The LORACs Prior for VAEs: Letting the Trees Speak for the Data

Sharad Vikram
U.C. San Diego[1]

Matthew D. Hoffman
Google AI

Matthew J. Johnson
Google Brain

Abstract

In variational autoencoders, the prior on the latent codes z is often treated as an afterthought, but the prior shapes the kind of latent representation that the model learns. If the goal is to learn a representation that is interpretable and useful, then the prior should reflect the ways in which the high-level factors that describe the data vary. The "default" prior is an isotropic normal, but if the natural factors of variation in the dataset exhibit discrete structure or are not independent, then the isotropic-normal prior will actually encourage learning representations that *mask* this structure. To alleviate this problem, we propose using a flexible Bayesian nonparametric hierarchical clustering prior based on the time-marginalized coalescent (TMC). To scale learning to large datasets, we develop a new inducing-point approximation and inference algorithm. We then apply the method without supervision to several datasets and examine the interpretability and practical performance of the inferred hierarchies and learned latent space.

1 Introduction

Variational autoencoders (VAEs; Kingma and Welling, 2014; Rezende et al., 2014) are a popular class of deep latent-variable models. The VAE assumes that observations x are generated by first sampling a latent vector z from some tractable prior $p(z)$, and then sampling x from some tractable distribution $p(x \mid g_\theta(z))$. For example, $g_\theta(z)$ could be a neural network with weights θ and $p(x \mid g_\theta(z))$ might be a Gaussian with mean $g_\theta(z)$.

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

VAEs, like other unsupervised latent-variable models (e.g.; Tipping and Bishop, 1999; Blei et al., 2003), can uncover latent structure in datasets. In particular, one might hope that high-level characteristics of the data are encoded more directly in the geometry of the latent space z than they are in the data space x. For example, when modeling faces one might hope that one latent dimension corresponds to pose, another to hair length, another to gender, etc.

What kind of latent structure will the VAE actually discover? Hoffman and Johnson (2016) observe that the ELBO encourages the model to make the statistics of the population of encoded z vectors resemble those of the prior, so that $p(z) \approx \mathbb{E}_{\text{population}}[p(z \mid x)]$. The prior $p(z)$ therefore plays an important role in shaping the geometry of the latent space. For example, if we use the "default" prior $p(z) = \mathcal{N}(z; 0, I)$, then we are asking the model to explain the data in terms of smoothly varying, completely independent factors (Burgess et al., 2018). These constraints may sometimes be reasonable—for example, geometric factors such as pose or lighting angle may be nearly independent and rotationally symmetric. But some natural factors exhibit dependence structure (for example, facial hair length and gender are strongly correlated), and others may have nonsmooth structure (for example, handwritten characters naturally cluster into discrete groups).

In this paper, we propose using a more opinionated prior on the VAE's latent vectors: the time-marginalized coalescent (TMC; Boyles and Welling, 2012). The TMC is a powerful, interpretable Bayesian nonparametric hierarchical clustering model that can encode rich discrete and continuous structure. Combining the TMC with the VAE combines the strengths of Bayesian nonparametrics (interpretable, discrete structure learning) and deep generative modeling (freedom from restrictive distributional assumptions).

Our contributions are:

- We propose a deep Bayesian nonparametric model that can discover hierarchical cluster structure in complex, high-dimensional datasets.

- We develop a minibatch-friendly inference procedure for fitting TMCs based on an inducing-point approximation, which scales to arbitrarily large datasets.
- We show that our model's learned latent representations consistently outperform those learned by other variational (and classical) autoencoders when evaluated on downstream classification and retrieval tasks.

2 Background

2.1 Bayesian priors for hierarchical clustering

Hierarchical clustering is a flexible tool in exploratory data analysis as trees offer visual, interpretable summaries of data. Typically, algorithms for hierarchical clustering are either agglomerative (where data are recursively, greedily merged to form a tree from the bottom-up) or divisive (where data are recursively partitioned, forming a tree from the top-down). Bayesian nonparametric hierarchical clustering (BNHC) additionally incorporates uncertainty over tree structure by introducing a prior distribution over trees $r(\tau)$ and a likelihood model for data $r(z_{1:N} \mid \tau)$, with the goal of sampling the posterior distribution $r(\tau \mid z_{1:N})$.[2]

In this paper, we focus on rooted binary trees with N labeled leaves adorned with branch lengths, called *phylogenies*. Prior distributions over phylogenies often take the form of a stochastic generative process in which a tree is built with random merges, as in the Kingman coalescent (Kingman, 1982), or random splits, as in the Dirichlet diffusion tree (Neal, 2003). These nonparametric distributions have helpful properties, such as exchangeability, which enable efficient Bayesian inference. In this paper, we focus on the time-marginalized coalescent (TMC; Boyles and Welling, 2012), which decouples the distribution over tree structure and branch length, a property that helps simplify inference down the line.

2.1.1 Time-marginalized coalescent (TMC)

The time-marginalized coalescent defines a prior distribution over phylogenies. A phylogeny $\tau = (V, E, T)$ is a directed rooted full binary tree, with vertex set V and edges E, together with time labels $T : V \to [0, 1]$ where we denote $t_v = T(v)$. The vertex set V is partitioned into N leaf vertices V_{leaf} and $N - 1$ internal vertices V_{int}, so that $V = V_{\text{int}} \cup V_{\text{leaf}}$, and we take $V_{\text{leaf}} = \{1, 2, \ldots, N\}$ to simplify notation for identifying leaves with N data points. The directed edges of the tree are encoded in the edge set $E \subset V_{\text{int}} \times V$, where we denote the root vertex as v_{root} and for $v \in V \setminus \{v_{\text{root}}\}$ we denote the parent of v as $\pi(v) = w$ where $(w, v) \in E$.

The TMC samples a random tree structure (V, E) by a stochastic process in which the N leaves are recursively merged uniformly at random until only one vertex is left. This process yields the probability mass function on valid (V, E) pairs given by

$$r(V, E) = \frac{(N-1)!}{\prod_{v \in V_{\text{int}}} c(v)} \prod_{i=1}^{N-1} \binom{i+1}{2}^{-1}, \qquad (1)$$

where $c(v)$ denotes the number of internal vertices in the subtree rooted at v. Given the tree structure, time labels are generated via the stick-breaking process

$$t_v = \begin{cases} 0 & v = v_{\text{root}}, \\ 1 & v \in V_{\text{leaf}}, \\ t_{\pi(v)} - \beta_v (1 - t_{\pi(v)}) & v \in V_{\text{int}} \setminus \{v_{\text{root}}\}, \end{cases} \qquad (2)$$

where $\beta_v \overset{\text{iid}}{\sim} \text{Beta}(a, b)$ for $v \in V$. These time labels encode a branch length $t_v - t_{\pi(v)}$ for each edge $e = (\pi(v), v) \in E$. We denote the overall density on phylogenies with N leaves as $\text{TMC}_N(\tau; a, b)$.

Finally, to connect the TMC prior to data in $\mathbb{R}^d$, we define a likelihood model $r(z_{1:N} \mid \tau)$ on N data points, with z_n corresponding to the leaf vertex $n \in V_{\text{leaf}}$. We use a Gaussian random walk (GRW), where for each vertex $v \in V$ a location $z_v \mid z_{\pi(v)}$ is sampled according to a Gaussian distribution centered at its parent's location with variance equal to the branch length,

$$z_v \mid z_{\pi(v)} \sim \mathcal{N}(z_{\pi(v)}, (t_v - t_{\pi(v)})I), \quad v \in V \setminus \{v_{\text{root}}\},$$

and we take $z_{v_{\text{root}}} \sim \mathcal{N}(0, I)$. As a result of this choice, we can exploit the Gaussian graphical model structure to efficiently marginalize out the internal locations z_v associated with internal vertices $v \in V_{\text{int}}$ and evaluate the resulting marginal density $r(z_{1:N} \mid \tau)$. For details about this marginalization, please refer to Appendix C. The final overall density is written as

$$r(z_{1:N}, \tau) = \text{TMC}_N(\tau; a, b) r(z_{1:N} \mid \tau). \qquad (3)$$

For further details and derivations related to the TMC, please refer to Boyles and Welling (2012).

2.1.2 TMC posterior predictive density

The TMC with N leaves and a GRW likelihood model can be a prior on a set of N hierarchically-structured data, i.e. data that correspond to nodes with small tree distance should have similar location values. In addition, it also acts as a density from which we can

[2] We use r to denote probability distributions relating to the TMC and distinguish from p and q distributions used later in the paper.

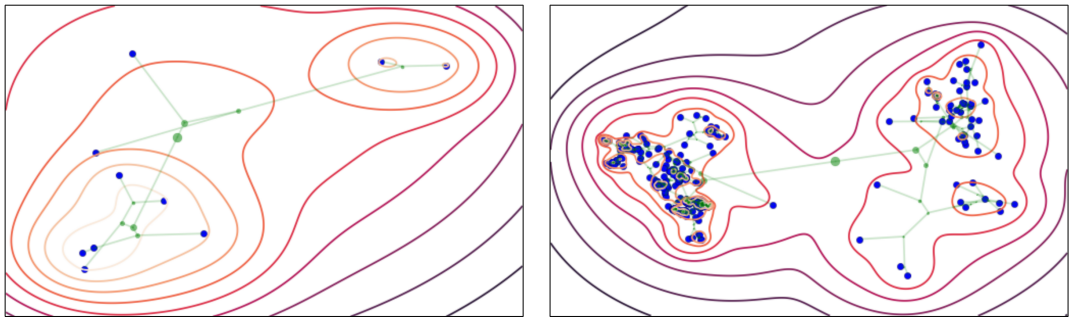


Figure 1: Independent samples from a time-marginalized coalescent (TMC) prior and two-dimensional Gaussian random walk likelihood model (10 and 300 leaves respectively). Contours in the plots correspond to posterior predictive density $r(z_{N+1} \mid z_{1:N}, \tau)$. As the number of leaves grow, the predictive density grows more complex.

sample new data. The posterior predictive density $r(z_{N+1} \mid z_{1:N}, \tau)$ is easy to sample thanks to the exchangeability of the TMC.

To sample a new data point z_{N+1}, we select a branch (edge) and a time to attach a new leaf node. The probability $r(e_{N+1} \mid V, E)$ of selecting branch e_{N+1} is proportional to the probability under the TMC prior of the tree with a new leaf attached to branch e_{N+1}. The density $r(t_{N+1} \mid e_{N+1}, V, E)$ for a time label t_{N+1} is determined by the stick-breaking process (see Appendix C for details). Both of these probabilities are easy to calculate and sample due to the exchangeability of the TMC.

The new location z_{N+1} can be sampled from $r(z_{N+1} \mid e_{N+1}, t_{N+1}, \tau)$, which is the Gaussian distribution that comes out of the GRW likelihood model. Pictured in Figure 1 are samples from a TMC prior and GRW likelihood, where contours correspond to $r(z_{N+1} \mid z_{1:N}, \tau)$. In addition to modeling hierarchical structure, the TMC is a flexible nonparametric density estimator.

2.1.3 TMC inference

The posterior distribution $r(\tau \mid z_{1:N})$ is analytically intractable due to the normalization constant $r(z_{1:N})$ involving a sum over all tree structures, but it can be approximately sampled via Markov chain Monte-Carlo (MCMC) methods. We utilize the Metropolis-Hastings algorithm with a subtree-prune-and-regraft (SPR) proposal distribution (Neal, 2003). An SPR proposal picks a subtree uniformly at random from τ and detaches it. It is then attached back on the tree to a branch and time picked uniformly at random. The Metropolis-Hastings acceptance probability is efficient to compute because the joint density $r(\tau, z_{1:N})$ can be evaluated using belief propagation to marginalize the latent values at internal nodes of τ, and many of the messages can be cached. See Appendix C for details.

2.2 Variational autoencoder

The variational autoencoder (VAE) is a generative model for a dataset $x_{1:N}$ wherein latent vectors $z_{1:N}$ are sampled from a prior distribution and then individually passed into a neural network observation model with parameters θ,

$$z_{1:N} \sim p(z_{1:N}), \qquad x_n \mid z_n \sim p_\theta(x_n \mid z_n), \tag{4}$$

We are interested in the posterior distribution $p(z_n \mid x_n)$, which is not analytically tractable but can be approximated with a variational distribution $q_\phi(z_n \mid x_n)$, typically a neural network that outputs parameters of a Gaussian distribution. The weights of the approximate posterior can be learned by optimizing the evidence-lower bound (ELBO),

$$\mathcal{L}[q] \triangleq \mathbb{E}_q \left[\log \frac{p_\theta(x_{1:N}, z_{1:N})}{\prod_n q_\phi(z_n \mid x_n)} \right] \tag{5}$$

The parameters of the model, θ and ϕ, are learned via stochastic gradient ascent on the ELBO, using the reparametrization trick for lower variance gradients (Kingma and Welling, 2014; Rezende et al., 2014).

3 The TMC-VAE

The choice of prior distribution in the VAE significantly affects the autoencoder and resulting latent space. The default standard normal prior, which takes $z_n \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$, acts as a regularizer on an otherwise unconstrained autoencoder, but can be restrictive and result in overpruning (Burda et al., 2015). Extremely flexible, learnable distributions like masked autoregressive flow (MAF) priors (Papamakarios et al., 2017) enable very rich latent spaces, but don't encode any interpretable bias for organizing the latent space (except perhaps smoothness).

In this paper, we explore the TMC prior for the VAE, which could potentially strike a sweet spot between

restrictive and flexible priors. We generate the latent values $z_{1:N}$ of a VAE according to the TMC prior, then generate observations $x_{1:N}$ using a neural network observation model,

$$\tau \sim \mathrm{TMC}_N(\tau; a, b), \tag{6}$$

$$z_{1:N} \mid \tau \sim r(z_{1:N} \mid \tau), \qquad x_n \mid z_n \sim p_\theta(x_n \mid z_n). \tag{7}$$

The TMC-VAE is a coherent generative process that captures discrete, interpretable structure in the latent space. A phylogeny not only has an intuitive inductive bias, but can be useful for exploratory data analysis and introspecting the latent space itself.

Consider doing inference in this model: first assume variational distributions $q_\phi(z_n \mid x_n)$ (as in the VAE) and $q(\tau)$, which results in the ELBO,

$$\mathcal{L}[q] = \mathbb{E}_q \left[\log \frac{p(\tau, z_{1:N}, x_{1:N})}{q(\tau) \prod_n q(z_n \mid x_n)} \right] \tag{8}$$

For fixed $q_\phi(z_n \mid x_n)$, we can sample the optimal $q^*(\tau)$,

$$q^*(\tau) \propto \exp\{\mathbb{E}_q \left[\log p(\tau, z_{1:N}, x_{1:N})\right]\} \tag{9}$$

Because $p(z_{1:N} \mid \tau)$ is jointly Gaussian (factorizing according to tree structure) and $q_\phi(z_n \mid x_n)$ is Gaussian, expectations with respect to $z_{1:N}$ can move into $\log p(\tau, z_{1:N}, x_{1:N})$. This enables sampling the expected joint likelihood $\mathbb{E}_q \left[\log p(\tau, z_{1:N})\right]$ using SPR Metropolis-Hastings. However, optimizing this ELBO is problematic. $p(z_{1:N}|\tau)$ does not factorize independently, so computing unbiased gradient estimates from minibatches is impossible and requires evaluating all the data. Furthermore, the TMC is limiting from a computational perspective. Since a phylogeny has as many leaves as points in the dataset, belief propagation over internal nodes of the tree slows down linearly as the size of the dataset grows. In addition, SPR proposals mix very slowly for large trees. We found these limitations make the model impractical for datasets of more than 1000 examples.

In the next section, we address these computational issues, while retaining the interesting properties of the TMC-VAE.

4 LORACs prior for VAEs

In this section, we introduce a novel approximation to the TMC prior, which preserves many desirable properties like structure and interpretability, while being computationally viable. Our key idea is to use a set of learned *inducing points* as the leaves of the tree in the latent space, analogous to inducing-input approximations for Gaussian processes (Snelson and Ghahramani, 2006). In this model, latent vectors $z_{1:N}$ are not directly hierarchically clustered, but are rather independent samples from the induced posterior predictive density of a TMC. We call this the Latent ORganization of Arboreal Clusters (*LORACs*, pronounced "lorax") prior.

To define the LORACs prior $p(\tau, z_{1:N})$, we first define an auxiliary TMC distribution $r(\tau, s_{1:M})$ with M leaf locations $s_{1:M}$. We treat $s_{1:M}$ as a set of learnable free parameters, and define the conditional $r(\tau \mid s_{1:M})$ as the LORACs prior on phylogenies τ:

$$p(\tau; s_{1:M}) \triangleq r(\tau \mid s_{1:M}). \tag{10}$$

That is, we choose the prior on phylogenies τ to be the posterior distribution of a TMC with pseudo-observations $s_{1:M}$. Next, we define the LORACs prior on locations $z_n \mid \tau$ as a conditionally independent draw from the predictive distribution $r(s_{M+1} \mid \tau, s_{1:M})$, writing the sampled attachment branch and time as e_n and t_n, respectively:

$$\begin{aligned} p(e_n, t_n \mid \tau) &\triangleq r(e_{M+1} = e_n, t_{M+1} = t_n \mid \tau), \\ p(z_n \mid e_n, t_n, \tau) &\triangleq r(s_{M+1} = z_n \mid e_n, t_n, \tau, s_{1:M}). \end{aligned} \tag{11}$$

To complete the model, we use an observation likelihood parameterized by a neural network, writing

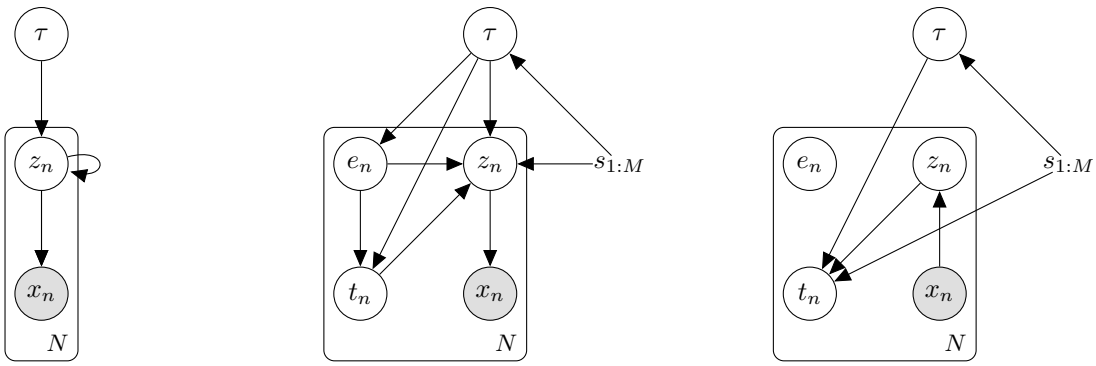
$$x_n \mid z_n \sim p_\theta(x_n \mid z_n). \tag{12}$$

By using the learned inducing points $s_{1:M}$, we avoid the main difficulty of inference in the TMC-VAE of Section 3, namely the need to do inference over all N points in the dataset. Instead, dependence between datapoints is mediated by the set of inducing points $s_{1:M}$, which has a size independent of N. As a result, with the LORACs prior, minibatch-based learning becomes tractable even for very large datasets. The quality of the approximation to the TMC-VAE can be tuned by adjusting the size of M.

However, this technique presents its own inference challenges. Sampling the optimal variational factor $q^*(\tau)$ is no longer an option as it was in the TMC-VAE:

$$\begin{aligned} q^*(\tau; s_{1:M}) &\propto \exp\{\mathbb{E}_q \left[\log p(\tau, z_{1:N}, x_{1:N})\right]\} \\ &\propto \exp\{\log \mathrm{TMC}_N(\tau; a, b) \\ &\qquad + \textstyle\sum_{m=1}^{M} \log r(s_m \mid s_{1:m-1}, \tau) \\ &\qquad + \textstyle\sum_n \mathbb{E}_q \left[\log p(z_n \mid e_n, t_n, \tau)\right]\}. \end{aligned} \tag{13}$$

This term has a sum over N expectations; therefore computing this likelihood for the purpose of MCMC would involve passing the entire dataset through a neural network. Furthermore, the normalizer for this likelihood is intractable, but necessary for computing



(a) TMC-VAE graphical model (b) LORACs-VAE graphical model (c) LORACs-VAE variational factors

Figure 2: Graphical models and variational approximations for TMC models described in the paper

gradients w.r.t $s_{1:M}$. We therefore avoid using the optimal $q^*(\tau; s_{1:M})$ and set $q(\tau; s_{1:M})$ to the prior. This has the additional computational advantage of cancelling out the $\mathbb{E}_q[\log p(\tau)]$ term in the ELBO, which also has an intractable normalizing constant. If the inducing points are chosen so that they contain most of the information about the hierarchical organization of the dataset, then the approximation $p(\tau \mid z) \approx r(\tau \mid s_{1:M}) = p(\tau)$ will be reasonable.

We also fit the variational factors $q(e_n)$, $q_\xi(t_n \mid e_n, z_n; s_{1:M})$, and $q_\phi(z_n \mid x_n)$. The factor for attachment times, $q_\xi(t_n \mid e_n, z_n; s_{1:M})$, is a recognition network that outputs a posterior over attachment times for a particular branch. Since the $q(\tau; s_{1:M})$ and $p(\tau; s_{1:M})$ terms cancel out, we obtain the following ELBO (some notation suppressed for simplicity):

$$\mathcal{L}[q] \triangleq \mathbb{E}_q \left[\log \frac{\prod_n p(e_n, t_n \mid \tau) p(z_n \mid e_n, t_n, \tau) p(x_n \mid z_n)}{\prod_n q(e_n) q(t_n \mid e_n, z_n) q(z_n \mid x_n)} \right]. \tag{14}$$

This ELBO can be optimized by first computing

$$q^*(e_n) = \exp\left\{\mathbb{E}_q\left[\log p(e_n \mid t_n, z_n, \tau; s_{1:M})\right]\right\} \tag{15}$$

and computing gradients with respect to θ, $s_{1:M}$, ϕ, and ξ using a Monte-Carlo estimate of the ELBO using samples from $q(\tau; s_{1:M})$, $q^*(e_n)$, $q_\phi(z_n \mid x_n)$, and $q_\xi(t_n \mid e_n, z_n; s_{1:M})$. The factor $q(\tau; s_{1:M})$ can be sampled using vanilla SPR Metropolis-Hastings. The detailed inference procedure can be found in Appendix C.

5 Related work

As mentioned above, LORACs connects various ideas in the literature, including Bayesian nonparametrics (Boyles and Welling, 2012), inducing-point approximations (e.g.; Snelson and Ghahramani, 2006; Tomczak and Welling, 2018), and amortized inference (Kingma and Welling, 2014; Rezende et al., 2014). Also relevant is a recent thread of efforts to endow VAEs with the interpretability of graphical models (e.g.; Johnson et al., 2016; Lin et al., 2018). In this vein, Goyal et al. (2017) propose using a different Bayesian nonparametric tree prior, the nested Chinese restaurant process (CRP) (Griffiths et al., 2004), in a VAE. We chose to base LORACs on the TMC instead, as the posterior predictive distribution of an nCRP is a finite mixture, whereas the TMC's posterior predictive distribution has more complex continuous structure. Another distinction is that Goyal et al. (2017) only consider learning from pretrained image features, whereas our approach is completely unsupervised.

6 Results

We analyze properties of the LORACs prior, focusing on qualitative aspects, like exploratory data analysis and interpretability, and quantitative aspects, like few-shot classification and information retrieval.

Experimental setup We evaluated the LORACs prior on three separate datasets: dynamically binarized MNIST (LeCun and Cortes, 2010), Omniglot (Lake et al., 2015), and CelebA (Liu et al., 2015). For all three experiments, we utilized convolutional/deconvolutional encoders/decoders and a 40-dimensional latent space (detailed architectures can be found in Appendix D). We used 200, 1000, and 500 inducing points for MNIST, Omniglot, and CelebA respectively with TMC parameters $a = b = 2$. $q_\xi(t_n \mid e_n, z_n; s_{1:M})$ was a two-layer 500-wide neural network with ReLU activations that output parameters of a logistic-normal distribution over stick size and all parameters were optimized with Adam (Kingma and Ba, 2015). Other implementation details can be found in Appendix D.

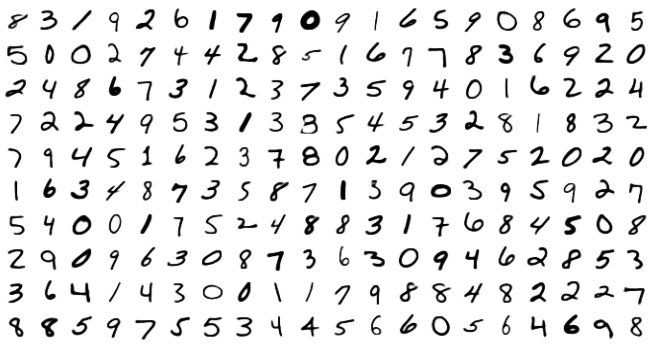
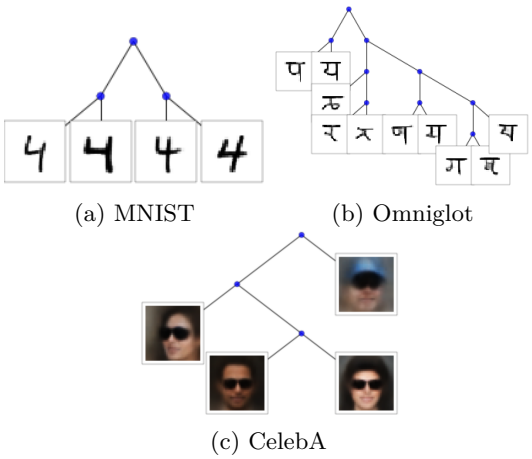


Figure 3: Learned inducing points for a LORACs(200) prior on MNIST.



(a) MNIST (b) Omniglot (c) CelebA

Figure 4: An example learned subtree from a sample of $q(\tau; s_{1:M})$ for each dataset. Leaves are visualized by passing inducing points throught the decoder.

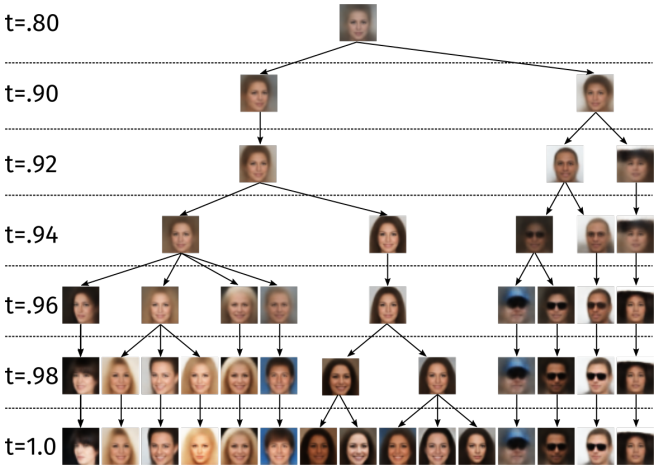


Figure 5: The evolution of a CelebA over a subset of inducing points. We create this visualization by taking slices of the tree at particular times and looking at the latent distribution at each of the sliced branches.

6.1 Qualitative results

A hierarchical clustering in the latent space offers a unique opportunity for interpretability and exploratory data analysis, especially when the data are images. Here are some methods for users to obtain useful data summaries and explore a dataset.

Visualizing inducing points We first inspect the learned inducing points $s_{1:M}$ by passing them through the decoder. Visualized in Figure 3 are the 200 learned inducing points for MNIST. The inducing points are all unique and are cleaner than pseudo-input reconstructions from VampPrior (shown in Figure A.13). Inducing points can help summarize a dataset, as visualizations of the latent space indicate they spread out and cover the data (see Figure A.11). Inducing points are also visually unique and sensible in Omniglot and CelebA (see Figure A.14 and A.15).

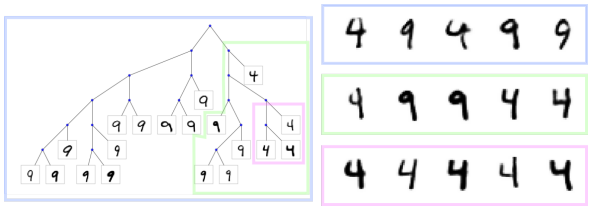
Hierarchical clustering We can sample $q(\tau; s_{1:M})$ to obtain phylogenies over the inducing points, and can visualize these clusterings using the decoded inducing points; subtrees from a sample in each dataset are visualized in Figure 4. In MNIST, we find large subtrees correspond to the discrete classes in the dataset. In Omniglot, subtrees sometimes correspond to language groups and letter shapes. In CelebA, we find subtrees sometimes correspond to pose or hair color and style.

We can further use the time at each internal node to summarize the data at many levels of granularity. Consider "slicing" the hierarchy at a particular time t by taking every branch $(\pi(v), v) \in E$ with $t_{\pi(v)} \leq t < t_v$ and computing the corresponding expected Gaussian random walk value at time t. At times closer to zero, we slice fewer branches and are closer to the root of the hierarchy, so the value at the slice looks more like the mean of the data. In Figure 5, we visualize this process over a subset of the inducing points of CelebA. Visualizing the dataset in this way reveals cluster structure at different granularities and offers an evolutionary interpretation to the data, as leaves that coalesce more "recently" are likely to be closer in the latent space.

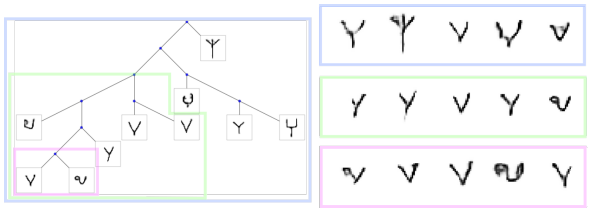
Although the hierarchical clustering is only over inducing points, we can still visualize where real data belong on the hierarchy by computing $q^*(e_n)$ and attaching the data to the tree. By doing this for many points of data, and removing the inducing points from the tree, we obtain an induced hierarchical clustering.

Generating samples Having fit a generative model to our data, we can visualize samples from the model. Although we do not expect the samples to have fidelity and sharpness comparable to those from GANs

or state-of-the-art decoding networks (Radford et al., 2015; Salimans et al., 2017), sampling with the LORACs prior can help us understand the latent space. To draw a sample from a TMC's posterior predictive density, we first sample a branch and time, assigning the sample a place in the tree. This provides each generated sample a *context*, i.e., the branch and subtree it was generated from. However, learning a LORACs prior allows us to conditionally sample in a novel way. By restricting samples to a subtree, we can generate samples from the support of the posterior predictive density limited to that subtree. This enables conditional sampling at many levels of the hierarchy. We visualize examples of this in Figure 6 and Figure 7.



(a) MNIST



(b) Omniglot

Figure 6: Conditional samples from subtrees.

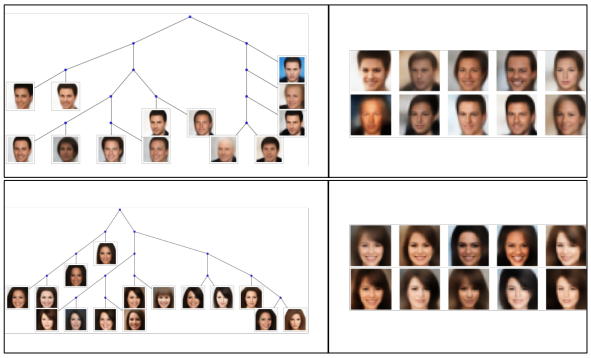
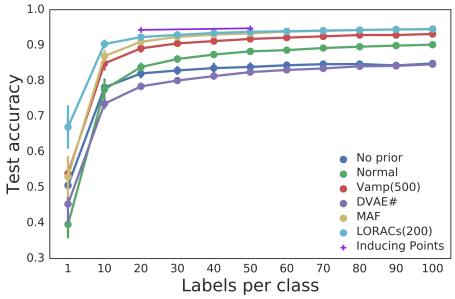


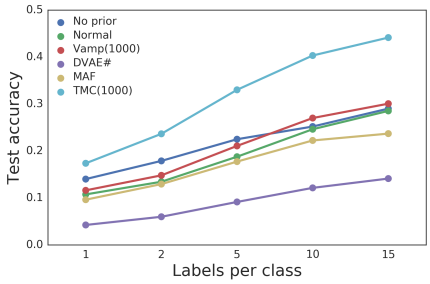
Figure 7: Samples from subtrees of CelebA.

6.2 Quantitative results

We ran experiments designed to evaluate the usefulness of the LORACs's learned latent space for downstream tasks. We compare the LORACs prior against a set of baseline priors on three different tasks: few-shot classification, information retrieval, and generative modeling. Our datasets are dynamically binarized MNIST and Omniglot (split by instance) and our baselines are representations learned with the same network architectures and latent dimensionality[3] but substituting the following prior distributions over z:



(a) MNIST



(b) Omniglot

Figure 8: Few-shot classification results

- No prior (classical autoencoder) - does not explicitly encourage any latent structure, and only encourages good representations for reconstruction
- Standard normal prior - not the correct inductive bias for classification tasks, which exhibit latent cluster structure
- VampPrior (Tomczak and Welling, 2018) - a mixture of Gaussians prior (a good inductive bias for classification) parameterized using a pseudo-input strategy related to ours
- DVAE♯ (Vahdat et al., 2018) - uses binary latent variables trained with continuous relaxations, included because it is a recently proposed model that also captures discrete structure in data
- Masked autoregressive flow (MAF; Papamakarios et al., 2017) - a state-of-the-art density estimator, expected to have the best held-out log-likelihood scores, but is biased towards smooth densities

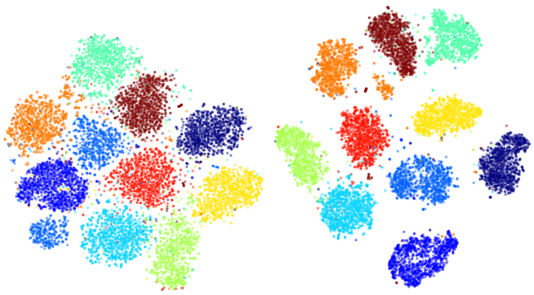
Details of architectures can be found in the appendix.

Few-shot classification In this task, we train a classifier with varying numbers of labels and measure

[3]Following the defaults in the author's reference implementation, we evaluated DVAE# on statically binarized MNIST with smaller neural networks, but with a higher-dimensional latent space.

test accuracy. We pick equal numbers of labels per class to avoid imbalance and we use a logistic regression classifier trained to convergence to avoid adding unnecessary degrees of freedom to the experiment. We replicated the experiment across 20 randomly chosen label sets for MNIST and 5 for Omniglot. The test accuracy on these datasets is visualized in Figure 8. For MNIST, we also manually labeled inducing points and found that training a classifier on 200 and 500 inducing points achieved significantly better test accuracy than randomly chosen labeled points, hinting that the LORACs prior has utility in an active learning setting.

The representations learned with the LORACs consistently achieve better accuracy, though in MNIST, LORACs prior and MAF reach very similar test accuracy at 100 labels per class. The advantage of the LORACs prior is especially clear in Omniglot (Table B.3 and Table B.4 contain the exact numbers). We believe our advantage in this task comes from ability of the LORACs prior to model discrete structure. TSNE visualizations in Figure 9 and Figure A.10 indicate clusters are more concentrated and separated with the LORACs prior than with other priors, though TSNE visualizations should be taken with a grain of salt.



(a) Normal prior (b) LORACs(200) prior

Figure 9: TSNE visualizations of the latent space of the MNIST test set with different priors, color-coded according to class. LORACs prior appears to learn a space with more separated, concentrated clusters.

Information retrieval We evaluated the meaningfulness of Euclidean distances in the learned latent space by measuring precision-recall when querying the test set. We take each element of the test set and sort all other members according to their L_2 distance in the latent space. From this ranking, we produce precision-recall curves for each of the query and plot the average precision-recall over the entire test set in Figure B.16. We also report the area-under-the-curve (AUC) measure for each of these curves in Table 1.

AUC numbers for Omniglot are low across the board because of the large number of classes and low number of instances per class. However, in both datasets the LORACs prior consistently achieves the highest AUC, especially with MNIST. The LORACs prior encourages tree-distance to correspond to squared Euclidean distance, as branch lengths in the tree are variances in a Gaussian likelihoods. We thus suspect distances in a LORACs prior latent space to be more informative and better for information retrieval.

Held-out log-likelihood We estimate held-out log-likelihoods for the four VAEs we trained with comparable architectures and different priors. (We exclude DVAE♯ since its architecture is substantially different, and the classical autoencoder since it lacks generative semantics.) We use 1000 importance-weighted samples (Burda et al., 2015) to estimate held-out log-likelihood, and report the results in Table 2. We find that, although LORACs outperforms the other priors on downstream tasks, it only achieves middling likelihood numbers. This result is consistent with the findings of Chang et al. (2009) that held-out log-likelihood is not necessarily correlated with interpretability or usefulness for downstream tasks.

Prior	MNIST	Omniglot
No prior	0.429	0.078
Normal	0.317	0.057
VAMP	0.502	0.063
DVAE#	0.490	0.024
MAF	0.398	0.070
LORACs	**0.626**	**0.087**

Table 1: Averaged precision-recall AUC on MNIST/Omniglot test datasets

Prior	MNIST	Omniglot
Normal	-83.789	-89.722
MAF	**-80.121**	**-86.298**
Vamp	-83.0135	-87.604
LORACs	-83.401	-87.105

Table 2: MNIST/Omniglot test log-likelihoods

7 Discussion

Learning discrete, hierarchical structure in a latent space opens a new opportunity: interactive deep unsupervised learning. User-provided constraints have been used in both flat and hierarchical clustering (Wagstaff and Cardie, 2000; Awasthi et al., 2014), so an interesting follow up to this work would be incorporating constraints into the LORACs prior, as in Vikram and Dasgupta (2016), which could potentially enable user-guided representation learning.

Acknowledgements

Sharad Vikram is supported by NSF grant CNS-1446912.

References

Awasthi, P., Balcan, M., and Voevodski, K. (2014). Local algorithms for interactive clustering. In *International Conference on Machine Learning*, pages 550–558.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Boyles, L. and Welling, M. (2012). The time-marginalized coalescent prior for hierarchical clustering. In *Advances in Neural Information Processing Systems*, pages 2969–2977.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Goyal, P., Hu, Z., Liang, X., Wang, C., Xing, E. P., and Mellon, C. (2017). Nonparametric variational auto-encoders for hierarchical representation learning. In *ICCV*, pages 5104–5112.

Griffiths, T. L., Jordan, M. I., Tenenbaum, J. B., and Blei, D. M. (2004). Hierarchical topic models and the nested chinese restaurant process. In *Advances in neural information processing systems*, pages 17–24.

Hoffman, M. D. and Johnson, M. J. (2016). Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*.

Johnson, M. J., Duvenaud, D., Wiltschko, A. B., Datta, S. R., and Adams, R. P. (2016). Composing graphical models with neural networks for structured representations and fast inference. In *Neural Information Processing Systems*.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations*.

Kingman, J. F. C. (1982). The coalescent. *Stochastic processes and their applications*, 13(3):235–248.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

Lin, W., Hubacher, N., and Khan, M. E. (2018). Variational message passing with structured inference networks. *arXiv preprint arXiv:1803.05589*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Neal, R. M. (2003). Density modeling and clustering using dirichlet diffusion trees. *Bayesian statistics*, 7:619–629.

Papamakarios, G., Murray, I., and Pavlakou, T. (2017). Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286.

Salimans, T., Karpathy, A., Chen, X., Kingma, D. P., and Bulatov, Y. (2017). Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations (ICLR)*.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.

Tomczak, J. and Welling, M. (2018). Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223.

Vahdat, A., Andriyash, E., and Macready, W. G. (2018). Dvae#: Discrete variational autoencoders with relaxed Boltzmann priors. In *Neural Information Processing Systems (NIPS)*.

Vikram, S. and Dasgupta, S. (2016). Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090.

Wagstaff, K. and Cardie, C. (2000). Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584.