected with eight neighboring pixels. A graph Haar scattering can be computed by pairing neighbor image pixels, alternatively along rows and columns as the depth j increases. When j is even, each $V_{j,n}$ is then a square group of 2^j pixels, as illustrated in Fig. 3(c). Shifting such a partition defines a new partition. Neighbor pixels can also be grouped in the diagonal direction, which amounts to rotate the sets $V_{j,n}$ by $\pi/4$ to define a new dyadic partition. Each of these partitions defines a different graph Haar scattering. Section 4 compares a Haar scattering computed on a known graph with pairings obtained by unsupervised learning, for image classification. Section 3.4 studies conditions so that pairings computed from data samples, with no graph knowledge, define a connected partition in the underlying graph.

3.2 *Scattering order*

The *order* m of a scattering coefficient is the number of subtractions involved in its computation, followed by absolute values. Subtractions compute coefficients which may be positive or negative. Their range of variation is contracted by the absolute value. As a consequence, we show that the amplitude of a scattering coefficient of order m has a fast decay as m increases. Classifications are thus computed from low-order scattering coefficients. The following proposition relates the column index q to the order m of a scattering coefficient $S_jx(n,q)$.

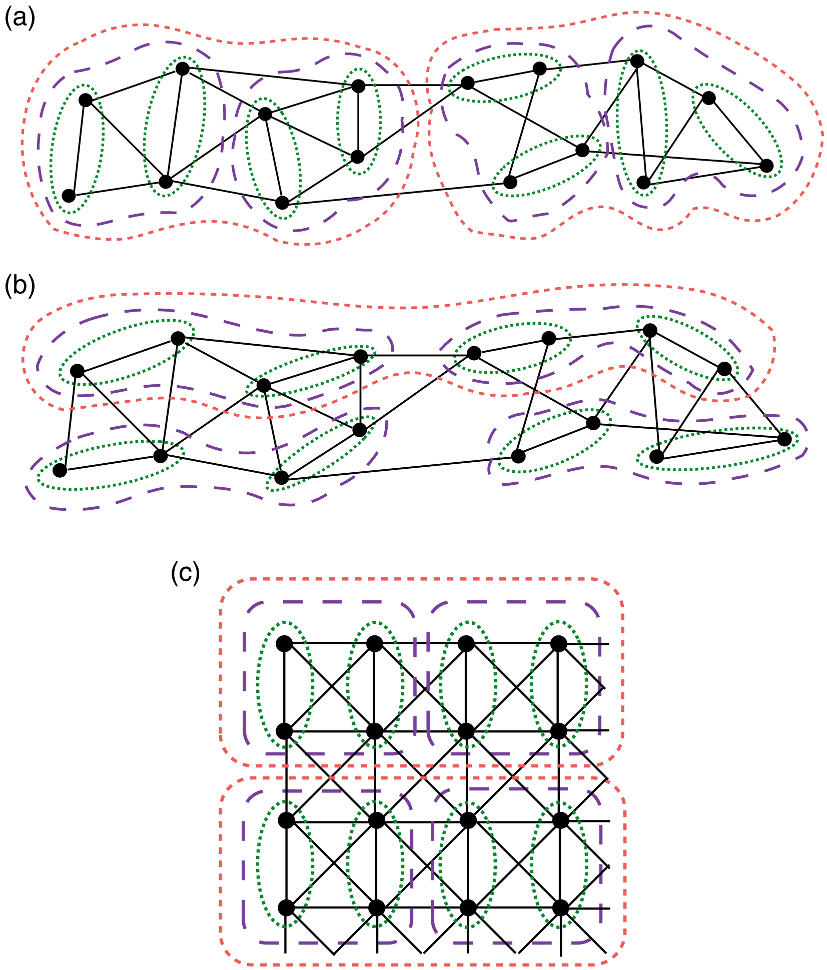

FIG. 3. Two different examples of hierarchical partitions of a graph into connected sets $V_{j,n}$ of size 2^j, for $j=1$ (green), $j=2$ (purple) and $j=3$ (red). (c) Hierarchical partitions on a square image grid.

PROPOSITION 3.1 If $q=0$, then $S_jx(n,q)$ is a coefficient of order 0. Otherwise, $S_jx(n,q)$ is a coefficient of order $m \leqslant j$ if there exists $0 \leqslant j_1 < \cdots < j_m < j$ such that

$$q=\sum_{k=1}^{m} 2^{j-j_k}. \tag{3.6}$$

There are $\binom{j}{m}2^{-j}d$ coefficients of order m in S_jx.

Proof. This proposition is proved by induction on j. For $j=0$ all coefficients are of order 0 since $S_0x(n,0)=x(n)$. If $S_jx(n,q)$ is of order m, then (3.3) and (3.4) imply that $S_{j+1}x(n,2q)$ is of order m and $S_{j+1}x(n,2q+1)$ is of order $m+1$. It results that (3.6) is valid for $j+1$ if is valid for j.

TABLE 1 *$\sigma^2_{m,J}$ is the normalized variance of all order m coefficients in $S_J x$, computed for a Gaussian white noise x with $J = 5$. It decays approximately like $(1-\frac{2}{\pi})^m \cdot \binom{j}{m}$*

m	1	2	3	4	5
$\sigma^2_{m,J}$	1.8	1.4	5.8×10^{-1}	1.2×10^{-1}	1.2×10^{-2}
$(1-\frac{2}{\pi})^m \cdot \binom{j}{m}$	1.8	1.3	4.8×10^{-1}	8.7×10^{-2}	6.3×10^{-3}

The number of coefficients $S_j x(n,q)$ of order m corresponds to the number of choices for q, and hence for $0 \leqslant j_1 < \cdots < j_m < j$, which is $\binom{j}{m}$. This must be multiplied by the number of indices n, which is $2^{-j}d$. □

The amplitude of scattering coefficients typically decreases exponentially when the scattering order m increases, because of the contraction produced by the absolute value. High-order scattering coefficients can thus be neglected. This is illustrated by considering a vector x of independent Gaussian random variables of variance 1. The value of $S_j x(n,q)$ only depends upon the values of x in $V_{j,n}$. Since $V_{j,n}$ does not intersect with $V_{j,n'}$ if $n \neq n'$, we derive that $S_j(n,q)$ and $S_j(n',q)$ are independent. They have same mean and same variance because x is identically distributed. Scattering coefficients are iteratively computed by adding pairs of such coefficients, or by computing the absolute value of their difference. Adding two independent random variables multiplies their variance by 2. Subtracting two independent random variables of same mean and variance yields a new random variable, whose mean is zero and whose variance is multiplied by 2. Taking the absolute value reduces the variance by a factor which depends upon its probability distribution. If this distribution is Gaussian, then this factor is $1-2/\pi$. If we suppose that this distribution remains approximately Gaussian, then applying m absolute values reduces the variance by approximately $(1-2/\pi)^m$. Since there are $\binom{j}{m}$ coefficients of order m, their total normalized variance $\sigma^2_{m,J}$ is approximated by $\binom{j}{m}(1-2/\pi)^m$. Table 1 shows that $\binom{j}{m}(1-2/\pi)^m$ is indeed of the same order of magnitude as the value $\sigma^2_{m,J}$ computed numerically. This variance becomes much smaller for $m > 4$. This observation remains valid for large classes of signals x. Scattering coefficients of order $m > 4$ usually have a negligible energy, and are thus removed in classification applications.

3.3 *Cascades of Haar wavelets on a graph*

We now prove that graph Haar scattering coefficients of order m are obtained by cascading m orthogonal Haar wavelet transforms defined on the graph.

Section 3.1 shows that a graph Haar scattering is constructed over dyadic partitions $\{V_{j,n}\}_n$ of V, which are obtained by progressively aggregating vertices by pairing $V_{j+1,n} = V_{j,\pi_j(2n)} \cup V_{j,\pi_j(2n+1)}$. We denote by $1_{V_{j,n}}(v)$ the indicator function of $V_{j,n}$ in V. A Haar wavelet computes the difference between the sum of signal values over two aggregated sets:

$$\psi_{j+1,n} = 1_{V_{j,\pi_j(2n)}} - 1_{V_{j,\pi_j(2n+1)}}. \tag{3.7}$$

Inner products between signals defined on V are written

$$\langle x, x' \rangle = \sum_{v \in V} x(v) x'(v).$$

For any $2^J < d$,

$$\{1_{V_{J,n}}\}_{0 \leqslant n < 2^{-J}d} \cup \{\psi_{j,n}\}_{0 \leqslant n < 2^{-j}d, 0 \leqslant j < J} \tag{3.8}$$

is a family of d orthogonal Haar wavelets which define an orthogonal basis of $\mathbb{R}^d$. The following theorem proves that order $m+1$ coefficients are obtained by computing the orthogonal Haar wavelet transform of coefficients of order m. The proof is in Appendix A.

Theorem 3.1 Let $q = \sum_{k=1}^{m} 2^{j-j_k}$ with $j_1 < \cdots < j_m \leqslant j$. If $j_{m+1} > j_m$, then for each $n \leqslant 2^{-j-1}d$

$$S_j x(n, q + 2^{j-j_{m+1}}) = \sum_{\substack{p \\ V_{j_{m+1},p} \subset V_{j,n}}} |\langle \bar{S}_{j_m} x(\cdot, 2^{j_m - j} q), \psi_{j_{m+1},p} \rangle|, \tag{3.9}$$

with

$$\bar{S}_{j_m} x(., q') = \sum_{n=0}^{2^{-j_m}d-1} S_{j_m} x(n, q') 1_{V_{j_m,n}}.$$

If $q = \sum_{k=1}^{m} 2^{j-j_k}$ and $j_{m+1} > j_m$, then $S_{j_m} x(n, 2^{j_m - j} q)$ are coefficients of order m, whereas $S_j x(n, q + 2^{j-j_{m+1}})$ is a coefficient of order $m+1$. Equation (3.9) proves that a coefficient of order $m+1$ is obtained by calculating the wavelet transform of scattering coefficients of order m, and summing their absolute values. A coefficient of order $m+1$ thus measures the averaged variations of the mth order scattering coefficients on neighborhoods of size $2^{j_{m+1}}$ in the graph. For example, if x is constant in a $V_{j,n}$ then $S_\ell x(n, q) = 0$ if $\ell \leqslant j$ and $q \neq 0$.

To further compare graph Haar scattering with Haar wavelet transforms, observe that if the absolute value in (3.4) is removed, these equations iterate linear Haar filters and define an orthogonal Walsh transform [**12**]. However, the absolute value operator completely modifies the properties of this transform from a Haar wavelet transform. The following proposition proves that graph Haar scattering is a transformation on a hierarchical grouping of the graph vertices, derived from the graph pairing (3.2).

Proposition 3.2 The coefficients $\{S_J x(n, q)\}_{0 \leqslant q < 2^J}$ are computed by applying a Hadamard matrix to the restriction of x to $V_{J,n}$. This Hadamard matrix depends on x, J and n.

Proof. Theorem 2.1 proves that $\{S_J x(n, q)\}_{0 \leqslant q < 2^J}$ is computed by applying an orthogonal transform to x. To prove that it is a Hadamard matrix, it is sufficient to show that its entries are ± 1. We verify by induction on $j \leqslant J$ that $S_j x(n, q)$ only depends on restriction of x to $V_{j,n}$, by applying (3.4) and (3.3) together with (3.5). We also verify that each $x(v)$ for $v \in V_{j,n}$ appears exactly once in the calculation, with an addition or a subtraction. Because of the absolute value, the addition or subtraction, which corresponds to 1 and -1 in the Hadamard matrix, therefore depends upon x, J and n. □

A graph Haar scattering can thus be interpreted as an adaptive Hadamard transform over groups of vertices, which outputs positive coefficients. Walsh matrices are particular cases of Hadamard matrices.

3.4 *Inferring graph connectivity from data*

In many applications, the graph connectivity is unknown. The inference of unknown graphs from data arises in many problems such as social networks, chemical and biological networks, and many others. It has been shown in [**39**] that correlation of pixel intensities can be used numerically to reconstruct an image grid, but recovering connections in an unknown graph is generally an NP-hard problem. For graph Haar scattering, the learning algorithm amounts to computing dyadic partitions, where scattering coefficients have a minimum total variation. As explained in the end of Section 2.4, this optimization at each network layer has a polynomial complexity, so the overall complexity is polynomial. Indeed, learning a connected dyadic partition is easier than learning the full connectivity of the graph.

We study the consistency of this pairing algorithm with respect to the underlying graph for data which are realizations of Gaussian stationary processes. We show that this estimation involves no curse of dimensionality. Consistency implies that a graph Haar scattering computes an invariant representation from local multiscale signal variations on the graph.

Suppose that the N training samples x_i are independent realizations of a random vector x. To guarantee that this pairing finds connected sets, we must insure that the total variation minimization favors regrouping neighborhood points, which means that x has some form of regularity on the graph. We also need N to be sufficiently large so that this minimization finds connected sets with high probability, despite statistical fluctuations. Avoiding the curse of dimensionality means that N does not need to grow exponentially with the signal dimension d to recover connected sets with high probability.

To attack this problem mathematically, we consider a very particular case, where signals are defined on a ring graph, and are thus d periodic. Two indices n and n' are connected if $|n-n'| = 1 \mod d$. We study the optimization of the first network layer for $j=0$, where $S_0 x(n,q) = x(n)$. The minimization of (3.1) amounts to computing a pairing π, which minimizes

$$\sum_{i=1}^{N} \left(\sum_{n=0}^{d/2-1} |x_i(\pi(2n)) - x_i(\pi(2n+1))| \right). \tag{3.10}$$

This pairing is connected if and only if for all n, $|\pi(2n) - \pi(2n+1)| = 1 \mod d$.

The regularity and statistical fluctuations of $x(n)$ are controlled by supposing that x is a circular stationary Gaussian process. The stationarity implies that its covariance matrix $\mathrm{Cov}(x(n), x(m)) = \Sigma(n,m)$ depends on the distance between points $\Sigma(n,m) = \rho((n-m) \mod d)$. The average regularity depends upon the decay of the correlation $\rho(u)$. We denote by $\|\Sigma\|_{\mathrm{op}}$ the sup operator norm of Σ. The following theorem proves that the training size N must grow like $d \log d$ in order to compute an optimal pairing with a high probability. The constant is inversely proportional to a normalized 'correlation gap', which depends upon the difference between the correlation of neighborhood points and more far away points. It is defined by

$$\Delta = \left(\sqrt{1 - \frac{\max_{n \geqslant 2} \rho(n)}{\rho(0)}} - \sqrt{1 - \frac{\rho(1)}{\rho(0)}} \right)^2. \tag{3.11}$$

THEOREM 3.2 Given a circular stationary Gaussian process with $\Delta > 0$, the pairing which minimizes the empirical total variation (3.10) has probability larger than $1-\epsilon$ to be connected if

$$N > \frac{\pi^3 \|\Sigma\|_{\mathrm{op}}}{2\Delta} d(3 \log d - \log \varepsilon). \tag{3.12}$$

The proof is based on the Gaussian concentration inequality for Lipschitz function [33,36], and is left to Appendix B. Fig. 4 displays numerical results obtained with a Gaussian stationary process of dimension d, where $\rho(1)/\rho(0) = 0.44$ and $\max_{n \geqslant 2} \rho(n)/\rho(0) = 0.06$. The gray level image gives the probability that a pairing is connected when computing this pairing by minimizing the total variation (3.10), as a function of the dimension d and of the number N of training samples. The black and white points correspond to probabilities 0 and 1, respectively. In this example, we see that the optimization gives a connected pairing with probability $1-\epsilon$ for N increasing almost linearly with d, which is

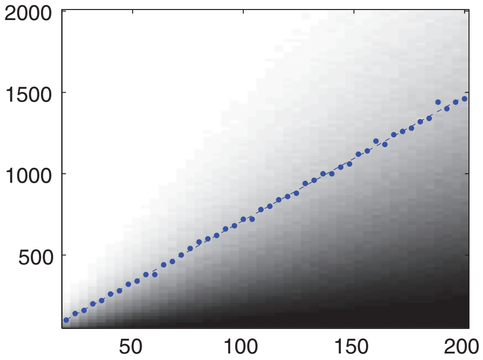

FIG. 4. Each image pixel gives the probability that the total variation minimization (3.10) finds pairs which are all connected, when x is a Gaussian stationary vector. It is computed as a function of the dimension d of the vector (horizontal axis) and of the number N of training samples (vertical axis). Black and white points are probabilities, respectively, equal to 0 and 1. The blue dotted line corresponds to a probability 0.8.

illustrated by the nearly straight line of dotted points corresponding to $\epsilon = 0.2$. The theorem gives an upper bound which grows like $d \log d$, though the constant involved is not tight.

For layer $j > 1$, $S_j x(n,q)$ is no longer a Gaussian random vector due to the absolute value non-linearity. However, the result can be extended using a Talagrand-type concentration argument instead of the Gaussian concentration. Numerical experiments presented in Section 4 show that this approach does recover the connectivity of high-dimensional images with a probability close to 100% for $j \leqslant 3$, and that the probability decreases as j increases. This seems to be due to the fact that the absolute value contractions reduce the correlation gap $\varDelta$ between connected coefficients and more far away coefficients when j increases.

A Haar scattering has some similarities with the work on provable bounds for deep representations [2], which studies polynomial-time algorithms for learning deep networks. In this model, correlations among activations of neurons on the same layer are used to recover connections in the neural network, which is seen as a hidden graph. The optimization of Haar pairings in Haar scattering also uses such correlations of activations. As has been pointed out in [2], using correlation is ‘a new twist on the old Hebbian rule that *things that fire together wire together*’. The algorithm in [2] is computed layer per layer, starting from the bottom, as in the Haar scattering learning. In [2], the underlying true network is assumed to be sparse, whereas in a Haar scattering the estimated connectivity is sparse by construction, with two out-going edges per node. Learning deep neural network is generally NP hard, whereas both Haar scattering and the algorithm [2] have a polynomial complexity. Both models have very different settings, but polynomial complexity arises because they are based on sparsely connected deep neural networks.

4. Numerical experiments

Haar scattering representations are tested on classification problems, over images sampled on a regular grid or an irregular graph. We consider the cases where the grid or the graph geometry is known *a priori*, or inferred by unsupervised learning. The efficiency of free and graph Haar scattering architectures are compared with state-of-the-art classification results obtained by deep neural networks.

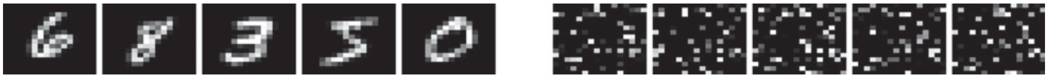

FIG. 5. MNIST images (left) and images after random pixel permutations (right).

A Haar scattering classification involves few parameters which are reviewed. The scattering scale $2^J \leqslant d$ is the permutation invariance scale. Scattering coefficients are computed up to the maximum order m, which is set to 4 in all experiments. Indeed, higher order scattering coefficient have a negligible relative energy, which is below 1%, as explained in Section 3.2. The unsupervised learning algorithm computes T different Haar scattering transforms by subdividing the training set in T subsets. Increasing T decreases the classification error, but it increases computations. The error decay becomes negligible for $T \geqslant 40$. The supervised dimension reduction selects a final set of M orthogonalized scattering coefficients. We set $M = 1000$ in all numerical experiments.

4.1 *Classification of image digits in MNIST*

NIST is a data basis with 6×10^4 hand-written digit images of size $d \leqslant 2^{10}$. There are 10 classes (one per digit) with 5×10^4 images for training and 10^4 for testing. Examples of MNIST images are shown in Fig. 5. To test the classification performances of a Haar scattering when the geometry is unknown, we scramble all image pixels with the same unknown random permutations, as shown in Fig. 5.

When the image geometry is known, i.e. using non-scrambled images, the best MNIST classification results without data augmentation are given in Table 2(a). Deep convolution networks with supervised learning reach an error of 0.53% [27], and unsupervised learning with sparse coding gives a slightly larger error of 0.59% [25]. A wavelet scattering computed with iterated Gabor wavelet transforms yields an error of 0.46% [6].

For a known image grid geometry, we compute a graph Haar scattering by pairing neighbor image pixels. It builds hierarchical square subsets $V_{j,n}$ illustrated in Fig. 3(c). The invariance scale is $2^J = 2^6$, which corresponds to blocks of 8×8 pixels. Random shifts and rotations of these pairing define $T = 64$ different Haar scattering transforms. The supervised classifier of Section 2.5 applied to this graph Haar scattering yields an error of 0.59%.

MNIST digit classification is a relatively simple problem, where the main source of variability is due to deformations of hand-written image digits. In this case, supervised convolution networks, sparse coding, Gabor wavelet scattering and orthogonal Haar scattering have nearly the same classification performances. The fact that a Haar scattering is only based on additions and subtractions does not affect its efficiency.

For scrambled images, the connectivity of image pixels is unknown, and needs to be learned from data. Table 2(b) gives the classification results of different learning algorithms. The smallest error of 0.79% is obtained with a Deep Belief Net optimized with a supervised backpropagation. Unsupervised learning of $T = 50$ graph Haar scatterings followed by a feature selection and a supervised SVM classifier produces an error of 0.90%. The variance of the classification accuracy with respect to the random splitting of training sets when learning the Haar pairing is negligible. Figure 6 gives the classification error rate as a function of T, for different values of maximum scale J. The error rates decrease slowly for $T > 10$, and do not improve beyond $T = 50$, which is much smaller than 2^J.

The unsupervised learning computes connected dyadic partitions $V_{j,n}$ from scrambled images by optimizing an $\mathbf{l}^1$-norm. At scales $1 \leqslant 2^j \leqslant 2^3$, 100% of these partitions are connected in the original

TABLE 2 *Percentage of errors for the classification of MNIST images, obtained by different algorithms*

(a) Known geometry	
Convolutional nets (supervised) [27]	0.53
Sparse coding (unsupervised) [25]	0.59
Gabor scattering [6]	**0.43**
Graph Haar scattering	0.59
(b) Unknown geometry	
Maxout MLP + dropout [17]	0.94
Deep convex net. [45]	0.83
DBM + dropout [22]	**0.79**
Graph Haar scattering	0.90

The bold values indicate the best performance.

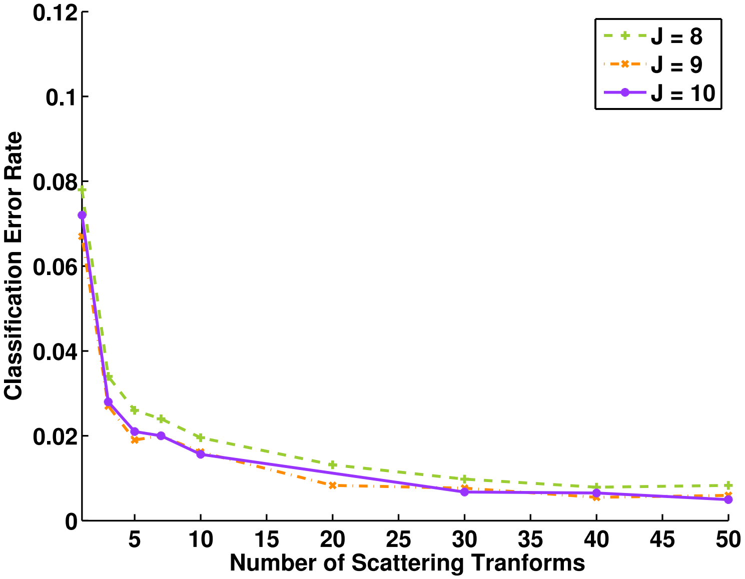

FIG. 6. Unsupervised Haar scattering classification error for MNIST, as a function of the number T of scattering transforms, for networks of depth $J = 8, 9, 10$.

image grid, which proves that the geometry is well estimated at these scales. This is only evaluated on meaningful pixels which do not remain zero on all training images. For $j = 4$ and $j = 5$, the percentages of connected partitions are 85 and 67%, respectively. The percentage of connected partitions decreases because long-range correlations are weaker.

A free orthogonal Haar scattering does not impose any condition on pairings. It produces a minimum error of 1% for $T = 20$ Haar scattering transforms, computed up to the depth $J = 7$. This error rate is higher because the supplement of freedom in the pairing choice increases the variance of the estimation.

FIG. 7. Examples of CIFAR-10 images in the classes of ‘cars’, ‘dogs’ and ‘boats’.

TABLE 3 *Percentage of errors for the classification of CIFAR-10 images, obtained by different algorithms*

(a) Known geometry	
Convolutional nets (supervised state-of-the-art) [28]	**9.8**
RFL (unsupervised state-of-the-art) [23]	16.9
Roto-translation scattering [35]	17.8
Graph Haar scattering	21.3
(b) Unknown geometry	
Fastfood [26]	37.6
Fastfood FFT [26]	36.9
Random kitchen sinks [26]	37.6
Graph Haar scattering	**27.3**

The bold values indicate the best performance.

4.2 *CIFAR-10 images*

CIFAR-10 is a data basis of tiny color images of 32×32 pixels. It includes 10 classes, such as ‘dogs’, ‘cars’, ‘ships’ with a total of 5×10^4 training examples and 10^4 testing examples. There are much more intra-class variabilities than in MNIST digit images, as shown in Fig. 7. The three color bands are represented with Y, U, V channels, and scattering coefficients are computed independently in each channel.

When the image geometry is known, a graph Haar scattering is computed by pairing neighbor image pixels. The best performance is obtained at the scale $2^J = 2^6$, which is below the maximum scale $d = 2^{10}$. Similarly to MNIST, we compute $T = 64$ connected dyadic partitions for randomly translated and rotated grids. After dimension reduction, the classification error is 21.3%. This error is above state-of-the-art results of unsupervised learning algorithms by ~20%, but it involves no learning. A minimum error rate of 16.9% is obtained by Receptive Field Learning [23]. The Haar scattering error is also above the 17.8% error obtained by a roto-translation invariant wavelet scattering network [35], which computes wavelet transforms along translation and rotation parameters. Supervised deep convolution networks provide an important improvement over all unsupervised techniques and reach an error of 9.8%. The study of these supervised networks is however beyond the scope of this paper. Results are summarized in Table 3(a).

When the image grid geometry is unknown, because of random scrambling, Table 3(a) summarizes results with different algorithms. For unsupervised learning with graph Haar scattering, the minimum classification error is reached at the scale $2^J = 2^7$, which maintains some localization information on scattering coefficients. With $T = 10$ connected dyadic partitions, the error is 27.3%. Table 3(b) shows that it is 10% below previously reported results on this data basis.

Nearly 100% of the dyadic partitions $V_{j,n}$ computed from scrambled images are connected in the original image grid, for $1 \leqslant j \leqslant 4$, which shows that the multiscale geometry is well estimated at these

TABLE 4 *Percentage of errors for the classification of CIFAR-100 images with known geometry, obtained by different algorithms*

Convolutional nets (supervised state-of-the-art) [28]	**34.6**
NOMP (unsupervised state-of-the-art) [29]	39.2
Gabor scattering [35]	43.7
Graph Haar scattering	47.4

The bold values indicate the best performance.

TABLE 5 *Percentage of errors for the classification of MNIST, CIFAR-10 and CIFAR-100 images with a graph or a free Haar scattering, for unsupervised computed by minimizing a mixed $\mathbf{l}^1/\mathbf{l}^2$-norm or an $\mathbf{l}^1$-norm*

	Graph, $\mathbf{l}^1$	Graph, $\mathbf{l}^1/\mathbf{l}^2$	Free, $\mathbf{l}^1$	Free, $\mathbf{l}^2/\mathbf{l}^1$
MNIST	**0.91**	0.95	1.09	1.02
CIFAR-10	28.8	**27.3**	29.2	29.3
CIFAR-100	**52.5**	53.1	56.3	56.1

The bold values indicate the best performance.

fine scales. For $j = 5, 6$ and 7, the proportions of connected partitions are 98, 93 and 83%, respectively. As for MNIST images, the connectivity estimation becomes less precise at large scales. Similarly to MNIST, a free Haar scattering yields a higher classification error of 29.2%, with $T = 20$ scattering transforms up to layer $J = 6$.

4.3 *CIFAR-100 images*

CIFAR-100 also contains tiny color images of the same size as CIFAR-10 images. It has 100 classes containing 600 images each, of which 500 are training images and 100 are for testing. Our tests on CIFAR-100 follow the same procedures as in Section 4.2. The three color channels are processed independently.

When the image grid geometry is known, the results of a graph Haar scattering are summarized in Table 4. The best performance is obtained with the same parameter combination as in CIFAR-10, which is $T = 64$ and $2^J = 2^6$. After dimension reduction, the classification error is 47.4%. As in CIFAR-10, this error is ~20% larger than state-of-the-art unsupervised methods, such as a Non-negative OMP (39.2%) [29]. A roto-translation wavelet scattering has an error of 43.7%. Deep convolution networks with supervised training produce again a lower error of 34.6%.

For scrambled images of unknown geometry, with $T = 10$ transforms and a depth $J = 7$, a graph Haar scattering has an error of 52.7%. A free Haar orthogonal scattering has a higher classification error of 56.1%, with $T = 10$ scattering transforms up to layer $J = 6$. No such result is reported with another algorithm on this data basis.

On all tested image databases, graph Haar scattering has a consistent 7–10% performance advantage over 'free' Haar scattering, as shown in Table 5. For orthogonal Haar scattering, all reported errors were calculated with an unsupervised learning which minimizes the $\mathbf{l}^1$-norm (2.9) of scattering coefficients, layer per play. As expected, Table 5 shows that minimizing a mixed $\mathbf{l}^1$- and $\mathbf{l}^2$-norm (2.8) yields nearly the same results on all data bases.

Haar scattering obtains comparable results to unsupervised deep learning algorithms when the graph geometry is unknown. For images on a known grid, Haar scattering produces an error ~20% larger.

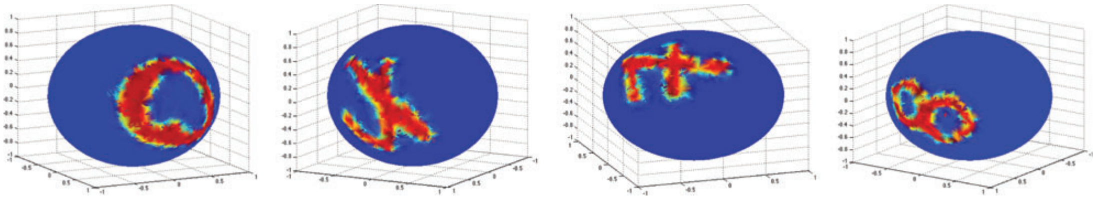

FIG. 8. Images of digits mapped on a sphere.

TABLE 6 *Percentage of errors for the classification of MNIST images rotated and sampled on a sphere* [7], *with a nearest neighbor classifier, a fully connected two layer neural network, a spectral network* [7] *and an unsupervised Haar scattering*

	Nearest neighbors	Fully connect.	Spectral net. [7]	Graph Haar scattering	Free Haar scattering
Small rotations	19	5.6	6	2.2	**1.6**
Large rotations	80	52	50	**47.7**	55.8

The bold values indicate the best performance.

The relative loss of accuracy when the graph is known means that the Haar architecture introduces a significant loss of model capacity, whereas in the unknown-graph case, the error is dominated by the estimation of the graph topology, where Haar scattering provides competitive results with unsupervised deep networks.

4.4 *Images on a graph over a sphere*

A data basis of irregularly sampled images on a sphere is provided in [7]. It is constructed by projecting the MNIST image digits on $d = 4096$ points randomly sampled on the 3D sphere, and by randomly rotating these images on the sphere. The random rotation is either uniformly distributed on the sphere or restricted with a smaller variance (small rotations) [7]. The digit ‘9’ is removed from the data set because it cannot be distinguished from a ‘6’ after rotation. Examples of sphere digits are shown in Fig. 8. This geometry of points on the sphere can be described by a graph which connects points having a sufficiently small distance on the sphere.

The classification algorithms introduced in [7] take advantage of the known distribution of points on the sphere, with a representation based on the graph Laplacian. Table 6 gives the results reported in [7], with a fully connected neural network, and with a spectral graph Laplacian network.

As opposed to these algorithms, the unsupervised graph Haar scattering algorithm does not use this geometric information and learns the graph information by pairing. Computations are performed on a scrambled set of signal values. Haar scattering transforms are calculated up to the maximum scale $2^J = d = 2^{12}$. A total of $T = 10$ connected dyadic partitions are estimated by unsupervised learning, and the classification is performed from $M = 10^3$ selected coefficients. Although the graph geometry is unknown, the graph Haar scattering reduces the error rate both for small and large 3D random rotations.

In this case, a free orthogonal Haar scattering has a smaller error rate than a graph Haar scattering for small rotations, but a larger error for large rotations. It illustrates the trade-off between the structural bias and the feature variance in the choice of the algorithms. For small rotation, the variability within classes is smaller, and a free scattering can take advantage of more degrees of freedom. For large rotations, the variance is too large and dominates the problem.

Two points of the sphere of radius 1 are considered to be connected if their geodesic distance is smaller than 0.1. With this convention, over the 4096 points, each point has on average 8 connected neighbors. The unsupervised Haar learning performs a hierarchical pairing of points on the sphere. For small and large rotations, the percentage of connected sets $V_{j,n}$ remains above 90% for $1 \leqslant j \leqslant 4$. This is computed over 70% of the points having a non-negligible energy. It shows that the multiscale geometry on the sphere is well estimated by hierarchical pairings.

5. Conclusion

We introduced an orthogonal Haar scattering, which is computed with a deep cascade of additions, subtractions and absolute values. The architecture preserves some important properties of unsupervised deep networks, while providing a simple model for their mathematical analysis. For signals defined on a graph, a Haar scattering iteratively computes orthogonal Haar wavelet transforms on the graph. It is invariant to local displacements of signal values on the graph. The unknown geometry of signals is estimated with an unsupervised learning algorithm. It minimizes the average total signal variation over dyadic partitions of graph vertices, with a polynomial complexity algorithm. Haar scattering classifications are numerically tested over image databases defined on uniform grids or irregular graphs, whose geometries are either known or estimated by unsupervised learning.

Supervised convolutional networks have led to considerable classification improvements compared with unsupervised convolutional network learning. An open issue is to understand how to optimize Haar pairings from supervised data, in order to improve results obtained from unsupervised data.

Acknowledgments

We would like to thank Ronald R. Coifman, Harrison Zhou and Tal Sarig for their comments and suggestions.

Funding

This work was supported by the ERC grant InvariantClass 320959.

References

1. Ankenman, J. I. (2014) Geometry and analysis of dual networks on questionnaires. *PhD thesis*, Yale University.
2. Arora, S., Bhaskara, A., Ge, R. & Ma, T. (2013) Provable bounds for learning some deep representations. Preprint arXiv:1310.6343.
3. Bengio, Y., Courville, A. & Vincent, P. (2013a) Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**, 1798–1828.
4. Bengio, Y., Courville, A. & Vincent, P. (2013b) Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**, 1798–1828.
5. Bengio, Y., Thibodeau-Laufer, E. & Yosinski, J. (2013) Deep generative stochastic networks trainable by backprop. Preprint arXiv:1306.1091.
6. Bruna, J. & Mallat, S. (2013) Invariant scattering convolution networks. *IEEE Trans. PAMI*, **35**, 1872–1886.

7. Bruna, J., Zaremba, W., Szlam, A. & LeCun, Y. (2014) Spectral networks and deep locally connected networks on graphs. *Conference Proceedings of International Conference on Learning Representations 2014*. (ICLR'14).

8. Chen, X., Cheng, X. & Mallat, S. (2014) Unsupervised deep Haar scattering on graphs. *Advances in Neural Information Processing Systems 27 (NIPS'14).* Curran Associates, Inc., pp. 1709–1717.

9. Chen, S., Cowan, C. F. & Grant, P. M. (1991) Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Netw.*, **2**, 302–309.

10. Coifman, R. & Gavish, M. (2011) Harmonic analysis of digital data bases. *Wavelets and Multiscale analysis*. Berlin, Heidelberg, New York: Springer, pp. 161–197.

11. Coifman, R. & Maggioni, M. (2006) Diffusion wavelets. *Appl. Comput. Harmonic Anal.*, **21**, 53–94.

12. Coifman, C., Meyer, Y. & Wickerhauser, M. (1992) Wavelet analysis and signal processing. *Wavelets and Their Applications* (Ruskai ed.). Boston: Jones and Bartlett, pp. 153–178.

13. Edmonds, J. (1965) Paths, trees, and flowers. *Can. J. Math.*, **17**, 449–467.

14. Erhan, D., Manzagol, P., Bengio, Y., Bengio, S. & Vincent, P. (2009) The difficulty of training deep architectures and the effect of unsupervised pre-training. *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS '09)*, pp. 153–160.

15. Gabow, H. N. (1976) An efficient implementation of Edmonds' algorithm for maximum matching on graphs. *J. ACM*, **23**, 221–234.

16. Gavish, M., Nadler, B. & Coifman, R. R. (2010) Multiscale Wavelets on Trees, Graphs and High Dimensional Data: Theory and Applications to Semi Supervised Learning. *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 367–374.

17. Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A. & Benjio, Y. (2013) Maxout networks. Preprint arXiv:1302.4389.

18. He, K., Zhang, X., Ren, S. & Sun, J. (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. Preprint arXiv:1502.01852.

19. Hinton, G., Deng, L., Yu, D., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Dahl, G. & Kingsbury, B. (2012) Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.*, **29**, 82–97.

20. Hinton, G., Osindero, S. & Teh, Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural Comput.*, **18**, 1527–1554.

21. Hinton, G. & Salakhutdinov, R. R. (2006) Reducing the dimensionality of data with neural networks. *Science*, **313**, 504–507.

22. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2012) Improving neural networks by preventing co-adaptation of feature detectors. Preprint arXiv:1207.0580.

23. Jia, Y., Huang, C. & Darrell, T. (2012) Beyond spatial pyramids: receptive field learning for pooled image features. *CVPR*, 2012 IEEE Conference on, Providence, RI, pp. 3370–3377.

24. Jones, P. W., Osipov, A. & Rokhlin, V. (2011) Randomized approximate nearest neighbors algorithm. *Proc. Natl. Acad. Sci.*, **108**, 15679–15686.

25. Labusch, K., Barth, E. & Martinetz, T. (2008) Simple method for highperformance digit recognition based on sparse coding. *IEEE TNN*, **19**, 1985–1989.

26. Le, Q., Sarlos, T. & Smola, A. (2013) Fastfood—approximating kernel expansions in loglinear time. *ICML. Proceedings of the 30th International Conference on Machine Learning (ICML'13)*, JMLR: W&CP, pp. 224–252.

27. LeCun, Y., Kavukvuoglu, K. & Farabet, C. (2010) Convolutional networks and applications in vision. *Proceedings of 2010 IEEE International Symposium on, Circuits and Systems (ISCAS),* pp. 253–256.

28. Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z. & Tu, Z. (2014) Deeply supervised nets. Preprint arXiv:1409.5185.

29. Lin, T.-H. & Kung, H.-T. (2014) Stable and efficient representation learning with nonnegativity constraints. *Proceedings of the 31st International Conference on Machine Learning (ICML'14)*, pp. 1323–1331. ICML.

30. Mallat, S. (2012) Group invariant scattering. *Commun. Pure Appl. Math.*, **65**, 1331–1398.

31. Mallat, S. (2016) Understanding deep convolutional networks. *Philosphical Trans. A*, **374**, 2065.

32. Mallat, S. & Waldspurger, I. (2013) Deep learning by scattering. Preprint arXiv:1306.5532.

33. Maurey, B. (1991) Some deviation inequalities. *Geom. Funct. Anal.*, **1**, 188–197.

34. Ngiam, J., Chen, Z., Chia, D., Koh, P. W., Le, Q. V. & Ng, A. Y. (2010) Tiled convolutional neural networks. *Adv. Neural Inf. Process. Syst.*, **23**, 1279–1287.

35. Oyallon, E. & Mallat, S. (2014) Deep roto-translation scattering for object classification. Preprint arXiv:1412.8659.

36. Pisier, G. (1985) *Probabilistic Methods in the Geometry of Banach Spaces*. Springer Lecture Notes in Math., vol. 1206. Springer: Berlin, Heidelberg. pp. 167–241.

37. Preis, R. (1999) *Linear Time $\frac{1}{2}$-Approximation Algorithm for Maximum Weighted Matching in General Graphs*, STACS99: 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, Springer: Berlin Heidelberg. pp. 259–269.

38. Rifai, S., Vincent, P., Muller, X., Glorot, X. & Bengio, Y. (2011) Contractive auto-encoders: explicit invariance during feature extraction. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 833–840.

39. Roux, N. L., Bengio, Y., Lamblin, P., Joliveau, M. & Kégl, B. (2008) Learning the 2-D topology of images. *Adv. Neural Inf. Process. Syst.*, **20**, 841–848.

40. Rustamov, R. & Guibas, L. (2013) *Wavelets on Graphs via Deep Learning*, Advances in Neural Information Processing Systems 26. ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger, Curran Associates, Inc. pp. 998–1006.

41. Salakhutdinov, R. (2015) Learning deep generative models. *Annu. Rev. Stat. Appl.*, **2**, 361–385.

42. Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A. & Vandergheynst, P. (2013) The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, **30**, 83–98.

43. Szlam, A. D., Maggioni, M., Coifman, R. & Bremer Jr, J. C. (2005) Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions. *Optics & Photonics 2005*. International Society for Optics and Photonics, pp. 59141D.

44. Taigman, Y., Yang, M., Ranzato, M. & Wolf, L. (2014) DeepFace: closing the gap to human-level performance in face verification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014*, pp. 1701–1708.

45. Yu, D. & Deng, L. (2011) Deep convex net: a scalable architecture for speech pattern classification. *Proceedings of INTERSPEECH*, pp. 2285–2288.

Appendix A. Proof of Theorem 3.1

Proof of Theorem 3.1. We derive from the definition of a scattering transform in equations (2.4, 2.5) in the text that

$$S_{j+1}x(n,2q) = S_jx(\pi_j(2n),q) + S_jx(\pi_j(2n+1),q) = \langle \bar{S}_jx(\cdot,q), 1_{V_{j+1,n}}\rangle,$$

$$S_{j+1}x(n,2q+1) = |S_jx(\pi_j(2n),q) - S_jx(\pi_j(2n+1),q)| = |\langle \bar{S}_jx(\cdot,q), \psi_{j+1,n}\rangle|,$$

where $V_{j+1,n} = V_{j,\pi_j(2n)} \cup V_{j,\pi_j(2n+1)}$. Define $\kappa = 2^{-j}q = \sum_{k=1}^m 2^{-j_k}$. Observe that

$$2^{j_{m+1}}(\kappa + 2^{-j_{m+1}}) = 2^{j_{m+1}}\kappa + 1 = 2(2^{j_{m+1}-1}\kappa) + 1,$$

thus $S_{j_{m+1}}x(n, 2^{j_{m+1}}(\kappa + 2^{-j_{m+1}}))$ is calculated from the coefficients $S_{j_{m+1}-1}x(n, 2^{j_{m+1}-1}\kappa)$ of the previous layer with

$$S_{j_{m+1}}x(n, 2^{j_{m+1}}(\kappa + 2^{-j_{m+1}})) = |\langle \bar{S}_{j_{m+1}-1}x(\cdot, 2^{j_{m+1}-1}\kappa), \psi_{j_{m+1},n}\rangle|. \tag{A.1}$$

Since $2^{j+1}\kappa = 2 \cdot 2^j \kappa$, the coefficient $S_{j_{m+1}-1}x(n, 2^{j_{m+1}-1}\kappa)$ is calculated from $S_{j_m}x(n, 2^{j_m}\kappa)$ by $(j_{m+1} - 1 - j_m)$ times additions, and thus

$$S_{j_{m+1}-1}x(n, 2^{j_{m+1}-1}\kappa) = \langle \bar{S}_{j_m}x(\cdot, 2^{j_m}\kappa), 1_{V_{j_{m+1}-1,n}} \rangle. \tag{A.2}$$

Combining equations (A.2) and (A.1) gives

$$S_{j_{m+1}}x(n, 2^{j_{m+1}}(\kappa + 2^{-j_{m+1}})) = |\langle \bar{S}_{j_m}x(\cdot, 2^{j_m}\kappa), \psi_{j_{m+1},n} \rangle|. \tag{A.3}$$

We go from the depth j_{m+1} to the depth $j \geqslant j_{m+1}$ by computing

$$S_j x(n, 2^j(\kappa + 2^{-j_{m+1}})) = \langle \bar{S}_{j_{m+1}}x(\cdot, 2^{j_{m+1}}(\kappa + 2^{-j_{m+1}})), 1_{V_{j,n}} \rangle.$$

Together with (A.3) it proves equation (3.9) of the proposition. The summation over $p, V_{j_{m+1},p} \subset V_{j,n}$ comes from the inner product $\langle 1_{V_{j_{m+1},p}}, 1_{V_{j,n}} \rangle$. This also proves that $\kappa + 2^{-j_{m+1}}$ is the index of a coefficient of order $m + 1$. □

Appendix B. Proof of Theorem 3.2

The theorem is proved by analyzing the concentration of the objective function around its expected value as the sample number N increases. We firstly introduce the Pisier and Maurey's version of the Gaussian concentration inequality for Lipschitz functions.

Proposition B.1 (Gaussian concentration for Lipschitz function [33,36]) Let $z_1, \dots z_m$ be i.i.d. $N(0, 1)$ random variables, and $f = f(z_1, \dots, z_m)$ a 1-Lipschitz function, then there exists $c_0 > 0$ so that

$$\mathbf{Pr}[f - \mathbb{E}f > t] < \exp\{-c_0 t^2\} \quad \text{and} \quad \mathbf{Pr}[f - \mathbb{E}f < -t] < \exp\{-c_0 t^2\}, \;\; \forall t > 0.$$

In the above proposition, the constant $c_0 = 2/\pi^2$ according to [36] and $\frac{1}{4}$ in [33].

To prove the theorem, recall that the pairing problem is computed by minimizing the $\mathbf{l}^1$-norm (3.10), which up to a normalization amounts to compute:

$$\pi^* = \arg\min_{\pi \in \Pi_d} F(\pi) \quad \text{with } F(\pi) = \frac{1}{N} \sum_{i=1}^{N} \sum_{(u,v)\in\pi} |x_i(u) - x_i(v)|, \tag{B.1}$$

where π is a pairing of d elements and we denote by Π_d the set of all possible such pairings.

The following lemma proves that $F(\pi)$ is a Lipschitz function of independent Gaussian random variables, with a Lipschitz constant equal to $\|\Sigma_d\|_{\text{op}}^{1/2}$, where $\|\Sigma_d\|_{\text{op}}$ is the operator norm of the covariance. We prove it on the normalized function $f = N^{1/2}d^{-1/2}F$.

Lemma B.1 Let $x_i = \Sigma_d^{1/2} z_i$ with $z_i = (z_i(1), \dots, z_i(d))^\top \sim \mathcal{N}(0, I_d)$ i.i.d. Given any pairing $\pi \in \Pi_d$, define

$$f(\{(z_i(v)\}_{1 \leqslant i \leqslant N, 1 \leqslant v \leqslant d}) = \frac{1}{\sqrt{dN}} \sum_{i=1}^{N} \sum_{(u,v)\in\pi} |x_i(u) - x_i(v)|,$$

then f is a Lipschitz function with constant $\sqrt{\|\Sigma_d\|_{\text{op}}}$, which does not depend on π.

Proof. With slight abuse of notation, denote by $v = \pi(u)$ if two nodes u and v are paired by π, then we have

$$\begin{aligned}\frac{\partial f}{\partial z_i(v')} &= \frac{1}{\sqrt{dN}} \sum_{(u,v)\in\pi} \mathrm{Sgn}(x_i(u) - x_i(v)) \frac{\partial}{\partial z_i(v')}(x_i(u) - x_i(v)) \\ &= \frac{1}{\sqrt{dN}} \sum_{u=1}^{d} \mathrm{Sgn}(x_i(u) - x_i(\pi(u))) \frac{\partial}{\partial z_i(v')} x_i(u) \\ &= \frac{1}{\sqrt{dN}} \sum_{u=1}^{d} \mathrm{Sgn}(x_i(u) - x_i(\pi(u)))(\Sigma_d^{1/2})_{u,v'} \\ &= \frac{1}{\sqrt{dN}} (\Sigma_d^{1/2} S_i)(v'),\end{aligned}$$

where $S_i := (\mathrm{Sgn}(x_i(u) - x_i(\pi(u))))_{u=1}^{d}$ is a vector of length d whose entries are ±1. Then

$$\|\Sigma_d^{1/2} S_i\| \leqslant \sqrt{\|\Sigma_d\|_{\mathrm{op}} d},$$

and it follows that

$$\begin{aligned}\|\nabla_z f\|^2 &= \sum_{i=1}^{N} \sum_{v'=1}^{d} \left| \frac{\partial f}{\partial z_i(v')} \right|^2 \\ &= \sum_{i=1}^{N} \frac{1}{dN} \|\Sigma_d^{1/2} S_i\|^2 \leqslant \|\Sigma_d\|_{\mathrm{op}}.\end{aligned}$$

□

Observe that the eigenvalues of Σ_d are the discrete Fourier transform coefficients of the periodic correlation function $\rho(u)$

$$\hat{\rho}(k) = \sum_{j=0}^{d-1} \rho(j) \exp\left\{-i2\pi \frac{jk}{d}\right\} = \sum_{j=0}^{d-1} \rho(j) \cos\left(2\pi \frac{jk}{d}\right), \quad k = 0, \ldots, d-1.$$

Observe that $\sum_{u=1}^{d} S_i(u) = 0$ for each i, that is, S_i is orthogonal to the eigenvector of $\hat{\rho}(0)$. So the Lipschitz constant $\sqrt{\|\Sigma_d\|_{\mathrm{op}}} = \sqrt{\max_k |\hat{\rho}(k)|}$ can be slightly improved to be $\sqrt{\max_{k>0} |\hat{\rho}(k)|}$.

Let us now prove the claim of Theorem 3.2. Since the pairing has a probability larger than $1 - \epsilon$ to be connected if $\mathbf{Pr}\left[\pi^* \notin \Pi_d^{(0)}\right] < \epsilon$, we need to show that under the inequality (3.12) the probability $\mathbf{Pr}\left[\pi^* \notin \Pi_d^{(0)}\right]$ is less than ϵ. Let us denote

$$\alpha_u = \sqrt{\frac{2}{\pi} \cdot 2(1 - \rho(u))}, \quad \text{and} \quad \bar{\alpha}_2 = \min_{2 \leqslant u \leqslant d/2} \alpha_u, \tag{B.2}$$

and define

$$C_\rho = \frac{c_0}{\|\Sigma_d\|_{\mathrm{op}}} \left(\frac{1}{2}(\bar{\alpha}_2 - \alpha_1)\right)^2. \tag{B.3}$$

Equation (3.12) can be rewritten as

$$C_\rho \frac{N}{d} > 3\log d - \log \epsilon. \tag{B.4}$$

As a result of Proposition B.1 and Lemma B.1, if $C = c_0/\|\Sigma_d\|_{\mathrm{op}} \cdot N/d$ then $\forall \pi \in \Pi_d$,

$$\mathbf{Pr}[F(\pi) - \mathbb{E}F(\pi) > \delta] < \exp\{-C\delta^2\} \quad \text{and} \quad \mathbf{Pr}[F(\pi) - \mathbb{E}F(\pi) < -\delta] < \exp\{-C\delta^2\}, \quad \forall \delta > 0.$$

Observe that

$$\Pi_d = \bigcup_{m=0}^{d/2} \Pi_d^{(m)},$$

where $\Pi_d^{(m)}$ are the set of pairings which have m non-neighbor pairs. $\Pi_d^{(0)}$ is the set of pairings which only pair connected nodes in the graph, and for the ring graph $\Pi_d^{(0)} = \{\pi_0^{(0)}, \pi_1^{(0)}\}$, two of which interlace. For any $\pi \in \Pi_d^{(m)}$, suppose that there are m_l pairs in π so that the distance between the two paired nodes is l, $m_1 = d/2 - m$, $m_2 + \cdots + m_{d/2} = m$.

Recalling the definition of α_k in Equation (B.2), we verify that

$$\mathbb{E}F(\pi) = \alpha_1\left(\frac{d}{2} - m\right) + \alpha_2 m_2 + \cdots + \alpha_{d/2} m_{d/2} \geqslant \alpha_1\left(\frac{d}{2} - m\right) + \bar{\alpha}_2 m$$

when $m \geqslant 1$, and

$$\mathbb{E}F(\pi_0^{(0)}) = \mathbb{E}F(\pi_1^{(0)}) = \alpha_1 \frac{d}{2}.$$

Thus when $m \geqslant 1$,

$$\mathbb{E}F(\pi) - \mathbb{E}F(\pi_0^{(0)}) \geqslant (\bar{\alpha}_2 - \alpha_1)m \quad \forall \pi \in \Pi_d^{(m)}.$$

Define

$$\delta_m = \tfrac{1}{2}(\bar{\alpha}_2 - \alpha_1)m, \quad m = 1, \ldots, d/2,$$

and we have that

$$\begin{aligned}\mathbf{Pr}[\pi^* \notin \Pi_d^{(0)}] &= \mathbf{Pr}\left[\exists \pi \in \bigcup_{m=1}^{d/2} \Pi_d^{(m)},\ F(\pi) < \min\{F(\pi_0^{(0)}), F(\pi_1^{(0)})\}\right] \\ &\leqslant \mathbf{Pr}\left[F(\pi_0^{(0)}) > \mathbb{E}F(\pi_0^{(0)}) + \delta_1\right] \\ &\quad + \mathbf{Pr}\left[F(\pi_0^{(0)}) < \mathbb{E}F(\pi_0^{(0)}) + \delta_1,\ \exists \pi \in \bigcup_{m=1}^{d/2} \Pi_d^{(m)},\ F(\pi) < F(\pi_0^{(0)})\right]\end{aligned}$$

$$
\begin{aligned}
&\leqslant \mathbf{Pr}[F(\pi_0^{(0)}) > \mathbb{E}F(\pi_0^{(0)}) + \delta_1] \\
&\quad + \mathbf{Pr}\left[\exists \pi \in \bigcup_{m=1}^{d/2} \Pi_d^{(m)},\ F(\pi) < \mathbb{E}F(\pi) - \delta_m\right] \quad \text{(by that } (\bar{\alpha}_2 - \alpha_1)m - \delta_1 \geqslant \delta_m) \\
&\leqslant \exp\{-C\delta_1^2\} + \sum_{m=1}^{d/2} |\Pi_d^{(m)}| \exp\{-C\delta_m^2\} \\
&= \exp\left\{-C_\rho \frac{N}{d}\right\} + \sum_{m=1}^{d/2} |\Pi_d^{(m)}| \exp\left\{-C_\rho \frac{N}{d} m^2\right\}, \qquad \text{(B.5)}
\end{aligned}
$$

where C_ρ is as in Equation (B.3).

One can verify the following upper bound for the cardinal number of $\Pi_d^{(m)}$:

$$
|\Pi_d^{(m)}| \leqslant \frac{d^{2m}}{(2m)!}.
$$

With the crude bound $(2m)! \geqslant 1$, the above inequality inserted in (B.5) gives

$$
\mathbf{Pr}[\pi^* \notin \Pi_d^{(0)}] \leqslant \exp\left\{-C_\rho \frac{N}{d}\right\} + \sum_{m=1}^{d/2} d^{2m} \exp\left\{-C_\rho \frac{N}{d} m^2\right\}. \qquad \text{(B.6)}
$$

If we keep the factor $(2m)!$, the upper bound for the summation over m can be improved to be

$$
d^2 \exp\{-C_\rho N/d\} \sum_{m=1}^{d/2} ((2m)!)^{-1} \leqslant c \cdot d^2 \exp\{-C_\rho N/d\},
$$

where $c = (e-1)/2$ is an absolute constant. By applying this in the final bound in the theorem, the constant in front of $\log d$ is 2 instead of 3. The constant of the theorem is not tight, while the $O(d \log d)$ is believed to be the tight order as d increases.

To proceed, define the function

$$
g(x) = -C_\rho \frac{N}{d} \cdot x^2 + (2 \log d) \cdot x, \quad 1 \leqslant x \leqslant \frac{d}{2},
$$

and observe that $\max_{1 \leqslant x \leqslant d/2} g(x) = g(1)$ whenever

$$
\frac{\log d}{C_\rho N/d} < 1,
$$

which holds as long as Equation (B.4) is satisfied. Thus we have

$$
\sum_{m=1}^{d/2} d^{2m} \exp\left\{-C_\rho \frac{N}{d} m^2\right\} \leqslant \sum_{m=1}^{d/2} d^2 \exp\left\{-C_\rho \frac{N}{d}\right\} = \frac{d^3}{2} \exp\left\{-C_\rho \frac{N}{d}\right\},
$$

then the inequality (B.6) becomes

$$\mathbf{Pr}[\pi^* \notin \varPi_d^{(0)}] \leqslant \left(\frac{d^3}{2}+1\right) \exp\left\{-C_\rho \frac{N}{d}\right\} \leqslant \exp\left\{-C_\rho \frac{N}{d} + 3\log d\right\}.$$

To have $\mathbf{Pr}[\pi^* \notin \varPi_d^{(0)}] < \epsilon$, a sufficient condition is therefore

$$\exp\left\{-C_\rho \frac{N}{d} + 3\log d\right\} < \epsilon, \tag{B.7}$$

which is reduced to Equation (B.4) and equivalently Equation (3.12).