implies that a new child/level must be added, since the first \( L - 1 \) sticks are not enough to capture how the data are clustered. If stick \( L \) is selected, then a new node/level is constituted (a new child is added), by drawing a new \( V_L \sim \text{Beta}(1, \gamma) \), and then \( V_{L+1} = 1 \), thereby now constituting an \((L + 1)\)-dimensional stick representation; the associated node-dependent statistics are constituted as discussed in Section 2 (drawing new probabilities over dictionary elements). The model therefore infers “in retrospect” that the \( L \)-level truncation was too small, and expands adaptively. The model also has the ability to shrink the number of sticks used at any component of the model, if less than the associated truncated level is needed to define the number of children/levels are actually utilized.

3.5 Generative Process

The generative process for the model is summarized as follows:

1. Draw dictionary \( D \sim \prod_{k=1}^{K} \mathcal{N}(0, \frac{1}{p}I_p) \)
2. Draw \( \gamma, \alpha, \gamma_c, \) and \( \gamma_s \) from respective gamma distributions
3. For each image \( m \in \{1, 2, \ldots, M\} \)
   (a) Draw \( c_m \sim \text{nCRP}(\gamma) \)
   (b) For each newly utilized node \( \epsilon \) in the tree, draw dictionary usage probabilities \( \pi_{\epsilon} \sim \prod_{k=1}^{K} \text{Beta}(a_0/K, b_0(K - 1)/K) \)
   (c) Draw \( \theta_m \sim \text{Stick}(\alpha) \)
   (d) For the \( l \)-th level or feature vector
      i. Draw level index \( l_{mi} \sim \sum_{l=1}^{\infty} \theta_m^l\delta_l \), which along with \( c_m \) defines node \( h_{mi} \)
      ii. Draw \( z_{mi} \sim \prod_{k=1}^{K} \text{Bernoulli}(\pi_{h_{mk}}) \), and \( s_{mi} \sim \mathcal{N}_+(0, \sigma_s^{-1}I_K) \)
      iii. Draw \( x_{mi} \sim \mathcal{N}(D(z_{mi} \odot s_{mi}), \gamma_{\epsilon}^{-1}I_p) \)
4. For each unique tree path \( p \), draw \( \psi_p \sim \text{Dir}(\frac{\alpha}{\sum_{k=1}^{p}}, \ldots, \frac{\alpha}{\sum_{k=1}^{p}}) \)
5. If annotations are available for image \( m \), \( y_m \sim \text{Mult}(\alpha(y_m), \psi_{c_m}) \), where \( \alpha(y_m) \) is the total number of words in \( y_m \)

In Step 3(b), new nodes (topics) are added “in retrospect”, as discussed in the previous subsection (nodes may also be pruned with this sampler). After completing Step 3, the tree size is constituted, which allows Step 4, imposition of a distribution over words for each path.

4 Model Inference

A contribution of this paper concerns use of retropective sampling to infer the tree width and depth. To save space for an extensive set of experimental results, we here only discuss updates associated with inferring the tree depth. A complete set of update equations are provided in Supplementary Material, where one may also find a summary of all notation.

To sample \( l_{mi} \) from the conditional posterior, we first need to specify the likelihood that \( \{\epsilon \in c_m\} \):

\[
p_l(z_{mi}|\pi_{\epsilon}, c_m) = \prod_{k=1}^{K} \pi_{\epsilon k}^{z_{mi k}} (1 - \pi_{\epsilon k})^{1-z_{mi k}}
\]

and the prior distribution, which is specified by a stick-breaking draw \( \theta_m \) for each image \( m \). Although \( l_{mi} \) can be sampled from a closed form posterior for a fixed \( L_m \), here to learn \( L_m \) adaptively we instead use an Metropolis-Hastings step, where the proposal distribution is defined as

\[
q(l_{mi} = j) \propto \left\{ \begin{array}{ll}
\theta_m z_{mi} | \pi_{j}, c_m, & j \leq L_m \\
\theta_m M_m(L_m), & j > L_m
\end{array} \right.
\]

where \( M_m(L_m) = \max_{1 \leq |\epsilon| \leq L_m} \{p_l(z_{mi} | \pi_{\epsilon}, c_m)\} \). Note that the sampled value of \( l_{mi} \) is allowed to be larger than the truncation level \( L_m \), consequently \( l_{mi} \) and the depth of the tree is learned adaptively. The acceptance probability \( \kappa_{mi}(j) \) for \( l_{mi} = j \) is

\[
\left\{ \begin{array}{ll}
1, & j \leq L_m \& L_m = L_m' \\
\min\{1, \frac{\hat{c}_{mi}(L_m)M_m(L_m')}{\hat{c}_{mi}(L_m')M_m(L_m)}\}, & j \leq L_m \& L_m' < L_m \\
\min\{1, \frac{\hat{c}_{mi}(L_m)M_m(L_m' \& L_m)}{\hat{c}_{mi}(L_m')M_m(L_m)}\}, & j > L_m 
\end{array} \right.
\]

where the normalizing constant is defined as

\[
\hat{c}_{mi}(L_m) = \sum_{|\epsilon|=1}^{L_m} \theta_m |\epsilon| p(z_{mi} | \pi_{\epsilon}, c_m) + M_m(L_m)(1-
\]

472