DRAFT (latest content update on Jan. 30, 2014, by Jyri J. Kivinen). Published at https://jyrikivinen.github.io/home/, first on Nov. 1, 2023.

HIERARCHICAL DIRICHLET PROCESS HIDDEN MARKOV TREES FOR MULTISCALE IMAGE ANALYSIS

By Jyri J. Kivinen[1] and Erik B. Sudderth[2] and Michael I. Jordan[3]

University of Edinburgh[1] *Brown University*[2] *University of California, Berkeley*[3]

We develop hierarchical, nonparametric Bayesian models of multiscale image representations. Individual wavelet coefficients or features are marginally distributed as Dirichlet process (DP) mixtures, yielding the heavy-tailed marginals characteristic of natural images. The hidden assignments of features to clusters are then globally linked via a tree-structured graphical model. The resulting multiscale stochastic process automatically adapts to the varying complexity of different datasets, and captures global, highly non-Gaussian statistical properties of natural images. This hierarchical Dirichlet process hidden Markov tree (HDP-HMT) framework extends prior work on hidden Markov trees, local Gaussian scale mixtures, and HDP hidden Markov models. By truncating the potentially infinite set of hidden states, we develop Monte Carlo methods which exploit belief propagation for efficient learning from large datasets. Our results show that the HDP-HMT captures interesting structure in natural scenes, and leads to effective algorithms for image categorization and denoising. Moreover, by transferring statistics learned from a database of natural images, we demonstrate significant improvements in denoising highly distorted images over a baseline empirical Bayesian approach, which uses image statistics learned only from the noisy image.

1. Introduction.

The visual information in typical computational vision tasks exhibits highly structured, but complex variability. In most tasks ranging from low- to high-level vision, drawing reasonable inferences requires a priori knowledge on the regularities of the stimuli via statistical modelling, often done in a simplifying feature-based representation. In this paper we will consider representations which are multiscale, commonly used in computational vision. Multiresolution representations have also natural ties to tree-structured graphical models, which benefit from efficient exact estimation methods [18]. One highly successful model adopting these ideas is the Hidden Markov Tree (HMT) - model [4]. A HMT consists of a tree of discrete-valued latent state-assignment variables, which generate pyramidally organizable image features. The dependency-structure in the model is extremely simple: Conditioned on a hidden state assignment variable, the associated observation is independent from the rest of the the tree. Despite such simplicity in model structure, it enables modeling of complex marginal distributions, and joint dependencies: As each value of a hidden state assignment variable describes the

Keywords and phrases: hidden Markov trees, hierarchical Dirichlet processes, Bayesian nonparametrics, wavelets, multiscale models, image denoising, visual scene categorization, transfer learning

random choice of emission distribution for the associated observation, observations are marginally represented as mixture models. Given an appropriate number of components, this enables effective modeling of the heavy-tailed marginal distributions observed in natural imagery. The Markov-dependencies between the hidden states then enable capturing higher-order dependencies between the observations.

Nonetheless, the HMT has a problem of great practical significance: it doesn't include a mechanism for the allocation of an appropriate number of components to the various scales in the model. This problem has been tried to solve previously by using information criterion-based model selection, but the hierarchical structure of the model makes the approach computationally very challenging and problematic for many real-life applications [23]. In this article, we present a nonparametric Bayesian modeling framework, Hierarchical Dirichlet Process Hidden Markov Tree (HDP-HMT), which provides an alternative to the model selection, and extends the HMT in several important ways. Most importantly, the number of hidden states in the models are unbounded, and are determined in a data-driven and -adaptive way. It is accomplished by associating each state transition mixture a Dirichlet Process Mixture, and coupling them in learning with the Hierarchical Dirichlet Process (HDP)-framework [26].

We propose Monte Carlo learning algorithms to estimate the posterior distributions of the models' parameters, which allow learning from large and potentially noisy image databases, and develop effective algorithms for scene categorization and image denoising. Scene categorization is directly useful in applications such as image annotation and retrieval [29]. The global identity and structure of natural scenes also provides important contextual cues for the detection and recognition of objects [27, 28]. In addition to being an important application as itself, image denoising is commonly used in assessing image model performance.

When making predictions about clean image data, most of the denoising methods rely solely on statistics learned from the noisy image at hand, such as in empirical Bayesian approaches [19], and engineering-based approaches [6]. At high noise levels there can be very little image information available, and regularities to learn about clean images to be used for making predictions. Such statistics could be learned a priori from a database of clean images, and there are some recent methods adopting these ideas [5, 21], but they cannot learn additional statistics from the noisy image in a principled fashion. Accomplishing such is straightforward in our case, as we show in section 5 where we propose a framework for denoising based on transfer learning, applicable also for other restoration tasks.

We begin in Sec. 2 by reviewing previous models for multiscale representations of natural images based on Gaussian scale mixtures, and tree–structured latent variable models. The section also describes nonparametric Baysian methods, and adapts them for unstructured image representations. We then integrate these research themes in the HDP-HMT, and develop Monte Carlo methods for learning from training images in Sec. 3, and evaluate its suitability as a model for natural images and scenes in scene categorization (Sec. 4), and denoising (Sec. 5). In categorization, we also demonstrate the importance of capturing dependencies between image features by comparing HDP-HMT to its “bag-of-feature” version, and that of the feature representation used. In denoising, we also show the significant benefits of using statistics from both clean images and the noisy image in estimation, by comparing denoising with the HDP-HMT using an empirical Bayesian and a transfer learning-based algorithm. In both of these image analysis tasks, we compare the performance of the HDP-HMT against state-of-the-art methods.

2. Multiscale Analysis of Natural Images.

Images of natural scenes typically contain large, homogeneously textured regions, as well as localized intensity changes caused by occlusion boundaries. Their statistics are thus most simply characterized in representations which are jointly localized in spatial position and frequency [24, 28]. These observations have motivated the use of many wavelet-based approaches to image modeling. In these methods, images are first decomposed using a linear basis into a pyramid of wavelet coefficients, whose statistics are then modeled.

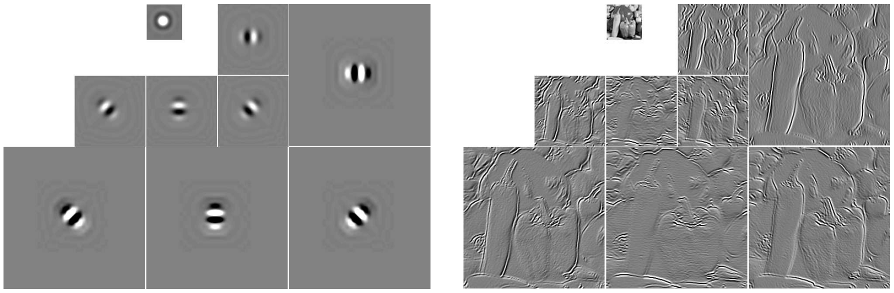
There are numerous efforts to capture the regularities existing in the decomposed images, ranging from local to global statistical analysis of the transform coefficients [24]. In the following, we describe briefly the multiscale decompositions, and the models used to describe their statistics, most relevant to our work.

In image restoration tasks, in which images are analyzed and processed in a transformed space, the underlying decomposition needs to be invertible. This is the case for our image denoising algorithms developed later. In scene categorization, there is no such restriction. In the following, in addition to invertible wavelet transforms, we will also discuss non-invertible SIFT-descriptor pyramids, which we show in section 4 lead into improved scene categorization results over wavelets.

2.1. *Wavelet Representations.*

Wavelet transforms decompose images at multiple scales by recursively filtering with a scaled, band-pass kernel function. This invertible linear transform produces a set of low-pass *scaling* coefficients, and higher frequency *detail* coefficients, organized in a pyramid (see Figure 1). Although the *fixed* set of basis functions (consisting of translated, dilated, and rotated versions of a common kernel) have not been designed a particular class of images in mind, they are similar to those obtained using sparse coding methods aiming to find a linear basis optimal (in terms of maximal sparseness and statistical independence of transform coefficients) for natural images [8, 17][1].

[1] as well as to the receptive fields of simple cells in primary visual cortex



(a) Basis functions illustration

(b) Natural image responses (high-pass residual not shown)

FIG 1. *Illustration of basis functions for 3^{rd}-order steerable pyramids, and the transform coefficients of a natural image (peppers). The top-left basis function corresponds to a low-pass filter/scaling function, whereas the other basis functions illustrated are oriented band-pass filters. Each of the illustrated basis function images have been obtained by reconstructing a steerable pyramid with a single non-zero coefficient positioned at the center of the corresponding subband, after which they have been cropped and resized for visualization purposes.*

Despite these similarities between wavelet transforms, there are differences of significant importance for modeling and analysis purposes. For example, transforms which are not translational invariant may be problematic for many image analysis tasks: while the critically sampled orthogonal wavelet transforms approximately decorrelate or *whiten* natural images, and thus lead to effective compression algorithms, their lack of translational invariance may lead to instability and highly visible aliasing artifacts (arising from the critical sampling) in the presence of noise. *Steerable pyramids* address these issues via an overcomplete basis, or frame, optimized for increased orientation selectivity [22]. While the statistics of such non-orthogonal transformations are more complex, they are advantageous for image analysis [11, 19]. These representations are also used in the wavelet-based image denoising methods we propose in Section 5, as in several other leading wavelet-based denoising methods (such as in [15, 19]).

2.2. *SIFT-descriptor pyramids.* The SIFT-transform [14] is a feature-extraction method which has been used extensively in recent years in various visual recognition tasks. It describes image information with histograms of oriented gradients within a neighborhood, which is split into a grid of analysis-subregions, around of point of interest. Within each of these subregions, oriented gradients are computed for each pixel, and a subregion-specific histogram with direction-quantized bins is built based on them. The full SIFT-descriptor is a 128-dimensional vector, consisting of 8-bin histograms (each representing gradient strength within that orientation regime), one for each of the 4×4 subregions.

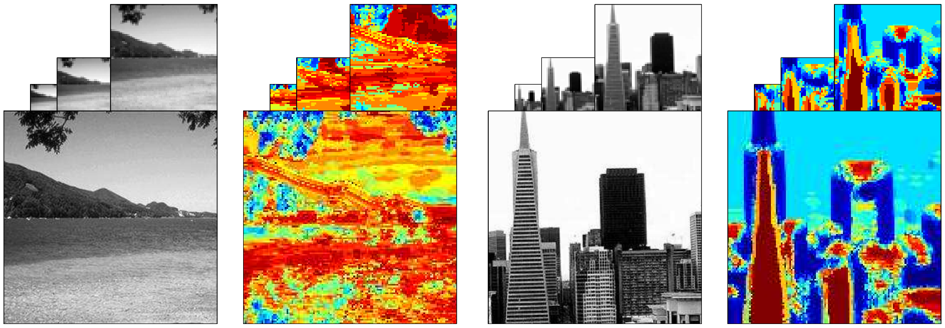


FIG 2. *Natural scene images and corresponding SIFT-codeword pyramids.*

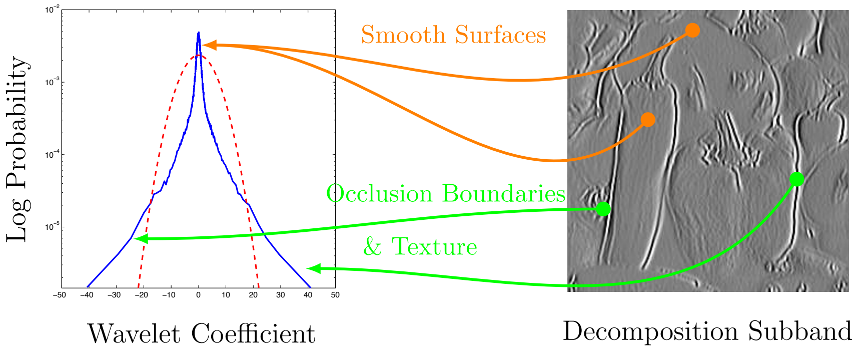


FIG 3. *Empirical histogram (blue solid line) of a natural image subband coefficient values with maximum likelihood Gaussian fit overlaid (red dashed line).*

Previous comparative studies have shown that for scene categorization tasks, best performance is attained by computing features on a dense, regular grid [1, 7], rather than at sparse interest points [14, 20]. The intuitive explanation for this phenomenon is that the presence of open, textureless regions is highly indicative of certain scene categories [16, 28]. In this work, we extract features from overlapping patches spaced on a grid. To provide further discriminative power, we also rescale the extraction window and extract dense features at multiple coarser scales. To reduce the dimensionality of the extracted features, they are vector-quantized into a dictionary of visual words. Figure 2 shows example natural scene images (resized into multiple scales), and corresponding SIFT-codeword pyramids, in which colors encode visual words from a 1000-element dictionary, obtained using K-means clustering.

2.3. *Models for Statistics of Individual Coefficients.* Wavelet coefficients typically have highly *kurtotic* marginal distributions, with “heavy tails” indicating that extreme values occur frequently compared to Gaussian distributions. A class of distributions that can provide good matches to the heavy-tailed marginals of individual coefficients x_i, are mixture distributions. One widely used *continuous* mixture is the *Gaussian scale mixture*, which models x_i as the product of two independent variables:

$$x_i = \sqrt{v_i} u_i \qquad v_i \geq 0,\ u_i \sim \mathcal{N}(0, \Lambda) \tag{1}$$

Marginalizing the scalar multiplier v_i mixes Gaussians of varying scales:

$$p(x_i) = \int \mathcal{N}\left(x_i; 0, \Lambda\right) dG(\Lambda) \tag{2}$$

A variety of continuous mixing distributions $G(\Lambda)$ provide good models of wavelet statistics [30]. In some cases, however, even simple two–component mixtures are effective:

$$x_i \sim \pi \mathcal{N}(0, \Lambda_0) + (1 - \pi)\mathcal{N}(0, \Lambda_1) \tag{3}$$

Here, π is the probability that x_i is drawn from an “outlier” component with large variance Λ_0, and Λ_1 is smaller to capture the many near–zero coefficients. Such *discrete* mixtures have important computational advantages, and have been successfully used for image denoising [3].

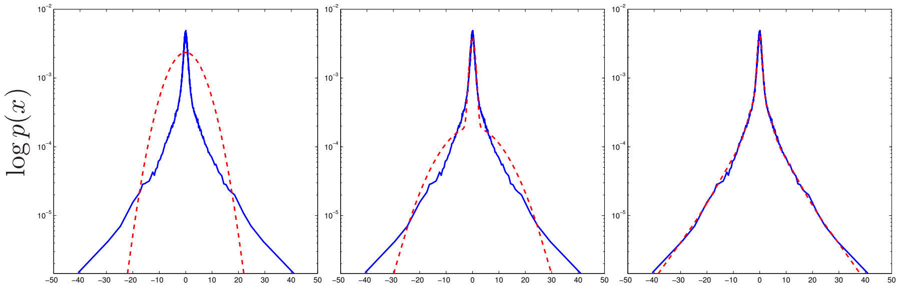
An example of using this density for modeling individual wavelet subband coefficients is shown in Figure 4. Even though the binary mixture provides a closer fit than a single Gaussian, many more components are needed to build a highly accurate model. However, choosing too many components leads to overfitting, and poor generalization in testing. Also for different datasets (such as for different images, or for a larger set of images) different complexities may be appropriate.

A practical solution is to use a Dirichlet process (DP) mixture of Gaussians, which assumes a *discrete* mixture of unbounded number of Gaussians, and uses a regularizing prior on the mixture weights π_k. Through this prior, which is part of the core machinery built in this paper, the appropriate dimensionality is determined in a data-driven fashion. The marginal distribution for a wavelet coefficient x_i under this model can be written as follows:

$$p(x_i) = \sum_{k=1}^{\infty} \pi_k f\left(x_i \mid \theta_k\right) = \sum_{k=1}^{\infty} \pi_k \mathcal{N}\left(x_i; 0, \Lambda_k\right) \tag{4}$$

The associated mixing distribution is *discrete*, and is defined as follows:

$$G(\Lambda) = \sum_{k=1}^{\infty} \pi_k \delta(\Lambda - \Lambda_k) \tag{5}$$



$\Delta H/H = 0.041$ $\quad$ 0.0025 $\quad$ 0.00073

FIG 4. *Empirical histograms (blue solid lines) of a natural image (Barbara) subband coefficient values with best fitting instances of three different models (red dashed lines) overlaid. Models from left to right: Gaussian, mixture of two Gaussians, Dirichlet process mixture of Gaussians. Below each plot is the relative entropy between the empirical histogram (using 500 variable-size bins) and the fitted model, as a fraction of the histogram entropy.*

where π_k denotes the mixing proportion for component k, associated with parameters Λ_k. The stick-breaking prior, which regularizes the infinite mixture, denoted by $\pi \sim \mathrm{GEM}(\gamma)$, defines the mixture weights using beta random variables:

$$\pi_k = \pi'_k \prod_{\ell=1}^{k-1} (1 - \pi'_\ell) \qquad\qquad \pi'_\ell \sim \mathrm{Beta}(1, \gamma) \tag{6}$$

The construction of the countably infinite set of mixture weights π can be seen as breaking proportions of a stick. It is started by breaking a random proportion given by $\mathrm{Beta}(1, \gamma)$ of a stick of length one. The consecutive breaks are done to the part of the stick, that's remaining after previous breaks, each with a random proportion again sampled from $\mathrm{Beta}(1, \gamma)$. The component parameters are independently sampled as $\Lambda_k \sim H$, where H denotes a prior over component parameters. Throughout this paper, we use conjugate inverse–Wishart H priors for zero–mean Gaussian component distributions of the continuous wavelet coefficients, and Dirichlet H for multinomial models of vector quantized SIFT descriptors.

In the generative process under the model, observations are then generated by first choosing a mixture component z_i with probabilities given by the mixture weights, and then drawing an observation x_i from the corresponding emission distribution $f(\theta_{z_i})$. For the continuous wavelet coefficients, this distribution $f(\theta_{z_i}) = \mathcal{N}(0, \Lambda_{z_i})$, whereas for discrete SIFT-codewords such as those shown in Figure 5, $f(\theta_{z_i}) = \mathrm{Multinomial}(\theta_{z_i})$.

When modeling multiscale data, it is often useful to separate data into groups, and yet allow the groups to be linked - to share statistical strength. One flexible framework for sharing mixture components, among *groups* of related data, is the *hierarchical Dirichlet process* (HDP) [26]. In this framework, group-specific DP

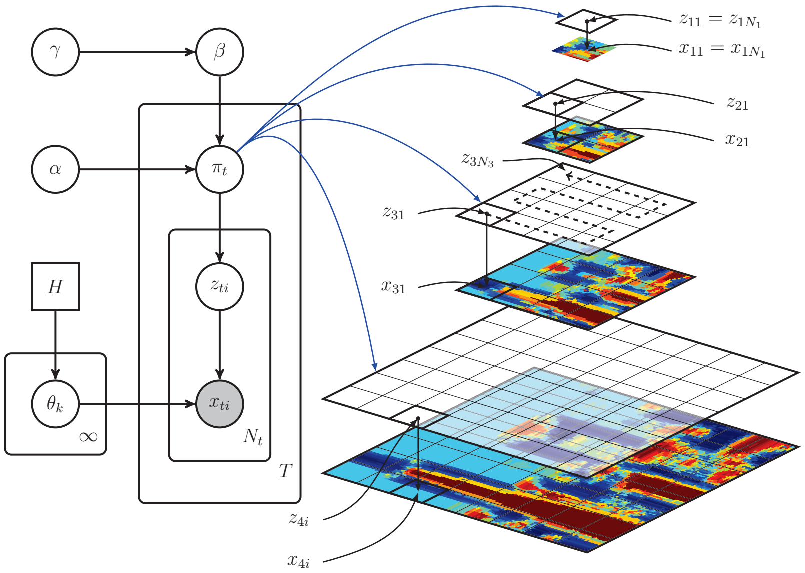


FIG 5. *Bag-of-Words HDP for a SIFT-descriptor pyramid. Each of the pyramid scales correspond to different groups (indexed by $t \in [1, \ldots, 4]$), and are each modeled using DP-mixtures of multinomials, which share mixture components θ_k via the HDP-framework.*

mixture models are coupled in learning by a global DP mixture with infinite set of shared mixture components θ, and their global mixing proportions β. For observations x_{ji} within a group j,

$$p(x_{ji} \mid \pi, \theta_1, \theta_2, \ldots) = \sum_{k=1}^{\infty} \pi_j(k) f(x_{ji} \mid \theta_k) \tag{7}$$

The sharing of mixture components is accomplished using the following generative process: Global mixture weights for an infinite set of shared mixture components $\{\theta_k\}$ are first sampled as in eq. (6). Each of the J groups (see Fig. 5) then reuses these same components in different proportions $\pi_j = (\pi_{j1}, \pi_{j2}, \ldots)$:

$$\pi_j \sim \mathrm{DP}(\alpha, \beta) \qquad \beta \sim \mathrm{GEM}(\gamma) \tag{8}$$

By defining β to be a discrete probability measure (samples from a DP are discrete with probability one), component sharing is ensured with high probability. Here, β determines the mean frequency of each component, while α controls the variability of component weights across groups [26]. Fixing these parameters, observations are then independently sampled as follows:

$$z_{ji} \sim \pi_j \qquad x_{ji} \,|\, z_{ji} \sim f\left(\theta_{z_{ji}}\right) \tag{9}$$

Rather than strictly constraining the number of latent states, the HDP's stick–breaking prior places a softer bias towards the simplest models which explain observed data. As we demonstrate in Sec. 4, this leads to rich models whose complexity grows as additional data is observed.

2.4. *Hidden Markov Trees for Global Statistics.* Although natural images often lead to uncorrelated wavelet coefficients, they retain important non–Gaussian dependencies[2]. In particular, large magnitude coefficients tend to *cluster* at nearby spatial locations, and *persist* across multiple scales [4, 30]. In overcomplete representations such as steerable pyramids, there are also significant dependencies between orientations. We can see these dependencies over orientation, scale, and position in the (gray-scale value coded) detail coefficients of the 'Peppers'-image shown in Figure 8, and from the 'bow-tie'-shapes in the conditional histograms of a wavelet coefficient conditioned on its adjacent wavelet coefficient, shown in Figure 11 (rows labeled 'Images'). These properties of wavelet-decomposed images are utilized in various image models. One of the most effective wavelet-based image denoising algorithms employs *local* Gaussian scale mixtures relating each wavelet coefficient *only* to its nearest neighbors in location and scale [19]. In this article, we instead develop a *global* graphical model of multiscale image decompositions.

The scale–recursive operations underlying wavelet decompositions suggest models defined on *Markov trees* [31]. For images, these graphical models associate detail coefficient x_{ti} with a single coarser scale *parent* $x_{\mathrm{Pa}(ti)}$, and four finer scale *children* $\{x_{tj} \mid tj \in \mathrm{Ch}(ti)\}$. Tree–structured Gaussian random fields have been used to capture correlations among wavelet coefficients [31], and to model the latent multipliers underlying a global Gaussian scale mixture [30]. In the stochastic process defined by the latter model, wavelet coefficient vectors are marginally distributed as infinite Gaussian Scale Mixtures. There are then two random processes determining how these observations evolve in the tree: A premultiplier MAR process captures self-reinforcing dependencies while a white noise process controls correlation structure among wavelet coefficients, which are then generated via a nonlinearity. Although the underlying representation is parsimonious, it suffers in expressiveness as the authors fix the order of the multiplier process and consider only fixed parametric forms of nonlinearity in the multiscale process.

Alternatively, the discrete mixture of eq. (3) has been generalized to define a binary *hidden Markov tree* (HMT) [4]. In HMTs, the mixture component z_{ti} generating detail coefficient x_{ti} is influenced by the corresponding parent coefficient:

$$z_{ti} \,|\, z_{\mathrm{Pa}(ti)} \sim \pi_{z_{\mathrm{Pa}(ti)}} \qquad x_{ti} \,|\, z_{ti} \sim \mathcal{N}(0, \Lambda_{z_{ti}}) \tag{10}$$

[2] this is the case also with bases obtained with the sparse coding methods mentioned earlier

As before, detail coefficient x_{ti} may be generated via *states* z_{ti} of low or high variance. However, by associating each parent state k with a different *transition distribution* π_k, HMTs also capture dependencies among nearby coefficients. For example, cascade effects of large magnitude coefficients can be obtained by associating transitions from high-variance parent state to high-variance child state likely.

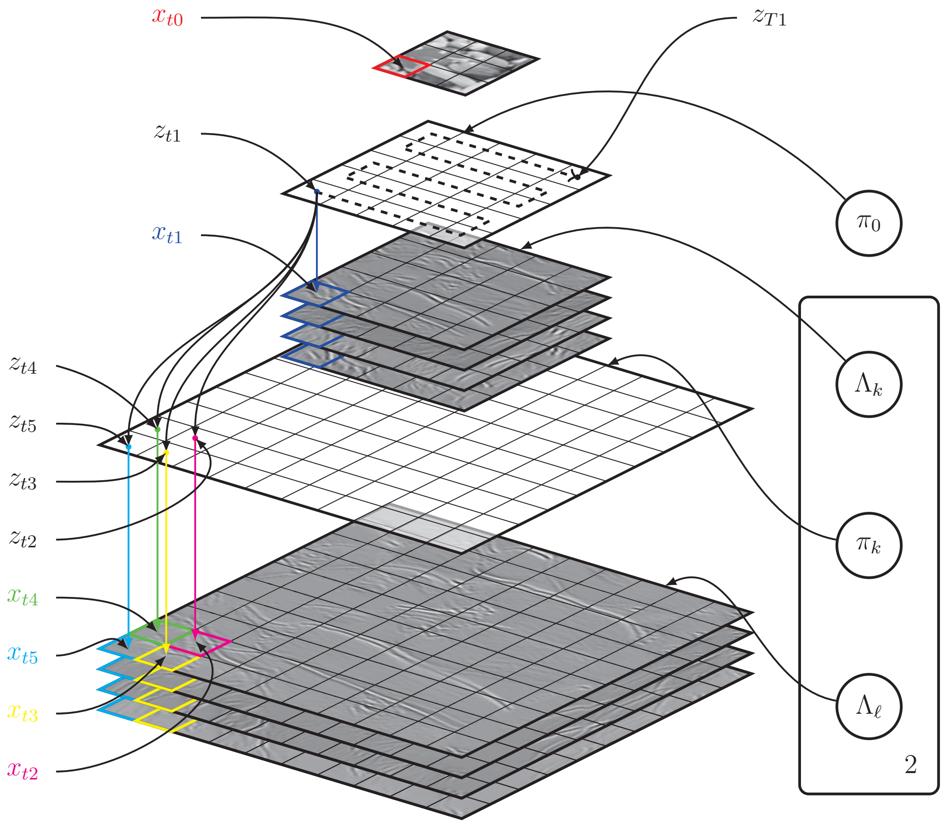


FIG 6. *An extended binary HMT model for the wavelet coefficients of a two-scale, four-orientation wavelet transform. The hidden discrete states z_{ti} generate vectors of observed features x_{ti} spanning over multiple orientations, and a spatial region (in the above illustration $t = 1$). The states at neighboring locations and scales are coupled by state transition distributions π, which in addition to emission distribution covariances are chosen independently for each scale.*

Although the HMT originally defined separate graphical models for each orientation subband, states may alternatively generate vectors of wavelet coefficients [23], as illustrated in Figures 6 & 7. Each of these wavelet vectors x_{ti} span over multiple (4 in Fig. 6 & 7) orientations, and for example also a spatial region as illustrated in

Figure 6 - what is only required is that the emissions have equal dimension, and the hidden states which generate them can be arranged as a tree. For a single decomposed image, there is a forest of T trees (or a single tree, if $T = 1$), equal to the number of coarsest scale observation vectors[3]. Dependencies among these multi-dimensional observations are then better captured by higher–order discrete models. To do this, one must select an appropriate *number* of hidden states K, as well as the pattern used to *share* states among different coefficient vectors. For example, the *hierarchical image probability* (HIP) model [23] shares parameters within each scale, and optimizes K via a minimum description length (MDL) criterion. This MDL-based model selection may however lead to combinatorial problems requiring greedy approximations, and the asymptotic justifications of MDL are poorly suited to small datasets. In the following section, we propose an alternative non-parametric approach which *learns* such model structures from training images, in a data-driven and -adaptive way. Furthermore, this approach enables efficient and natural mechanisms for reusing statistics learned from clean images in image denoising.

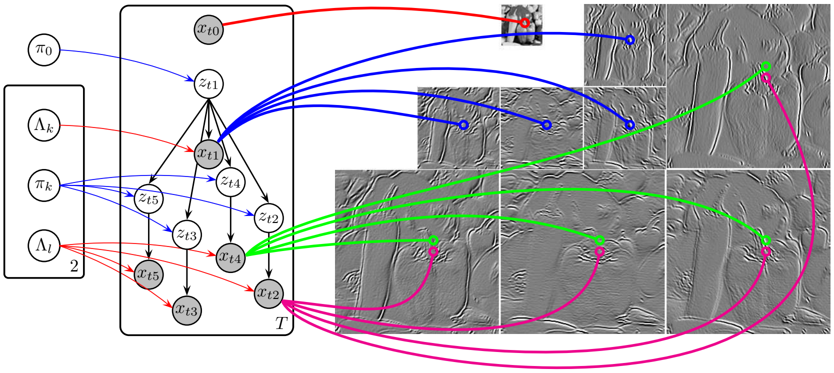


FIG 7. *Alternative representation of the extended binary HMT model for the wavelet coefficients of a two-scale, four-orientation wavelet transform shown in Figure 6.*

3. Hierarchical Dirichlet Process Hidden Markov Trees.

Hierarchical Dirichlet processes have been previously used to define an HDP-HMM which learns the structure of a countably infinite hidden Markov chain from training data [26]. In this section, we develop an *HDP hidden Markov tree* (HDP-HMT) which captures the global statistics of wavelet decompositions or locally extracted image features.

[3] For some decompositions, the number of observed scaling coefficients x_{t0} differs from T, but it doesn't affect the modeling as they are not part of the generative process.

In this section we also develop two Monte Carlo methods for estimating the posterior distributions of HDP-HMT parameters from training images. The collapsed Gibbs sampler employs Rao–Blackwellization to marginalize the underlying infinitely many infinite–dimensional transition distributions π_k^d, and emission distribution parameters θ_k; extending the direct assignment sampler by Teh et al.[26].

While the collapsed Gibbs sampler resamples individual assignments of features to hidden states, the truncated Gibbs sampler, on the other hand, resamples *jointly* entire trees of state assignments. This is made possible by considering truncated representations of the HDP, and results in more efficient blocked sampling algorithms, allowing learning from large datasets [12]. These truncations also provide a mechanism for balancing computational efficiency and representational accuracy, while maintaining a nonparametric model.

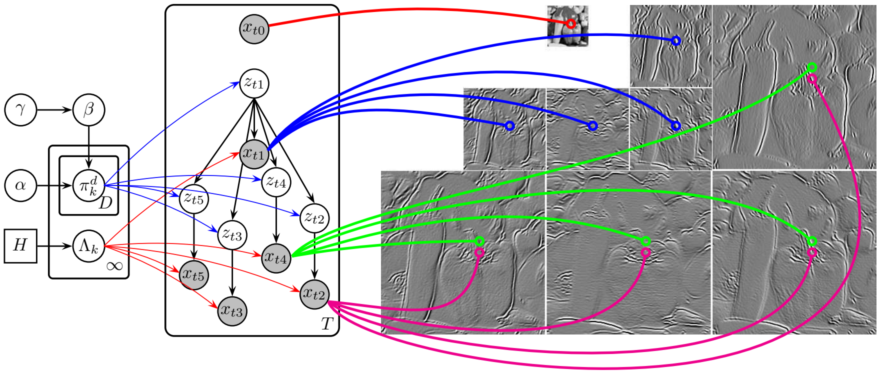


FIG 8. *An HDP-HMT for the coefficients of two-scale, four-orientation wavelet transform. The hidden discrete states z_{ti} generate observed detail coefficients x_{ti}, emitted by four-dimensional zero-mean Gaussian distributions, with covariance $\Lambda_{z_{ti}}$. The states at neighboring locations and scales are coupled by child position – dependent transition distributions π_k^d. A global measure β is used to couple these transitions when learning, encouraging reuse of hidden states.*

3.1. *Statistical Model.* Consider a hidden Markov tree, as in Fig. 8, with a countably infinite state space $z_{ti} \in \{1, 2, \ldots\}$. Each value k of the current state indexes a different transition distribution $\pi_k^d = (\pi_{k1}^d, \pi_{k2}^d, \ldots)$ over child states in different directions d. We couple these transitions via a shared DP prior:

$$\pi_k^d \sim \mathrm{DP}(\alpha, \beta) \qquad\qquad \beta \sim \mathrm{GEM}(\gamma) \tag{11}$$

The simplest approach ties all four children of each parent to follow the same transition distribution [11]. However, as reported in [12], we have found that allowing a distinct transition distribution π_k^d for each of the four child directions d more accurately models the asymmetries present in natural images. Given these infinite transition distributions, visual features are generated via the following coarse–to–fine recursion:

$$z_{ti} \mid z_{\mathrm{Pa}(ti)} \sim \pi^{d_{ti}}_{z_{\mathrm{Pa}(ti)}} \qquad x_{ti} \mid z_{ti} \sim F(\theta_{z_{ti}}) \tag{12}$$

By defining β to be a *discrete* probability measure, we ensure with high probability that a common set of child states are reachable from each parent state [26].

Analogously to the standard HDP of Fig. 5, this hierarchical construction encourages reuse of states when learning. However, the group associated with each observation is now dynamically determined by the state of its parent node, rather than being fixed *a priori*. This allows the HDP-HMT to learn complex patterns characteristic of multiscale observation sequences, and avoids the need to specify a fixed scheme for sharing states among observations. Furthermore, by defining a prior on infinite models, the HDP-HMT avoids the model selection issues considered by previous applications of Markov trees [23] and topic–based visual scene models [1, 7, 20].

Let us know consider a situation, in which observations w_{ti} are contaminated with zero-mean Gaussian noise of known variance Σ_n, so that $w_{ti} \sim \mathcal{N}(x_{ti}, \Sigma_n)$, where x_{ti} is a latent clean coefficient vector. To properly deal with such scenario, we augment the basic HDP-HMT with a set of unobserved clean coefficients x_{ti}, as illustrated in Figure 9. As the statistics of noise can now be separated from that of the signal, estimation of unobserved clean coefficients can now use also statistics learned from sets of clean images. Figure 10 illustrates a graphical model for a further HDP-HMT extension which also generates an observed database of clean images. In the following, learning algorithms are developed for the basic HDP-HMT shown in Figure 8. See Appendix D for learning algorithms for use with noise-contaminated observations.

3.2. *Learning by Collapsed Gibbs Sampling.* To learn the posterior distributions of the basic HDP-HMT parameters, the proposed Gibbs sampler alternates between sampling assignments z_{ti} to hidden states and global transition probabilities β, as summarized in Algorithm 1. Given fixed values for these variables, the state–specific transition distributions π_k and emission-distribution parameters θ_k can be marginalized in closed form. Such Rao-Blackwellization is guaranteed to reduce the variance of Monte Carlo estimators [25].

In the first first stage of the algorithm, assignments of features to clusters are resampled. In contrast with standard HDP models [26], the HDP-HMT *dynamically* regroups observed features as parent states indexing the groups are resampled. In sampling, we consider candidate states z_{ti} corresponding to every state which is

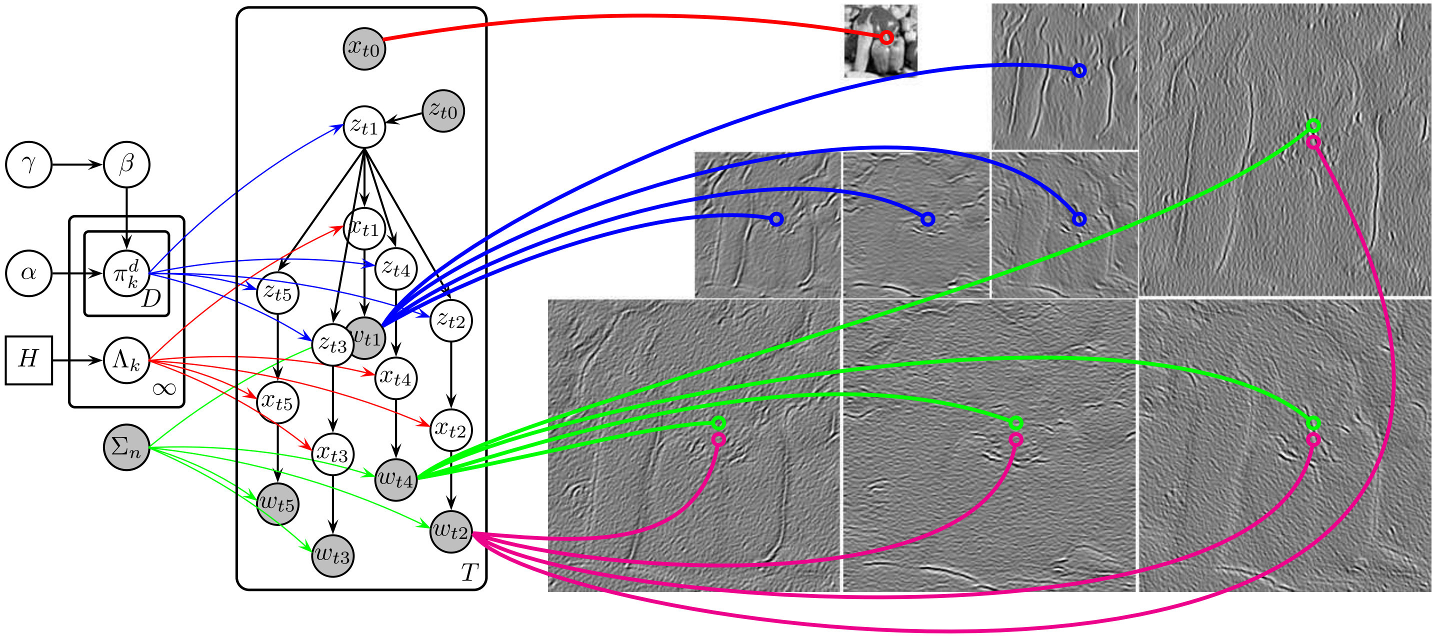


FIG 9. *A two-scaled HDP-HMT in which hidden discrete states z_{ti} generate noise–free features x_{ti}. Observations w_{ti} are corrupted by additive Gaussian noise with covariance Σ_n. States at neighboring locations and scales are linked by direction-dependent transition distributions π_k^d. A global measure β couples these transitions when learning, encouraging reuse of hidden states.*

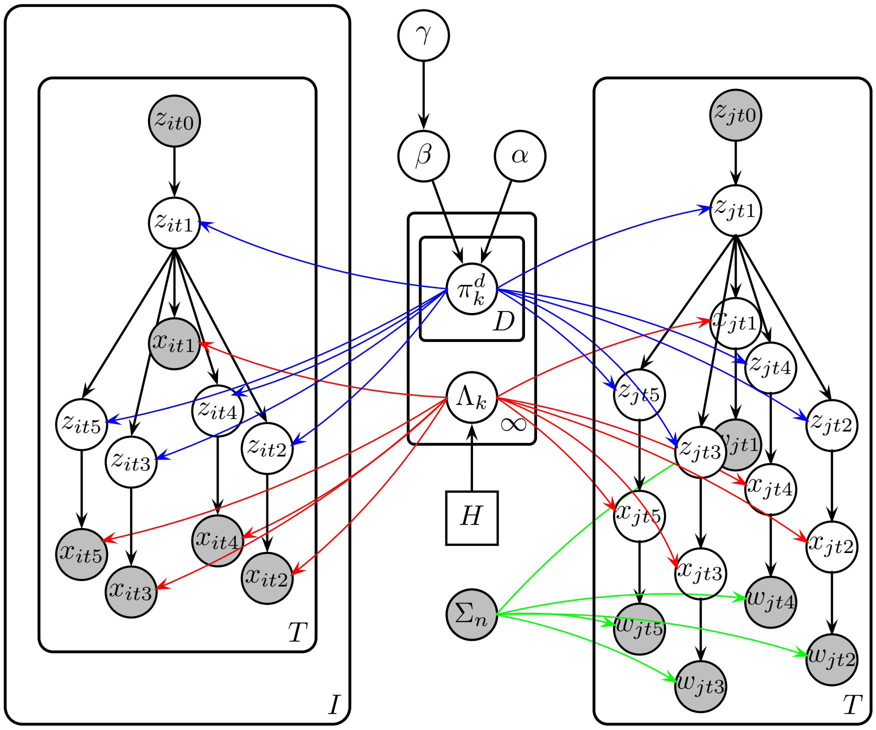


FIG 10. *A graphical representation of a two-scale model for transfer denoising. The leftmost trees generate observed wavelet coefficients of clean data. The rightmost tree generates observed noisy coefficients of an image to be denoised. In denoising, we wish to find the posterior means of the unobserved clean coefficients* x_{jt}.

```
Given current state of global mixture weights $\beta$, and assignments of features to states $\mathbf{z}$:

  1. Sample a random permutation of integers indexing images and their nodes $ti$.
  2. Sample state assignments $z_{ti}$, for each node $ti$.
       (a) Remove feature from cached class-specific statistics:
             • update hidden state transition counts:
                 $n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) \leftarrow n^{d(ti)}(z_{\mathrm{Pa}(ti)}, z_{ti}) - 1;$
                 $n^{d(ti)}_{\backslash ti}(z_{ti}, z_{\mathrm{Ch}(ti)}) \leftarrow n^{d(ti)}_{\backslash ti}(z_{ti}, z_{\mathrm{Ch}(ti)}) - 1$
             • update the inverse-Wishart posterior hyperparameters $\{\kappa_{z_{ti}}, \nu_{z_{ti}}, \Delta_{z_{ti}}\}$
               to account removal of $x_{ti}$
       (b) Determine predictive likelihoods for each candidate class:
             $p(x_{ti} \,|\, \mathbf{z}, \mathbf{x}_{\backslash ti}, H) = \text{Student-t}_{\nu-d+1}\left(x_{ti}; 0, \nu\Delta\frac{\kappa+1}{\kappa(\nu-d+1)}\right)$
       (c) Sample new class assignment from the following multinomial:
             $p(z_{ti} \,|\, \mathbf{z}_{\backslash ti}, \beta, \mathbf{x}) \propto p(z_{ti} \,|\, \mathbf{z}_{\backslash ti}, \beta) p(x_{ti} \,|\, \mathbf{z}, \mathbf{x}_{\backslash ti}, H)$
           where $p(z_{ti} \,|\, \mathbf{z}_{\backslash ti}, \beta)$ has different forms depending on the node position:
           leaf nodes:
             $\left(\frac{n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) + \alpha\beta_{z_{ti}}}{n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, \cdot) + \alpha}\right)$
           other nodes:
             $\left(\frac{n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) + \alpha\beta_{z_{ti}}}{n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, \cdot) + \alpha}\right) \left(\frac{n^{d(ti)}_{\backslash ti}(z_{ti}, z_{tl}) + \alpha\beta_{z_{tl}} + \delta(z_{\mathrm{Pa}(ti)}, z_{ti})\delta(z_{ti}, z_{tl})}{n^{d(ti)}_{\backslash ti}(z_{ti}, \cdot) + \alpha + \delta(z_{\mathrm{Pa}(ti)}, z_{ti})}\right)$
             $\prod_{tj \in \mathrm{Ch}(ti) \backslash tl} \left(\frac{n^{d(tj)}_{\backslash ti}(z_{ti}, z_{tj}) + \alpha\beta_{z_{tj}}}{n^{d(ti)}_{\backslash ti}(z_{ti}, \cdot) + \alpha}\right)$
             where $tl \in \mathrm{Ch}(ti)$, and $d(tl) = d(ti)$.
       (d) Add feature to cached class-specific statistics:
             • update hidden state transition counts:
                 $n^{d(ti)}(z_{\mathrm{Pa}(ti)}, z_{ti}) \leftarrow n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) + 1;$
                 $n^{d(ti)}(z_{ti}, z_{\mathrm{Ch}(ti)}) \leftarrow n^{d(ti)}(z_{ti}, z_{\mathrm{Ch}(ti)}) + 1$
             • update the inverse-Wishart posterior hyperparameters $\{\kappa_{z_{ti}}, \nu_{z_{ti}}, \Delta_{z_{ti}}\}$
               to account addition of $x_{ti}$
  3. Global mixture weights $\beta$ can be resampled via an auxiliary variable method [26].
```

Algorithm 1: Collapsed Gibbs sampling algorithm for an HDP-HMT model with zero-mean multivariate Normal emission distributions, and inverse-Wishart priors on their covariance matrices. We illustrate the resampling of assignments z_{ti} to hidden states and global transition probabilities β. A full iteration of the Gibbs sampler applies the feature assignment updates to all images in random order. For efficiency, we cache and recursively update the state-specific statistics.

used at least once elsewhere in the tree, as well as a potential *new* state. This predictive rule allows HDP-HMTs to determine state space cardinality in a data–driven fashion. The number of states grows when new clusters are added, and shrinks, when all observations currently assigned to a cluster are removed.

In the second stage of the algorithm, the global transition probabilities β are resampled. Given fixed assignments $\mathbf{z} = \{z_{ti}\}$ of coefficients to hidden states, β can be resampled using auxiliary variable methods [26].

We provide high–level derivations for the sampling updates underlying the Algorithm 1 in Appendix A. Although this direct assignment sampler desirably employs *Rao–Blackwellization* [25] to avoid explicitly sampling some latent variables, it can exhibit slow mixing because it only updates one hidden state assignment at a time. In addition, the recursive updates of sufficient statistics needed to marginalize parameters can be costly when performed after every feature reassignment. To address these issues, we propose in the following sections an alternative *truncated* sampler, which uses finite approximations of the Dirichlet process to allow joint resampling of entire trees of state assignments.

3.3. *Truncated Representations.* There are two basic methods for producing finite approximations to DP models. The first truncates the stick–breaking construction of eq. (6) by setting $\beta'_L = 1$ for some sufficiently large L. In this article, we instead use alternative, "weak limit" approximations which sample β from a K–dimensional finite Dirichlet distribution with symmetric parameters:

$$\beta = (\beta_1, \ldots, \beta_K) \sim \mathcal{D}(\gamma/K, \ldots, \gamma/K) \tag{13}$$

We then take β as the weight vector for a finite, K–component mixture model with parameters $\theta_k \sim H$ as before. It can then be shown that the predictions based on this finite model converge in distribution to those of a corresponding Dirichlet process $\mathrm{DP}(\gamma, H)$ as $K \to \infty$ [9, 10]. A similar finite approximation exists for the HDP [26] of Fig. 5, in which β is sampled as in eq. (13) and group–specific mixture weights are drawn according to

$$\pi_t = (\pi_{t1}, \ldots, \pi_{tK}) \sim \mathcal{D}(\alpha\beta_1, \ldots, \alpha\beta_K) \tag{14}$$

The next subsection extends this approximation to the HDP-HMT to develop a truncated Gibbs sampling algorithm.

It is important to note that the truncation level K is *not* taken to be the number of mixture components observed in the data, but rather a loose upper bound on that number. Indeed, as we show in Sec. 4, the Dirichlet priors of eqs. (13, 14) cause the sampler to explain observations via a dynamically chosen *subset* of the pool of available mixture states. Theoretical results are available which characterize the mixture size needed for accurate posterior approximations [10].

3.4. *Blocked Gibbs Sampler for Truncated Representations.* Given a truncation level K, our truncated Gibbs sampler for the basic HDP-HMT, summarized in Algorithm 2, alternates between blocked resampling of trees of state assignments $\mathbf{z}_t$, global mixture weights β, and state-specific model parameters and transition distributions $\{\theta_k, \pi_k\}_{k=1}^{K}$.

We begin by conditioning on each state's transition distribution π_k and observation distribution θ_k. Given these fixed parameters, the joint distribution of the hidden states $\mathbf{z}_t$ and observations $\mathbf{x}_t$ can be represented by a forest of tree–structured, directed graphical models (see Fig. 8). For such models, the belief propagation (or sum–product) algorithm can be used to efficiently resample *all* of the latent assignments in closed form [18, 31].

Messages are first passed from the leaves to the root of each tree to collect summary statistics, which can also be used to evaluate the marginal likelihood $p\big(\mathbf{x}_t \mid \{\pi_k, \theta_k\}_{k=1}^{K}\big)$ in closed form. A top–down recursion is then used to resample each node z_{ti} given its parent $z_{\mathrm{Pa}(ti)}$. The computational cost of resampling the assignments for N observed features is thus $\mathcal{O}(NK^2)$.

In the second stage of the truncated sampler, we condition on the assignments z of observations to hidden states. It is then straightforward to resample the observation distributions θ_k by aggregating statistics of the observations $\{x_{ti} \mid z_{ti} = k\}$ assigned to each state [25, 26]. To resample state–specific transition distributions π_k^d, we first count the number $n^d(k, \ell)$ of transitions from parent state k to child state ℓ, in direction d, instantiated by $\mathbf{z}$. The posterior is then Dirichlet with combined counts from the hidden state transitions and pseudo transitions from the prior. In our implementation, parameter sampling is done very efficiently by caching sufficient statistics of the state-specific parameters.

Finally, the global mixture weights β can be resampled via an auxiliary variable method [26]. The truncation level K can be either chosen larger than the number of expected states to ensure a good approximation to the underlying HDP, or set smaller to control computational complexity with large datasets. We provide high–level derivations for the sampling updates underlying the Algorithm 2 in Appendix C.

4. Categorization of Natural Scenes. This section develops natural scene models using the HDP-HMT framework, and evaluates their effectiveness in capturing natural scene statistics, and in categorizing images of new environments.

We begin by describing the proposed scene models, and how they are learned from training images. To provide understanding of the properties and capabilities of the models, we then visualize the statistics of scenes they have captured, and look at their categorization performance. Throughout the experiments, we compare the HDP-HMT against a bag-of-features HDP (HDP-BOW), a non-parametric

```
Given current state of global mixture weights $\beta$, state-specific model parameters and
transition distributions $\{\theta_k, \pi^d_k\}^K_{k=1}$, and assignments of features to states $z_{ti}$ for the
currently sampled image:
  1. Remove the statistics of previous assignments of features to classes:
       • update hidden state transition counts:
           $n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) \leftarrow n^{d(ti)}(z_{\mathrm{Pa}(ti)}, z_{ti}) - 1;$
           $n^{d(ti)}_{\backslash ti}(z_{ti}, z_{\mathrm{Ch}(ti)}) \leftarrow n^{d(ti)}_{\backslash ti}(z_{ti}, z_{\mathrm{Ch}(ti)}) - 1$
       • update the posterior hyperparameters to account removal of $x_{ti}$
  2. Sample state assignments $z_{ti}$ with belief propagation:
       (a) Compute messages upwards from the leaves up to the roots:
           for leaf nodes:
             $m^{tj}_{ti}(z_{tj}) \propto \sum_{z_{ti}} \pi^{d(ti)}_{z_{tj}}(z_{ti}) p(x_{ti}|\theta_{z_{ti}})$
           for non-leaf nodes:
             $m^{tj}_{ti}(z_{tj}) \propto \sum_{z_{ti}} \pi^{d(ti)}_{z_{tj}}(z_{ti}) p(x_{ti}|\theta_{z_{ti}}) \prod_{tk \in \mathrm{Ch}(ti)} m^{ti}_{tk}(z_{ti})$
       (b) Sample hidden states while traversing downwards:
           for non-leaf nodes:
             $p(z_{ti} \mid z_{\mathrm{Pa}(ti)}, \mathbf{x}_{t\cdot}, \pi, \theta) \propto \pi^{d(ti)}_{z_{\mathrm{Pa}(ti)}}(z_{ti}) p(x_{ti} \mid \theta_{z_{ti}}) \prod_{tj \in \mathrm{Ch}(ti)} m^{ti}_{tj}(z_{ti})$
           for leaf nodes:
             $p(z_{ti} \mid z_{\mathrm{Pa}(ti)}, \mathbf{x}_{t\cdot}, \pi, \theta) \propto \pi^{d(ti)}_{z_{\mathrm{Pa}(ti)}}(z_{ti}) p(x_{ti} \mid \theta_{z_{ti}})$
  3. Add the statistics of new assignments of features $x_{ti}$ to classes $z_{ti}$:
       • update hidden state transition counts:
           $n^{d(ti)}(z_{\mathrm{Pa}(ti)}, z_{ti}) \leftarrow n^{d(ti)}_{\backslash ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) + 1;$
           $n^{d(ti)}(z_{ti}, z_{\mathrm{Ch}(ti)}) \leftarrow n^{d(ti)}(z_{ti}, z_{\mathrm{Ch}(ti)}) + 1$
       • update the posterior hyperparameters to account addition of $x_{ti}$
  4. Sample model parameters $\{\theta_k, \pi_k\}^K_{k=1}$:
       (a) Sample direction-specific transition distributions by drawing a random
           Dirichlet-vector
             $\pi^d_k \sim \mathcal{D}\left(n^d(k,1) + \alpha\beta_1, \ldots, n^d(k,K) + \alpha\beta_K\right)$
       (b) Sample $\theta_k$'s by drawing a random vector from
             $p(\theta_k|\mathbf{x}, \mathbf{z}, H) \propto p(\theta_k \mid H) \prod_{j:z_j=k} p(x_j \mid \theta_k)$
  5. Global mixture weights $\beta$ can be resampled via an auxiliary variable method [26].
```

Algorithm 2: Blocked Gibbs sampler for learning truncated HDP-HMTs from training images. We illustrate the blocked resampling of trees of state assignments $\mathbf{z}_t$, global mixture weights β, and state-specific model parameters and transition distributions $\{\theta_k, \pi_k\}^K_{k=1}$. A full iteration of the sampler applies the updates to all images in random order. For efficiency, we cache the state-specific statistics and recursively update them when assignment changes.

Bayesian extension of the model proposed by [7] for scene categorization, which ignores global structure. This is to demonstrate the importance of capturing spatial feature dependencies in addition to local feature appearance. In the task of scene categorization (of eight natural scene categories provided by Oliva and Torralba [16]) we also compare the discriminative power of the multiscale oriented edge responses of steerable pyramids, and a discrete vocabulary of vector quantized SIFT descriptors.

4.1. *Hierarchical Nonparametric Scene Models.* The HDP-HMT models for scenes extend the basic model illustrated in figure 8, so that the states are shared across a database of natural images of a scene category instead of a single image, and in the SIFT-domain models latent states generate SIFT-codewords instead of wavelet coefficient vectors. The observations are separated into T sets, equal to the number of coarsest-scale observations, each generated by a quadtree of hidden variables. The hidden variables z_{t1} generating the observations x_{t1} are generated by special root states z_{t0}. We chose the root states heuristically, by assigning them into 32 different values, in a grid of 8x4 segments (results by completing the forest into a tree with hidden variables were similar in 8 scenes categorization). As the coarsest observation scale was of size 16x16, there were 2x4 trees beneath each root state segment.

The HDP-BOW, used as a baseline model, associates a group to each scale in the pyramidally organized data, and thus the observed features are drawn from scale-specific infinite Dirichlet Process mixtures, whose components are shared across scales via the HDP framework.

We used both wavelet-domain and SIFT-domain features to train the scene models. The wavelet-domain features were extracted from 128×128 grayscale images, using 4–scale steerable pyramids, with 6 and 8 orientations (sp5 and spf7, respectively), and the associated low-pass and high-pass residual bands were discarded. The SIFT descriptors were extracted on a dense grid from 256×256 grayscale images, at four resolutions produced by dyadic rescaling of analysis window size. We then used K–means clustering to create two 1000–element codebooks from $8 \cdot 5000$, and $15 \cdot 5000$ randomly chosen features in training images, from 8 and 15 natural scene categories, respectively.

4.2. *Visualization of Learned Scene Statistics.* In Fig. 11, we illustrate wavelet coefficient histograms [30] computed from grayscale images in two categories, “coast” and “tallbuilding.” We compare this raw data to coefficients simulated from the HDP-HMT, and the bag–of–words (HDP-BOW) models. The simulation for the models used the states of their respective Markov chains at iteration number 200 in the truncated Gibbs samplers using 100 training images.

We can see from the figure, that the HDP-HMT models the non–Gaussian “bow

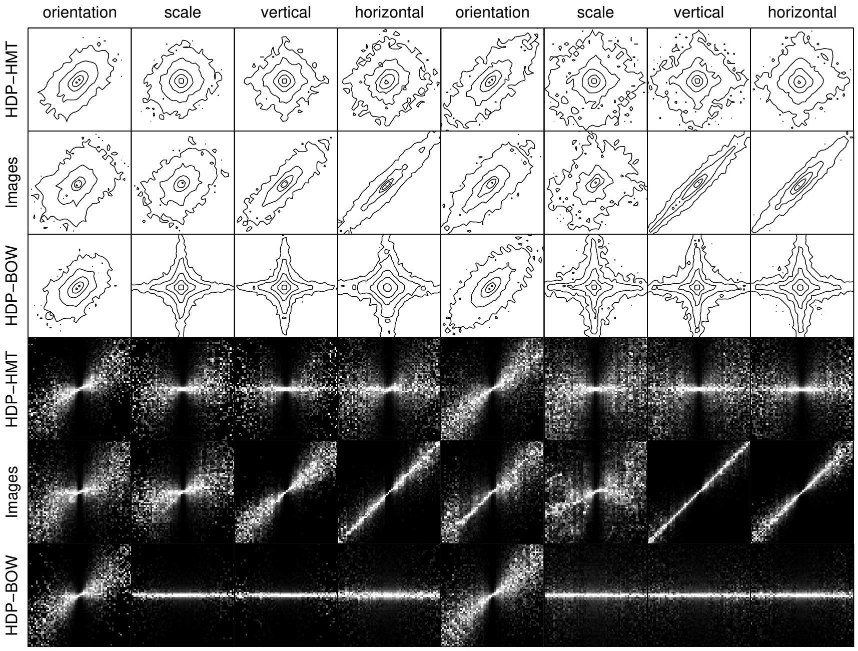


FIG 11. *Pairwise histograms of steerable pyramid detail coefficients computed from two* 128×128 *images. Columns 1–4 are computed from a "coast" image, while columns 5–8 are computed from a "tallbuilding" image. Rows 2 & 5 are computed from the observed images, while rows 3 & 6 and 1 & 4 summarize samples from the bag-of-features HDP-BOW and the HDP-HMT models, respectively. As in [30], we visualize log-contours of joint distributions (top) as well as normalized conditional distributions (bottom).*

tie" shapes of wavelet histograms, and also accurately capture the complex orientation and scale relationships exhibited by steerable pyramids. However, it underestimates the strong positive correlations between horizontally and vertically adjacent coefficients at horizontal and vertical finest scale bands, respectively. This is probably caused by the Markov tree boundaries which separate some pairs of finer scale coefficients [31].

In contrast, the HDP-BOW captures only the correlations between neighboring orientations, which are also well modeled with the HDP-HMT. The relations among the log-contours of raw and simulated data are also captured with both models, in the sense that the contours for "tallbuilding" images are more elongated than those for the "coast" images, which contain less high-frequency content. The vertically layered structure of large-scale environmental scenes [28] can be seen in the clear dominance of the horizontal band over the vertical band in the "coast" image histograms, and is captured by both models.

The inability of the HDP-BOW to capture scale and location correlations is also evident in the much less coherent maximum a posteriori (MAP) assignments of features to topics for test images in Fig. 12. The MAP assignments for the HDP-HMT, which were computed efficiently via the max-product algorithm, reveal that it more effectively models the dependencies between features, and interestingly even restores structure in the regions of the “tallbuilding” image corrupted by aliasing artifacts.

We also looked at samples from the models given the MAP assignments in the wavelet-domain. The samples we obtained by combining observed scaling coefficients with sampled detail coefficients and inverting the transform. These images shown in Figure 13 further verify the better capability of the HDP-HMT to capture spatial relationships. In our experiments we also found that the posterior distributions for the emission parameters were much tighter constrained for the tree model.

To further illustrate the nonparametric properties of the truncated model, we trained models for two categories with varying numbers of training images. During sampling, we collected 100 samples of the number of states, after allowing the Markov chain to burn-in for 100 iterations. Figure 14 shows the posterior mean of the number of hidden states, as a function of the number of wavelet–domain training images. As expected, the complexity of this nonparametric model grows as the number of training images increases, adapting *automatically* to the complexity of the data. Visual analysis of the Figure 14 indicates that in this experiment, the truncation limit did not limit the expressiveness of the models.

4.3. *Scene Categorization Results.* For our scene categorization experiments, we trained category-specific hierarchical nonparametric Bayesian models using 200 images for training and the rest were used for testing. For the HDP-HMT, we classified test images as the category which assigned the highest marginal likelihood to test features. These likelihoods can be efficiently computed in closed form with a single, coarse-to-fine belief propagation (BP) recursion [4, 31], as derived in B.1.

The categorization performance results obtained with HDP-HMT and HDP-BOW on the gray-scale eight category dataset [16] are summarized in Table 1, where average categorization performances are also shown for the “natural” and “man-made” subsets of the scene categories [16]. For the confusion matrices, see Fig. 21 in Appendix E. We can see that using the “stronger” local feature representation of the SIFT descriptors leads to significant improvements. Furthermore, results with the HDP-HMT model are better overall, demonstrating the benefits of coupling local features with global spatial models. In SIFT-domain, the HDP-HMT performs also better than the current leading approach [2], with average categorization accuracy of 86.5% against 84.7%.

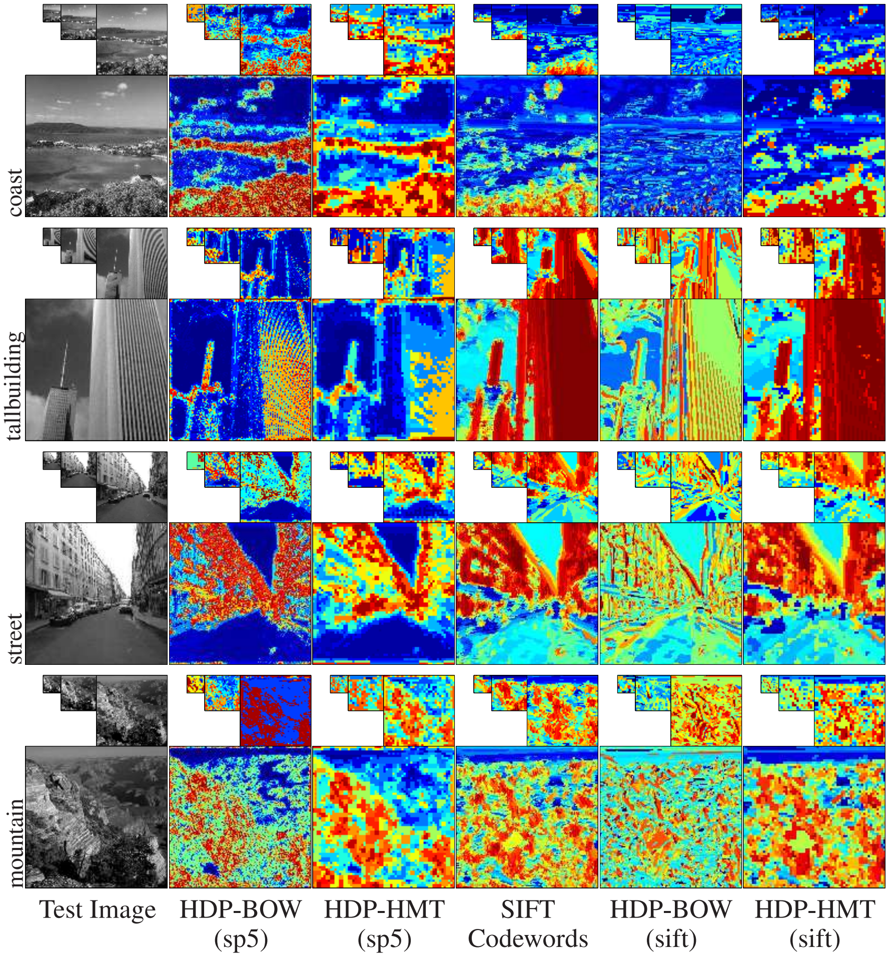


FIG 12. *Maximum a posteriori classification of the joint configuration of hidden states for a test image. In the wavelet domain, states are sorted based on the determinant of the covariance matrix of corresponding emission distribution. In the SIFT domain, states are sorted based on a trimmed posterior mean of corresponding emission distribution (in dominant orientation sorted space).*

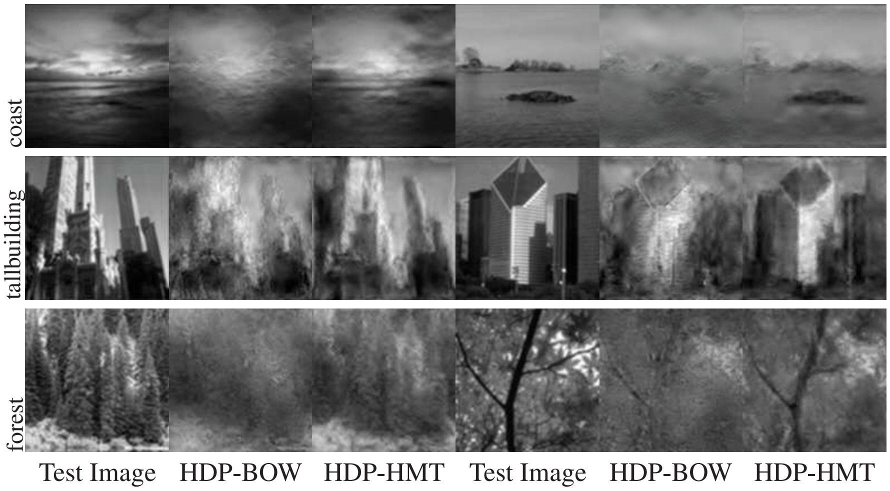
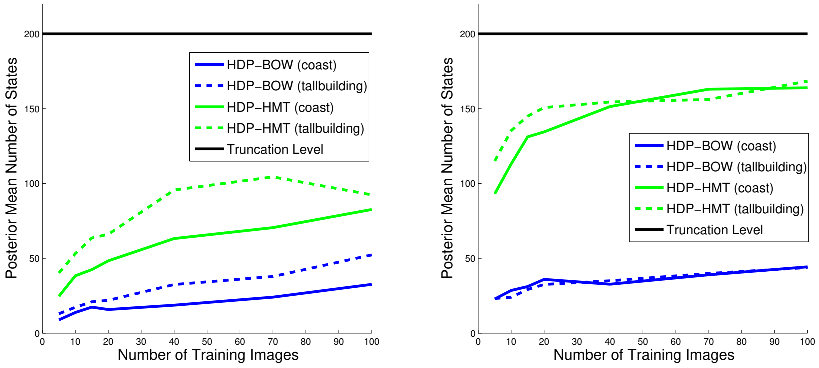


FIG 13. *Sampling scenes given maximum a posteriori assignments of the joint configuration of hidden states for a test image in a wavelet-domain (sp5). The scaling coefficients of the novel image were set as those of the test image.*



sp5 SIFT

FIG 14. *Number of states used by the HDP-BOW and HDP-HMT models as a function of the number of training images, for the "coast" and "tallbuilding" scene categories, in wavelet (sp5) and SIFT domains. The truncation limit was set to 200 for the models.*

	Wavelet (sfp7)		SIFT	
Man-made [16]	82.9	85.4	86.4	89.7
Natural [16]	78.6	83.5	85.7	87.7
Eight [16]	75.3	80.7	82.4	86.5
Thirteen [7]			75.9	81.8
Fifteen [13]			69.7	77.1
	HDP-BOW	HDP-HMT	HDP-BOW	HDP-HMT

TABLE 1

Average scene categorization results, determined as the mean of the diagonal entries of the corresponding confusion matrix.

We did a similar comparison with the gray-scale thirteen [7] and gray-scale fifteen [13] scene category datasets, but only in SIFT-domain, as clearly better performance was obtained in that domain already with the eight categories. Results from the comparison are summarized in Table 1. For the confusion matrices, see Figure 22 in Appendix E. As in the previous categorization experiment, the HDP-HMT obtains better overall performance, although significantly outperforms the HDP-BOW on the fifteen category dataset. On the thirteen category dataset, the HDP-HMT outperforms also the dataset authors with average categorization performance of 81.8% against 65.2%, which is also outperformed by the HDP-BOW (75.9%). However, on the fifteen category dataset, our average categorization rate of 77.1% is slightly less than that of the leading approach by the dataset authors 81.4%. However, our classifier does not need to keep around any training data in classification (as opposed to the methods of [1, 13, 16]), and the tree structure allows us to use fast belief propagation methods to efficiently compute likelihoods.

5. Image Denoising with HDP-HMTs. In this section, we use the HDP-HMT to restore images corrupted by additive white Gaussian noise, a standard task for evaluating image model effectiveness. We propose two denoising methods using the HDP-HMT framework, both applying conventional wavelet-based denoising methodology by denoising detail coefficients of the wavelet-transformed noisy image. In an empirical Bayesian approach, model parameters are estimated from the noisy image itself. In a transfer denoising approach, parameter estimation reuses statistics from a model trained on a database of clean images. We show in our experiments that the transfer denoising approach leads at high noise levels to more robust predictions than the empirical Bayesian approach, comparable to state-of-the-art methods.

5.1. *Empirical Bayesian Denoising.* Our overall learning algorithm for Empirical Bayesian denoising, summarized in Algorithm 4, first estimates parameters of the HDP-HMT shown in Figure 9 from the observed noisy image. We begin this learning by running a blocked Gibbs sampler summarized in Algorithm 5 on the noisy wavelet tree. The sampler extends Algorithm 2 to also resample noisy coefficients. Derivations for the updates are provided in Appendix C.3. After "burn–in", we collect samples $\theta^{(s)} = \{\pi_k^{(s)}, \Lambda_k^{(s)}\}_{k=1}^{K_s}$ from the parameters' posterior distribution. Note that each sample s instantiates a *different* number of states K_s.

As shown in Appendix D.1, given $\theta^{(s)}$, the conditional mean of w_{ti} equals

$$\mathbb{E}\big[x_{ti} \mid \mathbf{w}, \theta^{(s)}\big] = \sum_{k=1}^{K_s} p(z_{ti} = k \mid \mathbf{w}, \theta^{(s)})\, \mathbb{E}\big[x_{ti} \mid w_{ti}, \Lambda_k^{(s)}\big]$$

where the posterior state probabilities $p(z_{ti} \mid \mathbf{w}, \theta)$ may be efficiently computed via the belief propagation algorithm [4, 18, 31]. The sample–specific conditional mean estimate reduces to linear least squares smoothing:

$$\mathbb{E}\big[x_{ti} \mid w_{ti}, \Lambda_k^{(s)}\big] = \Lambda_k^{(s)}(\Lambda_k^{(s)} + \Sigma_n)^{-1} w_{ti}, \tag{15}$$

where we have assumed that the emission distributions are zero-mean. See Appendix D.1 for a general formulation. The denoised image is then determined via an inverse wavelet transform combining observed scaling coefficients with the posterior mean of each detail coefficient, obtained by averaging over the sample–specific conditional mean estimates $\mathbb{E}\big[x_{ti} \mid \mathbf{w}, \theta^{(s)}\big]$.

5.2. *Transfer Denoising.* Most existing image denoising algorithms estimate unknown parameters directly from the noisy image at hand. While effective in some cases, at high noise levels there can be insufficient information, and flexible models may lead to significantly distorted reconstructions. To avoid this, we propose a learning algorithm for denoising, summarized in algorithm 3, which uses information from the noisy image as well as prior knowledge of multiscale hidden state patterns learned from a database of clean images, in parameter estimation.

We begin learning by running a blocked Gibbs sampler summarized in Algorithm 2 on a set of clean images in wavelet-domain. After “burn–in”, we collect S samples from the parameters’ posterior distribution[4].

Then we transfer statistics by running the blocked Gibbs sampler summarized in Algorithm 5 on the noisy wavelet tree, conditioning on each of the samples separately. This way estimation uses statistics learned from both clean images, and the noisy test image at hand. After “burn–in”, we collect a sample $\theta^{(s)} = \{\pi_k^{(s)}, \Lambda_k^{(s)}\}_{k=1}^{K_s}$ from each of the chains, compute the conditional mean estimates $\mathbb{E}\big[x_{ti} \mid \mathbf{w}, \theta^{(s)}\big]$, average over them to obtain posterior mean estimates for clean detail coefficients, and finally apply the inverse wavelet transform combining observed scaling coefficients with the clean detail coefficient estimates.

Given a set of clean images, and a noisy image, corrupted by additive white $\mathcal{N}(0, \Sigma_n)$ noise:

1. Learn statistics of clean images by running the blocked Gibbs sampler summarized in Algorithm 2 on clean images in wavelet representations until burn-in.
2. Collect S samples from the converged chain.
3. Apply clean image statistics transfer by running the blocked Gibbs sampler summarized in Algorithm 5 on the noisy graph conditioning separately on the samples, and collecting a sample $\theta^{(s)} = \{\pi_k^{(s)}, \Lambda_k^{(s)}\}_{k=1}^{K_s}$ from each of the chains after burn-in.
4. Estimate posterior hidden state probabilities $p(z_{ti} \mid \mathbf{w}, \theta)$ via the belief propagation algorithm.
5. For each sample, estimate denoised coefficients in closed form:

$$\begin{aligned}\mathbb{E}\big[x_{ti} \mid \mathbf{w}, \theta^{(s)}\big] &= \sum_{k=1}^{K_s} p(z_{ti} = k \mid \mathbf{w}, \theta^{(s)})\, \mathbb{E}\big[x_{ti} \mid w_{ti}, \Lambda_k^{(s)}\big] \\ &= \sum_{k=1}^{K_s} p(z_{ti} = k \mid \mathbf{w}, \theta^{(s)})\, \Lambda_k^{(s)}(\,\Lambda_k^{(s)} + \Sigma_n)^{-1} w_{ti}.\end{aligned}$$

6. Average over samples of varying complexity $\mathbb{E}\big[x_{ti} \mid \mathbf{w}, \theta^{(s)}\big]$ to get posterior mean of detail coefficients.
7. Apply inverse wavelet transform to a combination of observed scaling coefficients w_{t0} with estimated detail coefficients x_{ti}.

Algorithm 3: The overall learning algorithm for transfer denoising with HDP-HMTs. Steps 1–2 are done offline, those computations being shared for all images to be denoised.

[4] these procedures are done offline and are shared for all images to be denoised

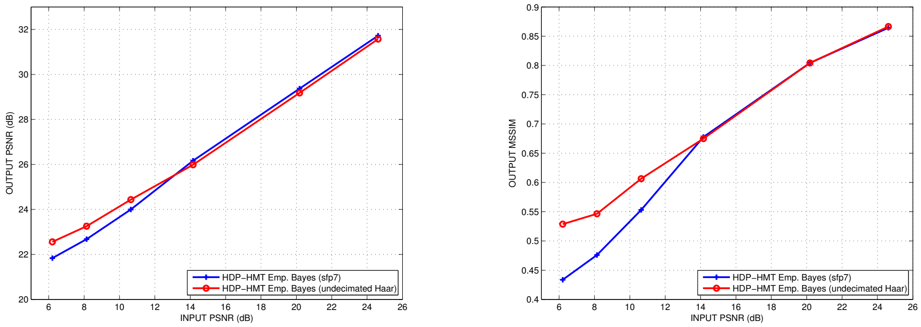


FIG 15. *Average peak signal-to-noise ratio (PSNR; left) and mean structural similarity index (MSSIM; right) values as a function of input PSNR, computed over the Empirical Bayesian denoising results of an ensemble of standard denoising images (cameraman, einstein, house, mandrill, peppers) using 7^{th}-order steerable pyramids, and undecimated Haar wavelets.*

5.3. *Results.* Figure 15 shows average denoising performance of the Empirical Bayesian algorithm in terms of peak signal-to-noise ratio (PSNR) and mean structural similarity index (SSIM). We can see that using 7^{th}–order steerable pyramid decomposition yields better results at lower noise levels, and worse results at higher noise levels than when using undecimated Haar wavelets. In figure 16 we compare the HDP-HMT's denoising performance (using 7^{th}-order steerable pyramids) to two other methods. Using an empirical Bayesian denoising algorithm, our results at low and moderate noise levels are comparable to BLS-GSM, one of the most effective wavelet-based denoising methods[5]. However, at higher noise levels, increasing high-frequency artifacts start to reduce restoration quality (see figures 18&16). By learning the statistics of a set of 200 clean natural images from the Berkeley segmentation dataset, the HDP-HMT learns that images typically contain many smooth or homogeneously textured regions, separated by sharp edges. The denoising algorithm transfers this prior knowledge by reusing multi-scale hidden state patterns, resulting in better reconstruction of distorted textures at higher noise levels, especially with respect to the perceptual MSSIM criterion, as also can be seen from the tables 2&3. As we can see from figure 20, statistics transfer is effective almost immediately, denoising performance converging using a single sample after a short number of iterations - regardless of the noise level. In the empirical Bayesian approach, convergence takes increasingly more iterations, and denoising performance relative to transfer denoising approach using samples from converged chain decreases, as the noise level increases. In these higher noise regimes, transfer denoising with the HDP-HMT also surpasses the performance

[5] PSNR-wise leading wavelet-based denoising method [15] is not used in our comparisons as sufficient performance information or software is not publicly available

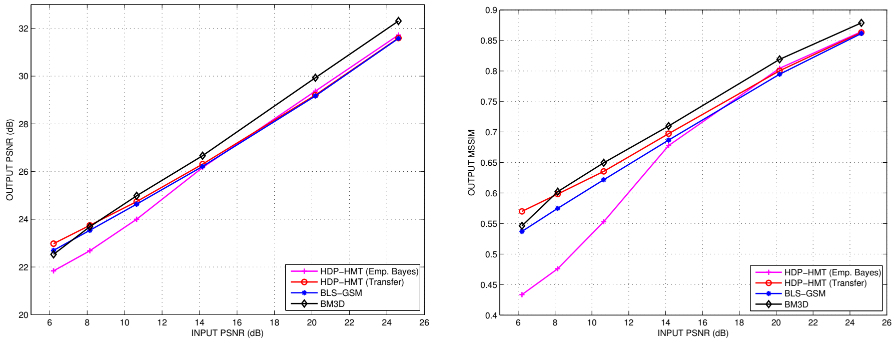
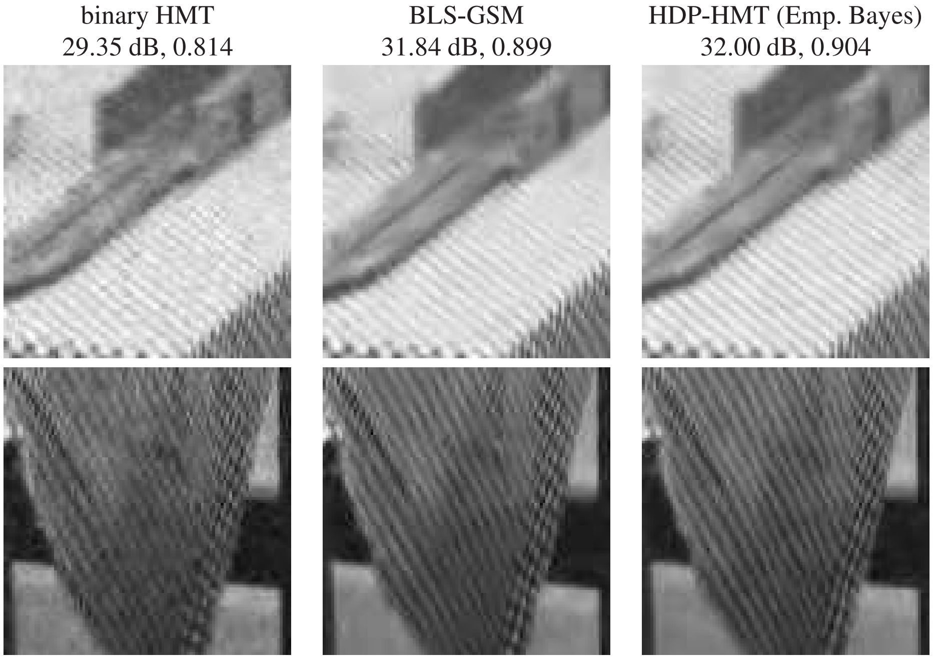


FIG 16. *Average peak signal-to-noise ratio (PSNR; left) and mean structural similarity index (MSSIM; right) values as a function of input PSNR, computed over the denoising results of an ensemble of standard denoising images (cameraman, einstein, house, mandrill, peppers). The wavelet-based methods (HDP-HMT, BLS-GSM) use 7^{th}-order steerable pyramids, whereas BM3D analyzes images in blocks.*


binary HMT
29.35 dB, 0.814

BLS-GSM
31.84 dB, 0.899

HDP-HMT (Emp. Bayes)
32.00 dB, 0.904

FIG 17. *Comparison of denoising results of the Barbara image with noise level $\sigma = 15$. Zoomed up regions are shown to reveal the artifacts.*

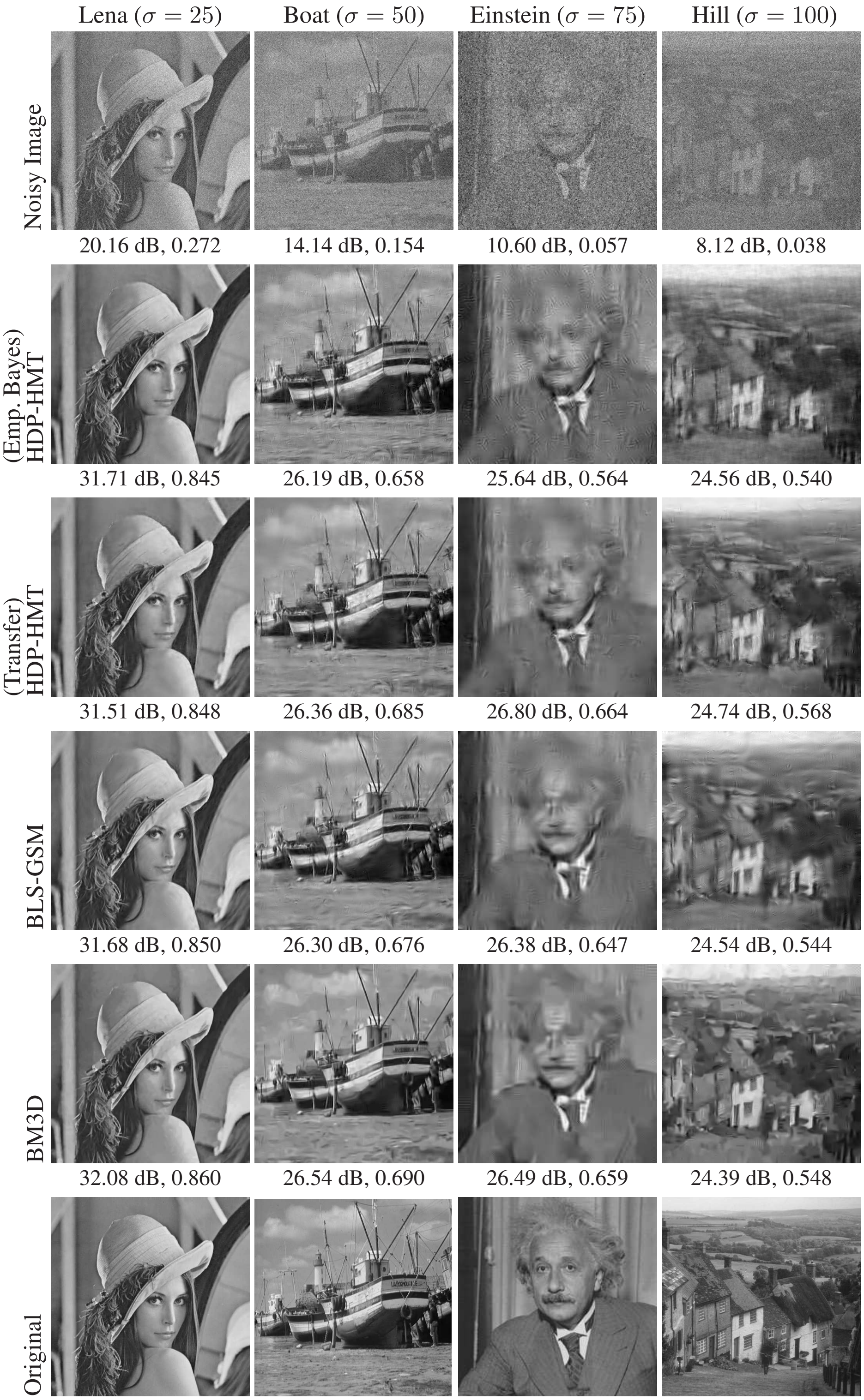


FIG 18. *Denoising* Lena, Boat, Einstein *and* Hill *images contaminated by additive white Gaussian noise of standard deviation σ; with HDP-HMT, BLS-GSM [19], and BM3D [6]. The left-most and right-most performance numbers beneath the images correspond to PSNR and MSSIM values, respectively.*

of the BLS-GSM. At extreme noise levels, the results are comparable to or better than those of BM3D, a state-of-the-art algorithm which averages similar blocks of pixels.

To further investigate the performance of the methods, we considered the task of natural scene denoising, from coast- and tall building-categories, and from the BSDB-dataset. The dataset-specific HDP-HMT models were trained on 200 images. For testing, we chose randomly[6] 10 images not used in training from each of the categories.

Figure 19 shows average denoising results relative to the results obtained with an HDP-HMT model pre-trained on (200) images from the same category as the test image. The figure contains also HDP-HMT results pre-trained on (200) images from other datasets than the test image as well as results of the empirical Bayesian approach and of BLS-GSM, and BM3D. When comparing the different HDP-HMT results, best results are obtained when the training and test data come from the same category. Interestingly the model trained on images from the Berkeley segmentation dataset (BSDB) performs better than scene-specific models tested on images from other scene categories than they are trained on. The result seems intuitive, since the BSDB-dataset is not specialized to any particular scene, and contains images with highly diverse structures, including structures similar to those exhibited in the scene-specific datasets.

When comparing HDP-HMT to the state-of-the-art methods, we can see that BLS-GSM obtains worst average results on this test. BM3D obtains best results on average on tall building-category and BSDB-dataset, while HDP-HMT clearly outperforms the other methods on the coast-category.

6. Conclusion.

We have developed a nonparametric, data–driven model for image features which captures spatial dependencies via a multiscale graphical model. Our results show that this HDP-HMT captures natural scene statistics more accurately than bag–of–feature models, and leads to improved categorization performance.

We have also shown that the HDP-HMT is able to learn complex statistics of wavelets, and demonstrated its effectiveness in an image denoising task. By learning the statistics from natural images, the HDP-HMT is able to transfer prior knowledge of clean multiscale hidden state patterns, resulting in better reconstruction of distorted textures at higher noise levels in denoising than an Empirical Bayesian approach. We expect that transfer of natural image statistics will prove useful for correcting other forms of image distortion, such as significant motion blur.

[6] images of poor quality or of too similar content with previously chosen ones were not considered

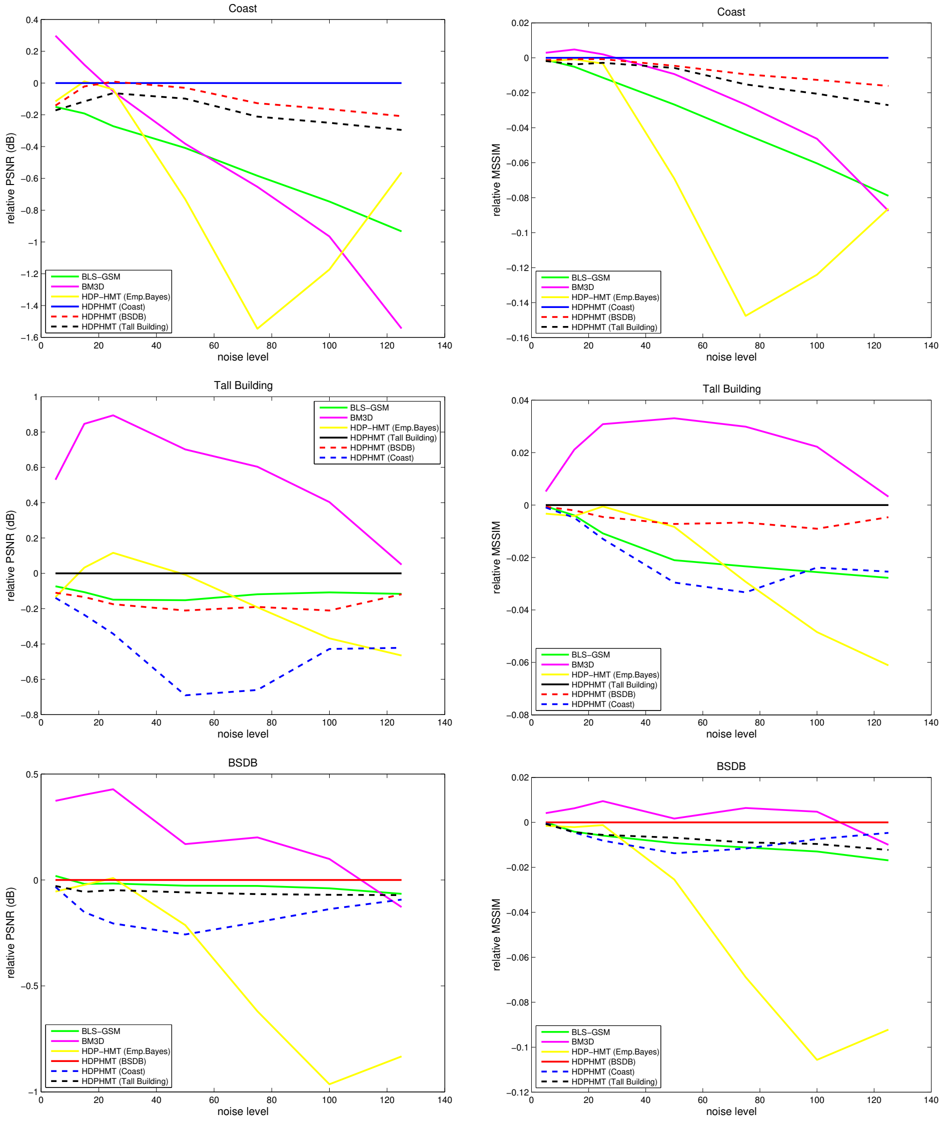


FIG 19. *Average peak signal-to-noise ratio (PSNR; top) and mean structural similarity index (MSSIM; bottom) values as a function of input PSNR, computed over the denoising results of an ensemble of images from a scene-specific dataset (from left to right coast, tall building, BSDB).*

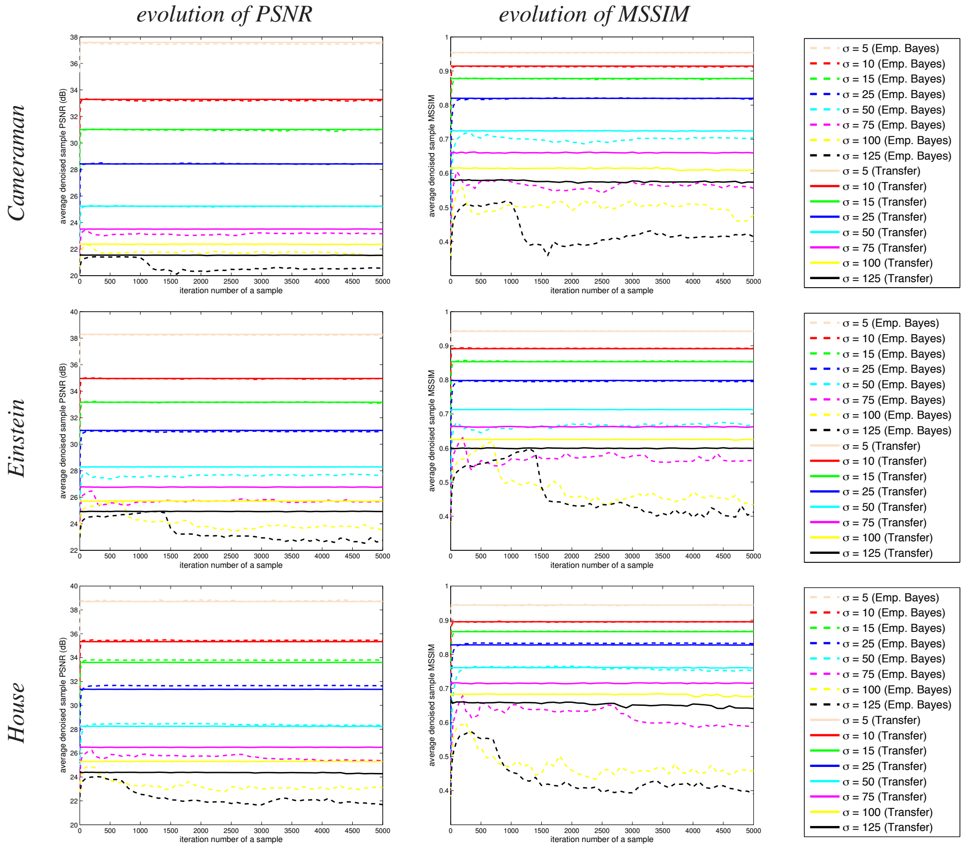


FIG 20. *Denoising performance based on using a single sample for estimation, as a function of the iteration number of the sample in the sampling algorithm. Empirical Bayesian denoising results are drawn with dashed lines, transfer denoising results with solid lines. Different colors correspond to different noise levels.*

References.

[1] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification via pLSA. In *European Conference on Computer Vision*, pages 517–530, 2006.

[2] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:712–727, 2008.

[3] H. A. Chipman, E. D. Kolaczyk, and R. E. McCulloch. Adaptive Bayesian wavelet shrinkage. *J. Amer. Stat. Assoc.*, 92(440):1413–1421, 1997.

[4] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet–based statistical signal processing using hidden Markov models. *IEEE Trans. Sig. Proc.*, 46(4):886–902, 1998.

[5] Zoran D. and Weiss Y. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision*, 2011.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. Image Proc.*, 16(8):2080–2095, 2007.

[7] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Vision and Pattern Recognition*, volume 2, pages 524–531, 2005.

[8] A. Hyvarinen, J. Hurri, and P. O. Hoyer. *Natural Image Statistics*. Springer-Verlag, 2009. In press.

[9] H. Ishwaran and M. Zarepour. Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87(2):371–390, 2000.

[10] H. Ishwaran and M. Zarepour. Exact and approximate sum–representations for the Dirichlet process. *Can. J. Stat.*, 30:269–283, 2002.

[11] J. J. Kivinen, E. B. Sudderth, and M. I. Jordan. Image denoising with nonparametric hidden Markov trees. In *IEEE International Conference on Image Processing*, volume 3, pages 121–124, 2007.

[12] J. J. Kivinen, E. B. Sudderth, and M. I. Jordan. Learning multiscale representations of natural scenes using Dirichlet processes. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Vision and Pattern Recognition*, volume 2, pages 2169–2176, 2006.

[14] D. G. Lowe. Distinctive image features from scale–invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] S. Lyu and E. P. Simoncelli. Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures. *IEEE Trans. Patt. Analysis and Machine Intelligence*, 31(4):693–706, 2009.

[16] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[17] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, (37), 1997.

[18] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, 1988.

[19] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Proc.*, 12(11):1338–1351, 2003.

[20] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *IEEE International Conference on Computer Vision*, volume 1, pages 883–890, 2005.

[21] S. Roth and M. J Black. Fields of Experts: A framework for learning image priors. In *CVPR*,

pages II: 860–867, 2005.

[22] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE Trans. Info. Theory*, 38(2):857–607, 1992.

[23] C. Spence, L. C. Parra, and P. Sajda. Varying complexity in tree-structured image distribution models. *IEEE Trans. Image Proc.*, 15(2):319–330, 2006.

[24] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18:17–33, 2003.

[25] E. B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology, May 2006.

[26] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Stat. Assoc.*, 101(476):1566–1581, 2006.

[27] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

[28] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Comp. Neural Sys.*, 14:391–412, 2003.

[29] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007.

[30] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky. Random cascades on wavelet trees and their use in analyzing and modeling natural images. *Applied and Computational Harmonic Analysis*, 11:89–123, 2001.

[31] A. S. Willsky. Multiresolution Markov models for signal and image processing. *Proc. IEEE*, 90(8):1396–1458, 2002.

APPENDIX A: COLLAPSED GIBBS SAMPLING

A.1. Sampling assignments of data points to clusters z_{ti}. From Fig. 8, the posterior distribution of z_{ti} given all other state assignments $\mathbf{z}_{\setminus ti}$ factors as

$$p(z_{ti} \mid \mathbf{z}_{\setminus ti}, \beta, \mathbf{x}) \propto p(z_{ti} \mid \mathbf{z}_{\setminus ti}, \beta)\, p(x_{ti} \mid \mathbf{x}_{\setminus ti}, \mathbf{z}) \tag{16}$$

The second term is the *predictive likelihood* of x_{ti}, which for inverse–Wishart priors is multivariate Student–t [25]. The form of the first term depends on the position i of the sampled coefficient, the states of its neighbors, and tying options.

Let $n_{\setminus ti}^{d(ti)}(k, \ell)$ denote the number of transitions from parent state k to child state ℓ with direction $d(ti)$ instantiated by $\mathbf{z}_{\setminus ti}$, and $n_{\setminus ti}^{d(ti)}(k, \cdot)$ the total number of outgoing transitions from state k to direction $d(ti)$. For finest scale coefficients,

$$\begin{aligned} p\big(z_{ti} \mid z_{\mathrm{Pa}(ti)} = k, \mathbf{z}_{\setminus ti}, \beta\big) &= \int \pi_k(z_{ti}) p(\pi_k \mid \mathbf{z}_{\setminus ti}, \beta)\ d\pi_k^{d(ti)} \\ &= \left(\frac{n_{\setminus ti}^{d(ti)}(k, z_{ti}) + \alpha\beta(z_{ti})}{n_{\setminus ti}^{d(ti)}(k, \cdot) + \alpha} \right) \end{aligned} \tag{17}$$

The form of this ratio follows from the properties of Dirichlet distributions.

When evaluating eq. (17), we consider candidate states z_{ti} corresponding to every state which is used at least once elsewhere in the wavelet tree, as well as a potential *new* state. This predictive rule allows HDP-HMTs to determine state space cardinality in a data–driven fashion.

For non–leaf nodes, $p(z_{ti} \mid \mathbf{z}_{\setminus ti}, \beta)$ is also influenced by its childrens' states $z_{\mathrm{Ch}(ti)} \triangleq \{z_{tj} \mid tj \in \mathrm{Ch}(ti)\}$, and tying options. In the following the analytical results derived are for the model assuming separate transition probabilities for each parent-child-direction, which we found performing best in our experiments.

In candidate states where $z_{ti} \neq z_{\mathrm{Pa}(ti)}$,

$$
\begin{aligned}
(18)\quad p(z_{ti} \mid \mathbf{z}_{\setminus ti}, \beta) = \int \pi^{d^{(ti)}}_{z_{\mathrm{Pa}(ti)}}(z_{ti}) p(\pi^{d^{(ti)}}_{z_{\mathrm{Pa}(ti)}} | \mathbf{z}_{\setminus ti}, \alpha, \beta)\, d\pi^{d^{(ti)}}_{z_{\mathrm{Pa}(ti)}} \\
\cdot \left(\prod_{tj \in \mathrm{Ch}(ti)} \int \pi^{d^{(tj)}}_{z_{ti}}(z_{tj}) p(\pi^{d^{(tj)}}_{z_{ti}} | \mathbf{z}_{\setminus ti}, \alpha)\, d\pi^{d^{(tj)}}_{z_{ti}} \right) \\
= \left(\frac{n^{d(ti)}_{\setminus ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) + \alpha\beta_{z_{ti}}}{n^{d(ti)}_{\setminus ti}(z_{\mathrm{Pa}(ti)}, \cdot) + \alpha} \right) \left[\prod_{tj \in \mathrm{Ch}(ti)} \left(\frac{n^{d(ti)}_{\setminus ti}(z_{ti}, z_{tj}) + \alpha\beta_{z_{tj}}}{n^{d(ti)}_{\setminus ti}(z_{ti}, \cdot) + \alpha} \right) \right]
\end{aligned}
$$

The case, when a candidate state equals that of the parent is slightly more complicated. Let z_{tl} denote the child of node z_{ti}, along the same transition direction as that from the parent (so that $d(tl) = d(ti)$). Then for candidate states where $z_{ti} = z_{\mathrm{Pa}(ti)} = k$,

$$
\begin{aligned}
(19)\quad p(z_{ti} \mid \mathbf{z}_{\setminus ti}, \beta) = \int \pi^{d(ti)}_{z_{\mathrm{Pa}(ti)}}(z_{ti}) \pi^{d(tl)}_{z_{ti}}(z_{tl}) p(\pi^{d(ti)}_{z_{\mathrm{Pa}(ti)}} | \mathbf{z}_{\setminus ti}, \alpha, \beta)\, d\pi^{d(ti)}_{z_{\mathrm{Pa}(ti)}} \\
\cdot \left(\prod_{tj \in \mathrm{Ch}(ti) \neq tl} \int \pi^{d(tj)}_{z_{ti}}(z_{tj}) p(\pi^{d(tj)}_{z_{ti}} | \mathbf{z}_{\setminus ti}, \alpha)\, d\pi^{d(tj)}_{z_{ti}} \right) = \\
\int \pi^{d(ti)}_{k}(k) \pi^{d(ti)}_{k}(z_{tl}) p(\pi^{d(ti)}_{k} | \mathbf{z}_{\setminus ti}, \alpha, \beta)\, d\pi^{d(ti)}_{k} \left[\prod_{tj \in \mathrm{Ch}(ti) \neq tl} \left(\frac{n^{d(tj)}_{\setminus ti}(z_{ti}, z_{tj}) + \alpha\beta_{z_{tj}}}{n^{d(tj)}_{\setminus ti}(z_{ti}, \cdot) + \alpha} \right) \right]
\end{aligned}
$$

Let us now compute the term involving the integral in the above product.

$$
\begin{aligned}
(20)\quad & \\
p(z_{ti} \mid \mathbf{z}_{\setminus ti}, \beta) = \left(\frac{n^{d(tl)}_{\setminus ti}(k, k) + \alpha\beta_{k}}{n^{d(tl)}_{\setminus ti}(k, \cdot) + \alpha} \right) \left(\frac{n^{d(tl)}_{\setminus ti}(k, z_{tl}) + \delta(k, z_{tl}) + \alpha\beta_{z_{tl}}}{n^{d(tl)}_{\setminus ti}(k, \cdot) + 1 + \alpha} \right) \\
\cdot \left[\prod_{tj \in \mathrm{Ch}(ti) \neq tl} \left(\frac{n^{d(tj)}_{\setminus ti}(z_{ti}, z_{tj}) + \alpha\beta_{z_{tj}}}{n^{d(tj)}_{\setminus ti}(z_{ti}, \cdot) + \alpha} \right) \right]
\end{aligned}
$$

Combining the results for non-leaf nodes,

$$p(z_{ti} \mid \mathbf{z}_{\setminus ti}, \beta) = \left(\frac{n^{d(ti)}_{\setminus ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) + \alpha\beta_{z_{ti}}}{n^{d(ti)}_{\setminus ti}(z_{\mathrm{Pa}(ti)}, \cdot) + \alpha} \right) \\ \cdot \left(\frac{n^{d(ti)}_{\setminus ti}(z_{ti}, z_{tl}) + \alpha\beta_{z_{tl}} + \delta(z_{\mathrm{Pa}(ti)}, z_{ti})\delta(z_{ti}, z_{tl})}{n^{d(ti)}_{\setminus ti}(z_{ti}, \cdot) + \alpha + \delta(z_{\mathrm{Pa}(ti)}, z_{ti})} \right) \prod_{tj \in \mathrm{Ch}(ti) \setminus tl} \left(\frac{n^{d(tj)}_{\setminus ti}(z_{ti}, z_{tj}) + \alpha\beta_{z_{tj}}}{n^{d(ti)}_{\setminus ti}(z_{ti}, \cdot) + \alpha} \right) \tag{21}$$

where $tl \in \mathrm{Ch}(ti)$, and $d(tl) = d(ti)$.

A.2. Sampling global transition counts $\boldsymbol{\beta}$. Sampling β can be done with the auxiliary variable technique by [26]. Let m_{jk} denote the number of tables assigned to mixture component k in group/mixture j in the chinese restaurant franchise. Given $n^d(j,k)$ transitions from state j to state k in direction d observed from $\mathrm{DP}(\alpha, \beta)$,

$$p(m^d_{jk} = m \,|\, \beta, \alpha, \mathbf{z}) = \frac{\Gamma(\alpha\beta_k)}{\Gamma(\alpha\beta_k + n^d(j,k))} s(n^d(j,k), m)(\alpha\beta_k)^m \tag{22}$$

where $s(n,m)$ are stirling numbers of first kind. If these numbers get large, sampling from the conditional can become computationally very expensive. However, one can also sample the number of tables by simulating the Chinese Restaurant Process (CRP) [26], counting the number of tables occupied after seating $n^d(j,k)$ customers. We found this approach much more efficient in our experiments with large datasets.

Then given m, β can be sampled from

$$\{\beta_1, \ldots, \beta_K, \beta_u\} \,|\, m, \gamma \sim \mathcal{D}\left(m^{\cdot}_{\cdot 1}, \ldots, m^{\cdot}_{\cdot K}, \gamma\right) \tag{23}$$

where $m^{\cdot}_{\cdot k}$ denotes the total number of tables assigned in the mixtures to mixture component k.

APPENDIX B: EXACT ESTIMATION USING BELIEF PROPAGATION

In this section, we will derive algorithms for exact estimation based on belief propagation, used in various learning algorithms developed in the manuscript. We start by the problem of computing the likelihood of a data case, used in the categorization of natural scenes. The subsection will be described in larger detail than the later subsections, as the problem solving mechanisms, and the computations underlying the problems are shared in great detail.

B.1. Computing likelihood of a data case.

$$\begin{aligned} p(\mathbf{x}\,|\,\pi,\theta) &= \sum_{\mathbf{z}} p(\mathbf{z},\mathbf{x}\,|\,\pi,\theta) = \sum_{\mathbf{z}} p(\mathbf{z}\,|\,\pi)p(\mathbf{x}\,|\,\mathbf{z},\theta) \\ &= \sum_{\mathbf{z}}\prod_{\ell=1}^{L}\prod_{i\in\mathcal{V}_\ell} p(z_i\,|\,z_{\mathrm{Pa}(i)},\pi)p(x_i\,|\,\theta_{z_i}) = \sum_{\mathbf{z}}\prod_{\ell=1}^{L}\prod_{i\in\mathcal{V}_\ell}\pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)\,p(x_i\,|\,\theta_{z_i}), \end{aligned}\tag{24}$$

where ℓ indexes the depth within the tree of L scales, and nodes at depth ℓ have an index set $\mathcal{V}_\ell$. The belief propagation algorithm solves the summation/elimination problem efficiently using a scale-recursive procedure. Starting by pushing the sums over the bottom scale hidden state-assignment variables as far as possible, we can write that

$$p(\mathbf{x}\,|\,\pi,\theta) = \sum_{\mathbf{z}_{\mathcal{V}\setminus L}}\prod_{\ell=1}^{L-1}\prod_{i\in\mathcal{V}_\ell}\pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)\,p(x_i\,|\,\theta_{z_i})\tag{25}$$

$$\cdot\sum_{\mathbf{z}_{\mathcal{V}_L}}\prod_{j\in\mathcal{V}_L}\pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\,p(x_j\,|\,\theta_{z_j})\tag{26}$$

$$= \sum_{\mathbf{z}_{\mathcal{V}\setminus L}}\prod_{\ell=1}^{L-1}\prod_{i\in\mathcal{V}_\ell}\pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)\,p(x_i\,|\,\theta_{z_i})\tag{27}$$

$$\cdot\prod_{j\in\mathcal{V}_L}\sum_{z_j}\pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\,p(x_j\,|\,\theta_{z_j}).\tag{28}$$

Proceeding on the elimination, we now push the sums over the hidden state-assignment nodes of the second-deepest level of the tree as far as possible, and have that

$$\begin{aligned} p(\mathbf{x}\,|\,\pi,\theta) = \sum_{\mathbf{z}_{\mathcal{V}\setminus L-1,L}}&\prod_{\ell=1}^{L-2}\prod_{i\in\mathcal{V}_\ell}\pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)\,p(x_i\,|\,\theta_{z_i})\cdot \\ &\prod_{j\in\mathcal{V}_{L-1}}\sum_{z_j}\pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\,p(x_j\,|\,\theta_{z_j})\prod_{k\in\mathcal{V}_L}\sum_{z_k}\pi^{d(k)}_{z_{\mathrm{Pa}(k)}}(z_k)\,p(x_k\,|\,\theta_{z_k}). \end{aligned}\tag{29}$$

This elimination structure persists over the different scales, and we can solve the elimination efficiently by a message passing recursion:

$$\omega_j^{\mathrm{Pa}(j)}\big(z_{\mathrm{Pa}(j)}\big) = \sum_{z_j}\pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\,p(x_j\,|\,\theta_{z_j})\prod_{k\in\mathrm{Ch}(j)}\omega_k^j(z_j)\,,\tag{30}$$

initialized by the computation of (un-normalized) messages from the bottom scale nodes to their parents $\omega_k^{\mathrm{Pa}(k)}\big(z_{\mathrm{Pa}(k)}\big) = \sum_{z_k}\pi^{d(k)}_{z_{\mathrm{Pa}(k)}}(z_k)\,p(x_k\,|\,\theta_{z_k})$, where $k \in$

$\mathcal{V}_L$, and terminated at the first scale nodes $j \in \mathcal{V}_1$ (in which case $k \in \mathcal{V}_2$). Note that for each node index j we have that

$$\omega_j^{\mathrm{Pa}(j)}\big(z_{\mathrm{Pa}(j)}\big) = p(x_{\mathrm{R}_j} \,|\, z_{\mathrm{Pa}(j)}, \pi, \theta), \tag{31}$$

where x_{R_j} denotes the observations of the subtree rooted from node j. Likelihood of the data can be then written as follows:

$$\begin{aligned} p(\mathbf{x} \,|\, \pi, \theta) &= \prod_{j \in \mathcal{V}_1} \sum_{z_j} \pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\, p(x_j \,|\, \theta_{z_j}) \prod_{k \in \mathrm{Ch}(j)} \omega_k^j(z_j) \\ &= \prod_{j \in \mathcal{V}_1} \omega_j^{\mathrm{Pa}(j)}\big(z_{\mathrm{Pa}(j)}\big)\,, \end{aligned} \tag{32}$$

where $z_{\mathrm{Pa}(j)}$ is a fixed distinguished state for the first scale nodes.

To avoid numerical issues, we use an alternative recursive scheme for exact likelihood computation based on normalized messages. The messages for bottom scale nodes ($k \in \mathcal{V}_L$) are the following:

$$m_k^{\mathrm{Pa}(k)}\big(z_{\mathrm{Pa}(k)}\big) = \frac{1}{c_k} \sum_{z_k} \pi^{d(k)}_{z_{\mathrm{Pa}(k)}}(z_k)\, p(x_k \,|\, \theta_{z_k}), \tag{33}$$

where the normalization constant

$$c_k = \sum_{z_{\mathrm{Pa}(k)}} \sum_{z_k} \pi^{d(k)}_{z_{\mathrm{Pa}(k)}}(z_k)\, p(x_k \,|\, \theta_{z_k}) = \sum_{z_{\mathrm{Pa}(j)}} \omega_j^{\mathrm{Pa}(j)}\big(z_{\mathrm{Pa}(j)}\big)\,.$$

For intermediate nodes ($j \in \mathcal{V}_{\setminus 1,L}$)

$$\begin{aligned} m_j^{\mathrm{Pa}(j)}\big(z_{\mathrm{Pa}(j)}\big) &= \frac{1}{c_j} \sum_{z_j} \pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\, p(x_j \,|\, \theta_{z_j}) \prod_{k \in \mathrm{Ch}(j)} m_k^j(z_j) \\ c_j &= \sum_{z_{\mathrm{Pa}(j)}} \sum_{z_j} \pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\, p(x_j \,|\, \theta_{z_j}) \prod_{k \in \mathrm{Ch}(j)} m_k^j(z_j)\,. \end{aligned} \tag{34}$$

For first scale nodes ($j \in \mathcal{V}_1$)

$$\begin{aligned} m_j^{\mathrm{Pa}(j)}\big(z_{\mathrm{Pa}(j)}\big) &= \sum_{z_j} \pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\, p(x_j \,|\, \theta_{z_j}) \prod_{k \in \mathrm{Ch}(j)} m_k^j(z_j) \\ &= c_j. \end{aligned} \tag{35}$$

After the upwards-recursive sweep, exact likelihood can be then computed as:

$$p(\mathbf{x}\,|\,\pi,\theta) = \prod_{j\in\mathcal{V}_1}\sum_{z_j}\pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\,p(x_j\,|\,\theta_{z_j})\prod_{k\in\mathrm{Ch}(j)}\omega^{j}_{k}(z_j)$$

$$= \left(\prod_{h\in\mathcal{V}_{\setminus 1}} c_h\right)\prod_{j\in\mathcal{V}_1}\sum_{z_j}\pi^{d(j)}_{z_{\mathrm{Pa}(j)}}(z_j)\,p(x_j\,|\,\theta_{z_j})\prod_{k\in\mathrm{Ch}(j)}m^{j}_{k}(z_j) \tag{36}$$

$$= \left(\prod_{h\in\mathcal{V}_{\setminus 1}} c_h\right)\prod_{j\in\mathcal{V}_1} c_j = \prod_{i\in\mathcal{V}} c_i, \tag{37}$$

where the pre-multiplying normalization constants in the latter two equalities are inverting the effect in terms of the final result of using normalized messages as opposed to un-normalized ones in the message-passing recursion, to get the exact result.

B.2. Computing hidden state marginal probabilities. For first scale nodes $(i \in \mathcal{V}_1)$:

$$p(z_i\,|\,z_{\mathrm{Pa}(i)},\mathbf{x},\pi,\theta) = \sum_{\mathbf{z}_{\mathrm{Ch}(i)}} p(z_i,\mathbf{z}_{\mathrm{Ch}(i)}\,|\,,z_{\mathrm{Pa}(i)},\mathbf{x},\pi,\theta)$$

$$= p(z_i\,|\,z_{\mathrm{Pa}(i)},\mathbf{x},\pi,\theta)\sum_{\mathbf{z}_{\mathrm{Ch}(i)}}\prod_{j\in\mathrm{Ch}(i)} p(z_j\,|\,z_i,\mathbf{x},\pi,\theta)$$

$$= \frac{p(x_i\,|\,\theta_{z_i})\pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)}{p(\mathbf{x}\,|\,z_{\mathrm{Pa}(i)},\pi,\theta)}\prod_{j\in\mathrm{Ch}(i)}\sum_{z_j}\frac{p(z_j,\mathbf{x}_{\mathrm{R}_j}\,|\,z_i,\pi,\theta)}{p(\mathbf{x}_{\mathrm{R}_j}\,|\,z_i,\pi,\theta)}$$

$$\propto p(x_i\,|\,\theta_{z_i})\pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)\prod_{j\in\mathrm{Ch}(i)} m^{i}_{j}(z_i)\,, \tag{38}$$

where $z_{\mathrm{Pa}(i)}$ is a fixed distinguished parent state. For nodes beneath the first scale ($i \in \mathcal{V}_{\setminus 1}$):

$$\begin{aligned}
p(z_i \,|\, \mathbf{x}, \pi, \theta) &= \sum_{z_{\mathrm{Pa}(i)}, \mathbf{z}_{\mathrm{Ch}(i)}} p(z_{\mathrm{Pa}(i)}, z_i, z_{\mathrm{Ch}(i)} \,|\, \mathbf{x}, \pi, \theta) \\
&= \sum_{z_{\mathrm{Pa}(i)}} p(z_{\mathrm{Pa}(i)} \,|\, \mathbf{x}, \pi, \theta) p(z_i \,|\, z_{\mathrm{Pa}(i)}, \mathbf{x}, \pi, \theta) \sum_{\mathbf{z}_{\mathrm{Ch}(i)}} \prod_{j \in \mathrm{Ch}(i)} p(z_j \,|\, z_i, \mathbf{x}, \pi, \theta) \\
&= \sum_{z_{\mathrm{Pa}(i)}} p(z_{\mathrm{Pa}(i)} \,|\, \mathbf{x}, \pi, \theta) p(z_i \,|\, z_{\mathrm{Pa}(i)}, \mathbf{x}, \pi, \theta) \prod_{j \in \mathrm{Ch}(i)} \sum_{z_j} p(z_j \,|\, z_i, \mathbf{x}, \pi, \theta) \\
&= \sum_{z_{\mathrm{Pa}(i)}} p(z_{\mathrm{Pa}(i)} \,|\, \mathbf{x}, \pi, \theta) \frac{p(\mathbf{x} \,|\, z_i, z_{\mathrm{Pa}(i)}, \pi, \theta) \pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)}{p(\mathbf{x} \,|\, z_{\mathrm{Pa}(i)}, \pi, \theta)} \\
&\cdot \prod_{j \in \mathrm{Ch}(i)} \sum_{z_j} p(z_j \,|\, z_i, \mathbf{x}, \pi, \theta) \\
&\propto \sum_{z_{\mathrm{Pa}(i)}} p(z_{\mathrm{Pa}(i)} \,|\, \mathbf{x}, \pi, \theta) \frac{p(x_i \,|\, z_i, \theta) \pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)}{m_i^{\mathrm{Pa}(i)}(z_{\mathrm{Pa}(i)})} \prod_{j \in \mathrm{Ch}(i)} m_j^i(z_i) \\
&= p(x_i \,|\, \theta_{z_i}) \prod_{j \in \mathrm{Ch}(i)} m_j^i(z_i) \sum_{z_{\mathrm{Pa}(i)}} \pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i) \frac{p(z_{\mathrm{Pa}(i)} \,|\, \mathbf{x}, \pi, \theta)}{m_i^{\mathrm{Pa}(i)}(z_{\mathrm{Pa}(i)})}. \qquad (39)
\end{aligned}$$

Therefore after we have computed the upwards message-passing sweep yielding the BP-messages for a node from its children as described in the previous subsection, we can recursively sweep downwards computing the conditional marginal probabilities of the hidden state-assignment variables. This second step is started at the first scale hidden state-assignment nodes, with the computation of (38) for each node, followed by a downwards-recursive sweep with the computation of (39) at the deeper scale nodes.

B.3. Computing the joint hidden state conditional probability. Using the product rule, and conditional independence properties of the tree-structured graph, we can write that

$$p(\mathbf{z} \,|\, \mathbf{x}, \pi, \theta) = \prod_i p(z_i \,|\, z_{\mathrm{Pa}(i)}, \mathbf{x}, \pi, \theta). \qquad (40)$$

These conditional distributions can be computed efficiently utilizing the upwards BP-messages. Indeed, for nodes on scales from 1 to $L-1$, indexed as $i \in \mathcal{V}_{\setminus L}$, we have by marginalizing over the child hidden state-assignment variables that

$$p(z_i \,|\, z_{\mathrm{Pa}(i)}, \mathbf{x}, \pi, \theta) \propto p(x_i \,|\, \theta_{z_i}) \pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i) \prod_{j \in \mathrm{Ch}(i)} m_j^i(z_i)\,. \qquad (41)$$

For bottom scale nodes, indexed as $i \in \mathcal{V}_L$, we have that

$$p(z_i \,|\, z_{\mathrm{Pa}(i)}, \mathbf{x}, \pi, \theta) \propto p(x_i \,|\, \theta_{z_i}) \pi^{d(i)}_{z_{\mathrm{Pa}(i)}}(z_i)\,. \tag{42}$$

APPENDIX C: BLOCKED GIBBS SAMPLING FOR TRUNCATED REPRESENTATIONS

C.1. Sampling Assignments via Belief Propagation. Messages are first passed from the leaves to the root of each tree to collect summary statistics, which can also be used to evaluate the marginal likelihood $p\big(\mathbf{x}_t \mid \{\pi_k, \theta_k\}_{k=1}^K\big)$ in closed form. The bottom–up message passing can be written as

$$m^{tj}_{ti}(z_{tj}) \propto \sum_{z_{ti}} \psi^{tj}_{ti}(z_{ti}, z_{tj}) \psi_{ti}(z_{ti}, x) \prod_{tk \in N(ti) \backslash tj} m^{ti}_{tk}(z_{ti}) \tag{43}$$

where $N(ij)$ denotes the neighbors of node ij, $\psi_{ij}(z_{ij}, x)$ is the joint belief of the hidden variable and the observations, and $m^{tj}_{t}(z_{tj})$ is the message from hidden variable ti to tj. For bottom scale nodes the messages can be written as

$$m^{tj}_{ti}(z_{tj}) \propto \sum_{z_{ti}} \pi^{d(ti)}_{z_{tj}}(z_{ti}) p(x_{ti} | \theta_{z_{ti}}) \tag{44}$$

For other nodes the messages are of form

$$m^{tj}_{ti}(z_{tj}) \propto \sum_{z_{ti}} \pi^{d(ti)}_{z_{tj}}(z_{ti}) p(x_{ti} | \theta_{z_{ti}}) \prod_{tk \in \mathrm{Ch}(ti)} m^{ti}_{tk}(z_{ti}) \tag{45}$$

A top–down recursion is then used to resample each node z_{ti} given its parent $z_{\mathrm{Pa}(ti)}$. Using the product rule and conditional independency rules for directed graphs, we can write the joint conditional probability of hidden states as

$$p(\mathbf{z} \,|\, \mathbf{x}_, \pi, \theta) = \prod_{t=1}^{T} \prod_{i=1}^{N(t)} p(z_{ti} \,|\, z_{\mathrm{Pa}(ti)}, \mathbf{x}_{t\cdot}, \pi, \theta). \tag{46}$$

For bottom scale nodes z_{ti},

$$p(z_{ti} \,|\, z_{\mathrm{Pa}(ti)}, \mathbf{x}_{t\cdot}, \pi, \theta) \propto \pi^{d(ti)}_{z_{\mathrm{Pa}(ti)}}(z_{ti}) p(x_{ti} \,|\, \theta_{z_{ti}}), \tag{47}$$

and for other nodes,

$$\begin{aligned} p(z_{ti} \,|\, z_{\mathrm{Pa}(ti)}, \mathbf{x}_{t\cdot}, \pi, \theta) &= \sum_{\mathbf{z}_{\mathrm{Ch}(ti)}} p(z_{ti}, \mathbf{z}_{\mathrm{Ch}(ti)} \,|\, z_{\mathrm{Pa}(ti)}, \mathbf{x}_{t\cdot}, \pi, \theta) \\ &= p(z_{ti} \,|\, z_{\mathrm{Pa}(ti)}, \mathbf{x}_{t\cdot}, \pi, \theta) \prod_{tj \in \mathrm{Ch}(ti)} \sum_{z_{tj}} p(z_{tj} \,|\, z_{ti}, \mathbf{x}_{t\cdot}, \pi, \theta) \qquad (48) \\ &\propto \pi^{d(ti)}_{z_{\mathrm{Pa}(ti)}}(z_{ti}) p(x_{ti} \,|\, \theta_{z_{ti}}) \prod_{tj \in \mathrm{Ch}(ti)} m^{ti}_{tj}(z_{ti})\,, \qquad (49) \end{aligned}$$

where $m_{tj}^{ti}(z_{ti})$ is a belief propagation (BP) message from node tj to node ti. The computational cost of resampling the assignments for N observed features is thus $\mathcal{O}(NK^2)$.

C.2. Sampling Model Parameters. In the second stage of the truncated sampler, we condition on the assignments $\mathbf{z}$ of observations to hidden states. It is then straightforward to resample the observation distributions θ_k by aggregating statistics of the observations $\{x_{ti} \mid z_{ti} = k\}$ assigned to each state [25, 26]. To resample state–specific transition distributions π_k^d, we first count the number $n^d(k, \ell)$ of transitions from parent state k to child state ℓ, in direction d, instantiated by $\mathbf{z}$. The posterior is then Dirichlet:

$$\pi_k^d \sim \mathcal{D}\left(n^d(k,1) + \alpha\beta_1, \ldots, n^d(k,K) + \alpha\beta_K\right) \tag{50}$$

Finally, the global mixture weights β can be resampled via an auxiliary variable method [26]. We first sample the number of tables assigned to components in the mixtures m_{jk}^d as in the collapsed Gibbs sampler (see appendix A). Then given m, β can be sampled from

$$\{\beta_1, \ldots, \beta_K\} \,|\, m, \gamma) \sim \mathcal{D}\left(m^{\cdot}_{\cdot 1} + \gamma/K, \ldots, m^{\cdot}_{\cdot K} + \gamma/K\right) \tag{51}$$

C.3. Noisy Data. In the first main step of the blocked Gibbs sampler for noisy graphs such as in figure 9 summarized in Algorithm 5, we fix the emission distribution parameters Λ, transition probabilities π and global transitions β, and sample hidden state assignments $\mathbf{z}$ and clean wavelet coefficients $\mathbf{x}$ from their joint distribution

$$p(\mathbf{x}, \mathbf{z} \,|\, \mathbf{w}) = p(\mathbf{z} \,|\, \mathbf{w}) \prod_{ti} p(x_{ti} \,|\, z_{ti}, w_{ti}) \tag{52}$$

To do this, we start by computing the joint assignments of hidden states $p(\mathbf{z} \,|\, \mathbf{w})$ with belief propagation. Local evidence for each node $p(w_{ti} \,|\, z_{ti})$ can be obtained by marginalizing x_{ti}:

$$p(\mathbf{w} \,|\, \mathbf{z}) = \prod_{ti} p(w_{ti} \,|\, z_{ti}) = \prod_{ti} \int_{\mathcal{X}_{ti}} p(w_{ti} \,|\, x_{ti}) p(x_{ti} \,|\, z_{ti}) \, dx_{ti} \tag{53}$$

where $p(w_{ti} \,|\, x_{ti}) = \mathcal{N}(w_{ti}; x_{ti}, \Sigma_n)$ and $p(x_{ti} \,|\, z_{ti}) = \mathcal{N}(x_{ti}; 0, \Lambda_{z_{ti}})$. From the properties of Normal distributions it results that $p(w_{ti} \,|\, z_{ti}) \sim \mathcal{N}\left(0, \Lambda_{z_{ti}} + \Sigma_n\right)$.

Then given sampled hidden state assignments z_{ti}, clean coefficients x_{ti} can be sampled from

$$\begin{aligned} p(x_{ti} \,|\, z_{ti}, w_{ti}) &\propto p(x_{ti} \,|\, z_{ti}) p(w_{ti} \,|\, x_{ti}) \\ &= \mathcal{N}\left(x_{ti}; \left(\Lambda_{z_{ti}}^{-1} + \Sigma_n^{-1}\right)^{-1} \Sigma_n^{-1} w_{ti}, \left(\Lambda_{z_{ti}}^{-1} + \Sigma_n^{-1}\right)^{-1}\right) \end{aligned} \tag{54}$$

In the second main step of the algorithm, we fix the hidden state assignments z_{ti} and clean coefficients x_{ti}, and sample the parameters. This step is identical to that described in the subsection above.

APPENDIX D: LEARNING ALGORITHMS FOR IMAGE DENOISING

D.1. Closed-form image denoising. In the developed denoising algorithms, the denoised image is determined as an inverse wavelet transform (T^{-1}) of observed wavelet coefficients, with detail coefficients replaced by posterior mean estimates of their respective noise-free coefficients, obtained by (Monte Carlo) averaging over sample–specific conditional mean estimates $\mathbb{E}\big[x_i \mid \mathbf{w}_{\backslash 0}, \theta^{(s)}\big]$. Using S samples, we have that

$$y = \mathcal{T}^{-1}\left(\big[\mathbf{w}_0 \,;\, \mathbb{E}\big[x_i \mid \mathbf{w}_{\backslash 0}\big]_{i\in 1}^{I}\big]\right), \tag{55}$$

where $\mathbb{E}\big[x_i \mid \mathbf{w}\big] = \frac{1}{S}\sum_1^S \mathbb{E}\big[x_i \mid \mathbf{w}, \theta^{(s)}\big]$. The sample-specific conditional means can be written as follows:

$$\begin{aligned}
\mathbb{E}\big[x_i \mid \mathbf{w}, \theta^{(s)}\big] &= \int_{\mathcal{X}_i} x_i p(x_i \,|\, \mathbf{w}, \theta^{(s)})\, dx_i && (56)\\
&= \int_{\mathcal{X}_i} x_i \sum_{z_i} p(z_i, x_i \,|\, \mathbf{w}, \theta^{(s)})\, dx_i && (57)\\
&= \int_{\mathcal{X}_i} x_i \sum_{z_i} p(z_i \,|\, \mathbf{w}, \theta^{(s)}) p(x_i \,|\, z_i, \mathbf{w}, \theta^{(s)})\, dx_i && (58)\\
&= \int_{\mathcal{X}_i} x_i \sum_{z_i} p(z_i \,|\, \mathbf{w}, \theta^{(s)}) p(x_i \,|\, z_i, w_i, \theta^{(s)})\, dx_i && (59)\\
&= \sum_{z_i} p(z_i \,|\, \mathbf{w}, \theta^{(s)}) \int_{\mathcal{X}_i} x_i p(x_i \,|\, z_i, w_i, \theta^{(s)})\, dx_i && (60)\\
&= \sum_{k=1}^{K^{(s)}} p(z_i = k \,|\, \mathbf{w}, \theta^{(s)}) \mathbb{E}\big[x_i \mid w_i, \theta^{(s)}, z_i = k\big]. && (61)
\end{aligned}$$

Using the normal equations, we obtain that

$$\begin{aligned}
&\mathbb{E}\big[x_i \mid w_i, \theta^{(s)}, z_i = k\big] = \mathbb{E}\big[x_i \mid \theta^{(s)}, z_i = k\big] + \\
&\mathrm{Cov}[x_i \,|\, \theta^{(s)}, z_i = k] \left(\mathrm{Cov}[w_i \,|\, \theta^{(s)}, z_i = k]\right)^{-1} \left(w_i - \mathbb{E}\big[w_i \mid \theta^{(s)}, z_i = k\big]\right).
\end{aligned} \tag{62}$$

Since $w_i = x_i + n_i$, where $n_i \sim \mathcal{N}(0, \Sigma_n)$, we have that $\mathbb{E}\big[w_i \mid \theta^{(s)}, z_i = k\big] = \mathbb{E}\big[x_i \mid \theta^{(s)}, z_i = k\big] = \mu_k^{(s)}$, and $\mathrm{Cov}[w_i \,|\, \theta^{(s)}, z_i = k] = \mathrm{Cov}[x_i \,|\, \theta^{(s)}, z_i = k] +$

$\Sigma_n = \Lambda_k^{(s)} + \Sigma_n$. Plugging these into (62), we obtain

$$\mathbb{E}\big[x_i \mid w_i, \theta^{(s)}, z_i = k\big] = \mu_k^{(s)} + \Lambda_k^{(s)} \left(\Lambda_k^{(s)} + \Sigma_n\right)^{-1} \left(w_i - \mu_k^{(s)}\right). \tag{63}$$

Combining the results, we can write the state-specific conditional mean estimates as follows:

$$\mathbb{E}\big[x_i \mid \mathbf{w}, \theta^{(s)}\big] = \sum_{k=1}^{K^{(s)}} p(z_i = k \,|\, \mathbf{w}, \theta^{(s)}) \left[\mu_k^{(s)} + \Lambda_k^{(s)} \left(\Lambda_k^{(s)} + \Sigma_n\right)^{-1} \left(w_i - \mu_k^{(s)}\right)\right], \tag{64}$$

where $p(z_i = k \,|\, \mathbf{w}, \theta^{(s)})$ is computed as in Appendix B.2.

Given a noisy image, corrupted by additive white $\mathcal{N}(0, \Sigma_n)$ noise:

1. Apply wavelet transform, obtain scaling coefficients $\mathbf{w}_{t0}$, and detail coefficients $\mathbf{w}_{\setminus t0}$.
2. Learn model parameter posteriors by running a proposed Gibbs sampler on training data $\mathbf{w}_{\setminus t0}$ until burn-in.
3. Collect S samples $\theta^{(s)} = \{\pi_k^{(s)}, \Lambda_k^{(s)}\}_{k=1}^{K_s}$.
4. Estimate posterior hidden state probabilities $p(z_{ti} \mid \mathbf{w}, \theta)$ via the belief propagation algorithm.
5. For each sample, estimate denoised coefficients in closed form:

$$\begin{aligned} \mathbb{E}\big[x_{ti} \mid \mathbf{w}, \theta^{(s)}\big] &= \sum_{k=1}^{K_s} p(z_{ti} = k \mid \mathbf{w}, \theta^{(s)})\, \mathbb{E}\big[x_{ti} \mid w_{ti}, \Lambda_k^{(s)}\big] \\ &= \sum_{k=1}^{K_s} p(z_{ti} = k \mid \mathbf{w}, \theta^{(s)})\, \Lambda_k^{(s)} (\, \Lambda_k^{(s)} + \Sigma_n)^{-1} w_{ti}. \end{aligned}$$

6. Average over samples of varying complexity $\mathbb{E}\big[x_{ti} \mid \mathbf{w}, \theta^{(s)}\big]$ to get posterior mean of detail coefficients.
7. Apply inverse wavelet transform to a combination of observed scaling coefficients w_{t0} with estimated detail coefficients x_{ti}.

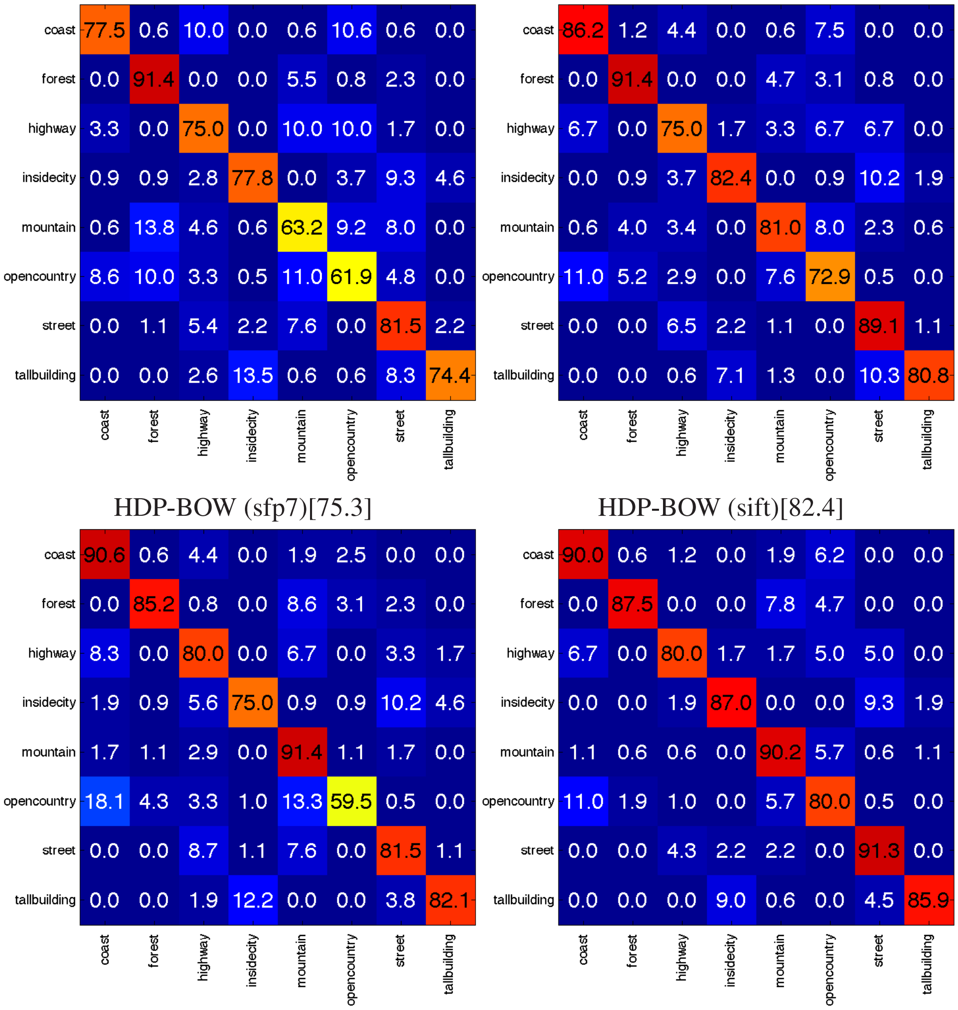
Algorithm 4: The overall learning algorithm for empirical Bayesian image denoising with HDP-HMTs.

Given current state of global mixture weights β, state-specific model parameters and transition distributions $\{\Lambda_k, \pi_k^d\}_{k=1}^K$, hidden state variables z_{ti} and clean wavelet coefficients x_{ti}:

1. Remove the statistics of previous assignments of features to classes:
   - update hidden state transition counts:
     $$n^{d(ti)}_{\setminus ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) \leftarrow n^{d(ti)}(z_{\mathrm{Pa}(ti)}, z_{ti}) - 1;$$
     $$n^{d(ti)}_{\setminus ti}(z_{ti}, z_{\mathrm{Ch}(ti)}) \leftarrow n^{d(ti)}_{\setminus ti}(z_{ti}, z_{\mathrm{Ch}(ti)}) - 1$$
   - update the inverse-Wishart posterior hyperparameters $\{\kappa_{z_{ti}}, \nu_{z_{ti}}, \Delta_{z_{ti}}\}$ to account removal of x_{ti}
2. Sample state assignments using BP with local evidence $p(w_{ti} \mid z_{ti}) = \mathcal{N}(w_{ti}; 0, \Lambda_{z_{ti}} + \Sigma_n)$:
   (a) Compute messages upwards from the leaves up to the roots:
   for leaf nodes:
   $$m_{ti}^{tj}(z_{tj}) \propto \sum_{z_{ti}} \pi_{z_{tj}}^{d(ti)}(z_{ti}) p(w_{ti}|z_{ti})$$
   for non-leaf nodes:
   $$m_{ti}^{tj}(z_{tj}) \propto \sum_{z_{ti}} \pi_{z_{tj}}^{d(ti)}(z_{ti}) p(w_{ti}|z_{ti}) \prod_{tk \in \mathrm{Ch}(ti)} m_{tk}^{ti}(z_{ti})$$
   (b) Sample hidden states while traversing downwards:
   for non-leaf nodes:
   $$p(z_{ti} \mid z_{\mathrm{Pa}(ti)}, \mathbf{w}_{t\cdot}, \pi, \Lambda, \Sigma) \propto \pi_{z_{\mathrm{Pa}(ti)}}^{d(ti)}(z_{ti}) p(w_{ti}|z_{ti}) \prod_{tj \in \mathrm{Ch}(ti)} m_{tj}^{ti}(z_{ti})$$
   for leaf nodes:
   $$p(z_{ti} \mid z_{\mathrm{Pa}(ti)}, \mathbf{w}_{t\cdot}, \pi, \Lambda, \Sigma) \propto \pi_{z_{\mathrm{Pa}(ti)}}^{d(ti)}(z_{ti}) p(w_{ti}|z_{ti})$$
3. Sample clean wavelet coefficients by drawing a random vector from
   $$p(x_{ti} \mid z_{ti}, w_{ti}) \propto \mathcal{N}\left(x_{ti}; \left(\Lambda_{z_{ti}}^{-1} + \Sigma_n^{-1}\right)^{-1} \Sigma_n^{-1} w_{ti}, \left(\Lambda_{z_{ti}}^{-1} + \Sigma_n^{-1}\right)^{-1}\right)$$
4. Add the statistics of new assignments of features x_{ti} to classes z_{ti}:
   - update hidden state transition counts:
     $$n^{d(ti)}(z_{\mathrm{Pa}(ti)}, z_{ti}) \leftarrow n^{d(ti)}_{\setminus ti}(z_{\mathrm{Pa}(ti)}, z_{ti}) + 1;$$
     $$n^{d(ti)}(z_{ti}, z_{\mathrm{Ch}(ti)}) \leftarrow n^{d(ti)}(z_{ti}, z_{\mathrm{Ch}(ti)}) + 1$$
   - update the inverse-Wishart posterior hyperparameters $\{\kappa_{z_{ti}}, \nu_{z_{ti}}, \Delta_{z_{ti}}\}$ to account addition of x_{ti}
5. Sample model parameters $\{\Lambda_k, \pi_k\}_{k=1}^K$:
   (a) Sample direction-specific transition distributions by drawing a random Dirichlet-vector from
   $$\pi_k^d \sim \mathcal{D}\left(n^d(k,1) + \alpha\beta_1, \ldots, n^d(k,K) + \alpha\beta_K\right)$$
   (b) Sample Λ_k's by drawing a random Inverse-Wishart vector from
   $$p(\Lambda_k | \mathbf{x}, \mathbf{z}, H) \propto p(\Lambda_k \mid H) \prod_{j:z_j=k} p(x_j \mid \Lambda_k)$$
6. Global mixture weights β can be resampled via an auxiliary variable method [26].

Algorithm 5: Blocked Gibbs sampler for truncated HDP-HMTs when data is corrupted by additive $\mathcal{N}(0, \Sigma_n)$ noise.

APPENDIX E: SUPPLEMENTARY INFORMATION



	coast	forest	highway	insidecity	mountain	opencountry	street	tallbuilding
coast	77.5	0.6	10.0	0.0	0.6	10.6	0.6	0.0
forest	0.0	91.4	0.0	0.0	5.5	0.8	2.3	0.0
highway	3.3	0.0	75.0	0.0	10.0	10.0	1.7	0.0
insidecity	0.9	0.9	2.8	77.8	0.0	3.7	9.3	4.6
mountain	0.6	13.8	4.6	0.6	63.2	9.2	8.0	0.0
opencountry	8.6	10.0	3.3	0.5	11.0	61.9	4.8	0.0
street	0.0	1.1	5.4	2.2	7.6	0.0	81.5	2.2
tallbuilding	0.0	0.0	2.6	13.5	0.6	0.6	8.3	74.4

HDP-BOW (sfp7)[75.3]

	coast	forest	highway	insidecity	mountain	opencountry	street	tallbuilding
coast	86.2	1.2	4.4	0.0	0.6	7.5	0.0	0.0
forest	0.0	91.4	0.0	0.0	4.7	3.1	0.8	0.0
highway	6.7	0.0	75.0	1.7	3.3	6.7	6.7	0.0
insidecity	0.0	0.9	3.7	82.4	0.0	0.9	10.2	1.9
mountain	0.6	4.0	3.4	0.0	81.0	8.0	2.3	0.6
opencountry	11.0	5.2	2.9	0.0	7.6	72.9	0.5	0.0
street	0.0	0.0	6.5	2.2	1.1	0.0	89.1	1.1
tallbuilding	0.0	0.0	0.6	7.1	1.3	0.0	10.3	80.8

HDP-BOW (sift)[82.4]

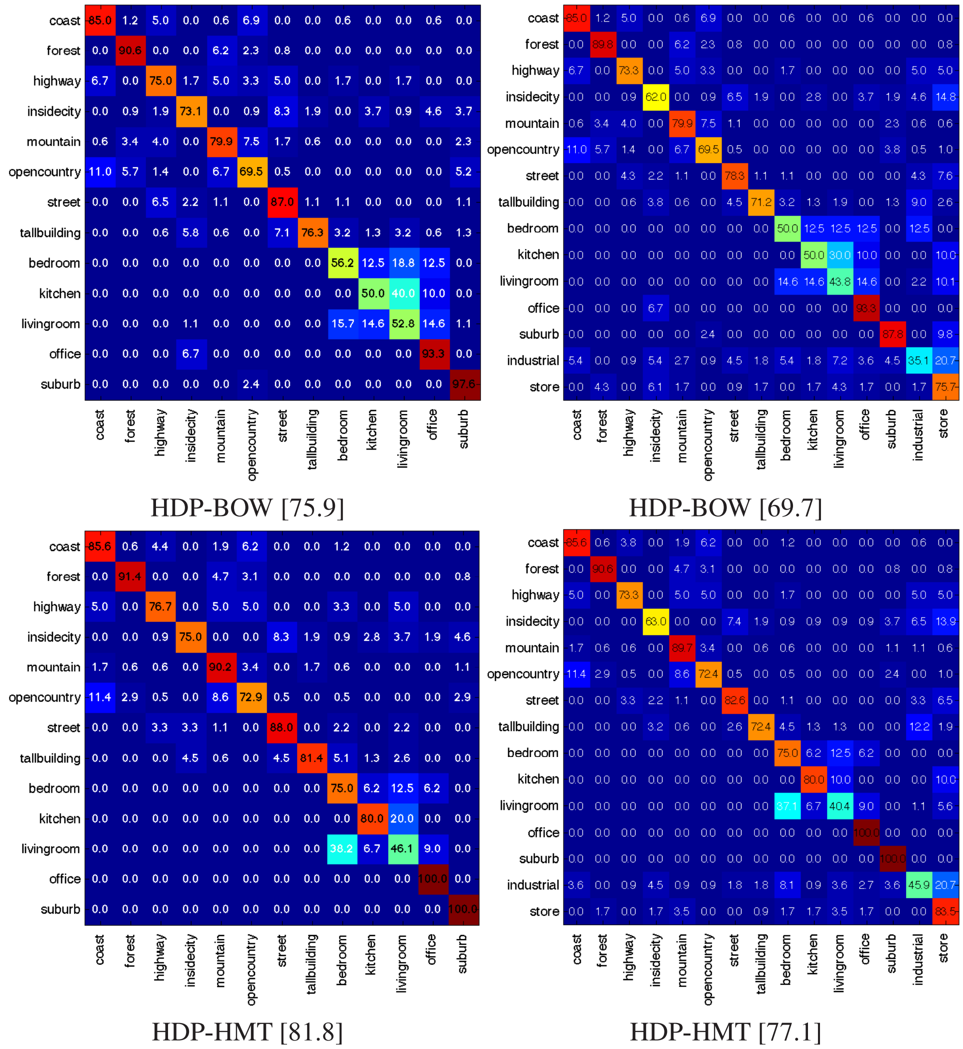
	coast	forest	highway	insidecity	mountain	opencountry	street	tallbuilding
coast	90.6	0.6	4.4	0.0	1.9	2.5	0.0	0.0
forest	0.0	85.2	0.8	0.0	8.6	3.1	2.3	0.0
highway	8.3	0.0	80.0	0.0	6.7	0.0	3.3	1.7
insidecity	1.9	0.9	5.6	75.0	0.9	0.9	10.2	4.6
mountain	1.7	1.1	2.9	0.0	91.4	1.1	1.7	0.0
opencountry	18.1	4.3	3.3	1.0	13.3	59.5	0.5	0.0
street	0.0	0.0	8.7	1.1	7.6	0.0	81.5	1.1
tallbuilding	0.0	0.0	1.9	12.2	0.0	0.0	3.8	82.1

HDP-HMT (sfp7)[80.7]

	coast	forest	highway	insidecity	mountain	opencountry	street	tallbuilding
coast	90.0	0.6	1.2	0.0	1.9	6.2	0.0	0.0
forest	0.0	87.5	0.0	0.0	7.8	4.7	0.0	0.0
highway	6.7	0.0	80.0	1.7	1.7	5.0	5.0	0.0
insidecity	0.0	0.0	1.9	87.0	0.0	0.0	9.3	1.9
mountain	1.1	0.6	0.6	0.0	90.2	5.7	0.6	1.1
opencountry	11.0	1.9	1.0	0.0	5.7	80.0	0.5	0.0
street	0.0	0.0	4.3	2.2	2.2	0.0	91.3	0.0
tallbuilding	0.0	0.0	0.0	9.0	0.6	0.0	4.5	85.9

HDP-HMT (sift)[86.5]

FIG 21. *Confusion matrices for the 8 scenes category dataset [16] using HDP-BOW (top row) and the HDP-HMT (bottom row) models in wavelet (left column) and SIFT domain (right column). Average performance across all categories is shown in parentheses.*



HDP-BOW [75.9]

HDP-BOW [69.7]

HDP-HMT [81.8]

HDP-HMT [77.1]

FIG 22. *Confusion matrices for the 13 scenes category dataset [7] (left) and 15 scenes category dataset [13](right), using HDP-BOW (top row) and HDP-HMT (bottom row) models in SIFT domain. Average performance across all categories is shown in parentheses.*

σ	*Mandrill*		*Peppers*	
5	35.44 / 0.960	35.52 / 0.962	37.52 / 0.950	37.53 / 0.950
10	30.84 / 0.896	30.90 / 0.900	33.99 / 0.914	33.92 / 0.912
15	28.56 / 0.835	28.58 / 0.837	32.02 / 0.885	31.91 / 0.883
25	26.16 / 0.730	26.10 / 0.724	29.64 / 0.842	29.47 / 0.836
50	23.39 / 0.540	23.57 / 0.538	26.14 / 0.729	26.18 / 0.745
75	22.14 / 0.435	22.54 / 0.449	23.58 / 0.586	24.37 / 0.678
100	21.61 / 0.379	21.95 / 0.402	22.13 / 0.511	23.21 / 0.635
125	21.27 / 0.348	21.50 / 0.370	21.08 / 0.463	22.36 / 0.610
	Emp. Bayes	Transfer	Emp. Bayes	Transfer

σ	*Cameraman*		*Einstein*		*House*	
5	37.50 / 0.954	37.56 / 0.954	38.27 / 0.943	38.23 / 0.943	38.82 / 0.945	38.68 / 0.946
10	33.29 / 0.914	33.30 / 0.915	34.91 / 0.892	34.84 / 0.890	35.47 / 0.895	35.35 / 0.897
15	31.05 / 0.879	30.99 / 0.879	33.18 / 0.855	33.06 / 0.851	33.80 / 0.867	33.43 / 0.865
25	28.45 / 0.823	28.34 / 0.821	30.94 / 0.795	30.91 / 0.795	31.67 / 0.832	31.21 / 0.827
50	25.25 / 0.704	25.30 / 0.730	27.65 / 0.663	28.26 / 0.712	28.40 / 0.753	28.23 / 0.761
75	23.07 / 0.569	23.50 / 0.669	25.64 / 0.564	26.80 / 0.664	25.56 / 0.611	26.49 / 0.716
100	21.67 / 0.486	22.40 / 0.627	23.65 / 0.443	25.84 / 0.635	24.34 / 0.560	25.32 / 0.692
125	20.65 / 0.438	21.57 / 0.593	24.30 / 0.514	25.01 / 0.609	21.87 / 0.406	24.45 / 0.668
	Emp. Bayes	Transfer	Emp. Bayes	Transfer	Emp. Bayes	Transfer

TABLE 2

Peak signal-to-noise ratio (PSNR) and mean structural similarity (SSIM) of a set of denoised standard images of size 256×256 with the HDP-HMT using the Empirical Bayesian (Emp. Bayes) and the transfer denoising (Transfer) approach.

σ	*Barbara*		*Boat*		*Couple*	
5	37.88 / 0.962	37.67 / 0.960	37.19 / 0.940	36.80 / 0.931	37.21 / 0.949	37.16 / 0.949
10	34.13 / 0.932	33.63 / 0.925	33.62 / 0.887	33.37 / 0.878	33.52 / 0.901	33.41 / 0.898
15	32.00 / 0.904	31.29 / 0.891	31.68 / 0.847	31.49 / 0.840	31.54 / 0.861	31.39 / 0.856
25	29.38 / 0.850	28.35 / 0.825	29.32 / 0.785	29.21 / 0.780	29.14 / 0.797	29.05 / 0.794
50	25.60 / 0.694	24.69 / 0.692	26.19 / 0.658	26.36 / 0.685	25.81 / 0.650	26.12 / 0.686
75	23.50 / 0.581	23.18 / 0.614	25.00 / 0.576	24.77 / 0.623	24.31 / 0.574	24.63 / 0.619
100	22.40 / 0.524	22.32 / 0.563	23.68 / 0.549	23.79 / 0.586	23.45 / 0.534	23.69 / 0.574
125	21.70 / 0.491	21.77 / 0.535	22.84 / 0.513	23.06 / 0.555	22.74 / 0.496	22.98 / 0.539
	Emp. Bayes	Transfer	Emp. Bayes	Transfer	Emp. Bayes	Transfer
σ	*Hill*		*Lena*		*Man*	
5	37.00 / 0.942	36.97 / 0.941	38.65 / 0.945	38.51 / 0.943	37.47 / 0.950	37.43 / 0.950
10	33.36 / 0.881	33.31 / 0.878	35.67 / 0.913	35.42 / 0.909	33.60 / 0.900	33.53 / 0.898
15	31.53 / 0.833	31.45 / 0.828	33.96 / 0.889	33.75 / 0.885	31.57 / 0.857	31.50 / 0.855
25	29.40 / 0.760	29.37 / 0.757	31.71 / 0.845	31.51 / 0.848	29.20 / 0.784	29.17 / 0.786
50	26.46 / 0.621	26.94 / 0.661	27.71 / 0.692	28.56 / 0.784	26.28 / 0.656	26.46 / 0.684
75	25.39 / 0.574	25.60 / 0.602	26.65 / 0.683	26.80 / 0.735	24.79 / 0.585	25.02 / 0.624
100	24.52 / 0.533	24.74 / 0.568	25.49 / 0.650	25.79 / 0.701	23.81 / 0.540	24.07 / 0.584
125	23.87 / 0.504	24.09 / 0.541	24.51 / 0.613	24.82 / 0.681	23.10 / 0.508	23.34 / 0.555
	Emp. Bayes	Transfer	Emp. Bayes	Transfer	Emp. Bayes	Transfer

TABLE 3

Peak signal-to-noise ratio (PSNR) and mean structural similarity (SSIM) of a set of denoised standard images of size 512×512 with the HDP-HMT using the Empirical Bayesian (Emp. Bayes) and the transfer denoising (Transfer) approach.

UNIVERSITY OF EDINBURGH
SCHOOL OF INFORMATICS
E-MAIL: J.J.Kivinen@sms.ed.ac.uk

BROWN UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
E-MAIL: sudderth@cs.brown.edu

UNIVERSITY OF CALIFORNIA, BERKELEY
COMPUTER SCIENCE DIVISION &
DEPARTMENT OF STATISTICS
E-MAIL: jordan@cs.berkeley.edu