**Published at ProbML 2026**

# RAMP: Recognition parametrisation by Amortised Message Passing

**Lior Fox**[1] **Kai Biegun**[1,2] **James Heald**[1] **Samo Hromadka**[1] **Arielle Rosinski**[1] **Maneesh Sahani**[1]

[1]Gatsby Computational Neuroscience Unit
[2]Centre for Artificial Intelligence
University College London

## Abstract

A central aim of unsupervised learning is to uncover latent factors that explain dependencies among observations. Probabilistic models typically achieve this by introducing multiple latent variables linked through a graph of conditional relationships, with distributional parameters and their dependence learnt from data. Learning relies either on distributional choices that allow tractable belief propagation, or on approximations that scale poorly with model size and complexity. We build on the recently developed recognition-parametrised modelling paradigm to propose an alternative approach: RAMP, a method that implicitly defines latent structure by learning a flexible, nonlinear, amortised message-passing framework. We show that RAMP enables efficient likelihood-based recovery of latent-variable distributions within expressive nonlinear models acting on complex high-dimensional data.

## 1. Introduction

We understand the world by building a mental model of it. Good models help to describe complex phenomena simply, by identifying and recovering relevant latent factors that explain patterns of regularity and dependence amongst observations. Mechanising such model building has been an overarching goal for artificial intelligence and machine learning for decades. Probabilistic methods are particularly well-suited to this task. They express non-deterministic links between variables, represent uncertainty in a mathematically coherent way, and offer a principled approach for inference in the form of the posterior distribution over hidden variables, given the observations.

Yet, in the tradeoff between expressivity and simplicity, probabilistic methods often lean to the latter. The challenge is that computations in probabilistic models depend on integration over variables, and these integrals are only tractable for the simplest of distributional choices. Numerical or approximate methods have been developed in many other settings, but these become increasingly forbidding as the complexity of probabilistic dependence increases. This need to approximate has been a major driving force behind attempts to integrate neural network approaches, which excel at learning to approximate complicated functions, into probabilistic methods. Despite some success, a general unified approach is still missing for this integration. One issue is that the nature of approximations offered by the two approaches is not always compatible, and the internal representations generated by generic deep learning methods do not easily lend themselves to probabilistic interpretation.

This paper presents a novel approach towards merging adaptive neural-network components into a probabilistic framework, based on three core ideas. The first is to train a set of networks to collectively perform inference, leveraging the ability of pattern-recognition methods to amortise

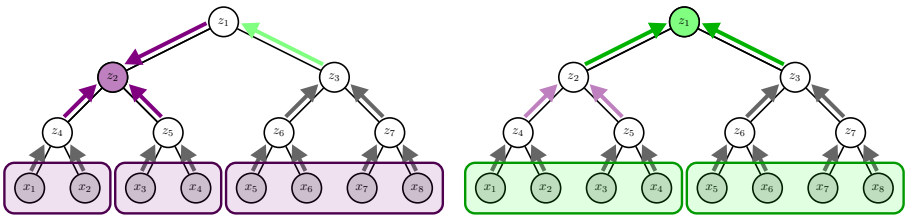


Figure 1: RAMP works by learning a system of amortised message-passing transformations (directed arrows) within a tree structured graphical model. At each latent node, the incoming messages (bold coloured arrows) serve as a set of *recognition factors*, that are trained to model the specific conditional independence relationships among observations induced by the corresponding latent (denoted by the coloured bounding boxes). The use of a single set of message-passing functions, shared between all the nodewise RPMs, leads to every message reporting a sufficient statistic for the set of its incoming observations, relative to the rest of the observations.

complicated transformations. The second is to constrain the way in which the outputs of these networks are interpreted, transformed, and combined together. These constraints, together with the learning objective itself, are derived directly from probabilistic considerations encoded in a graphical model. Finally, the third core idea is that of recognition-parametrisation, allowing the *inference* ("recognition") procedure to directly define the model itself.

## 2. Mathematical Background

### 2.1. Recognition-Parametrised Models

The Recognition-Parametrised Model (RPM; Walker et al., 2023) describes $P$ observed variables $\mathcal{X} = \{x_1 \dots x_P\}$ in terms of one or more latents $\mathcal{Z} = \{z_1 \dots z_K\}$, such that the dependence amongst the observations is captured by $\mathcal{Z}$. Such a model would conventionally be defined by parametrising a prior $p(\mathcal{Z})$ and generative conditionals $p(x_p|\mathcal{Z})$. The RPM is semi-parametric, combining a nonparametric summary of the $x_p$-marginals, $p_{\text{emp}}(x_p)$, with normalised recognition factors $f(\mathcal{Z}|x_p)$ that map observed values to posterior belief distributions over the latent variables. The joint is then

$$p(\mathcal{Z}, \mathcal{X}) = p(\mathcal{Z}) \prod_p \frac{f_p(\mathcal{Z}|x_p)}{F_p(\mathcal{Z})} p_{\text{emp}}(x_p) \qquad \text{with} \qquad F_p(\mathcal{Z}) = \int dx_p \, f_p(\mathcal{Z}|x_p) p_{\text{emp}}(x_p) \,. \tag{1}$$

Like Walker et al., we take $p_{\text{emp}}(x_p) = \frac{1}{N} \sum_{n=1}^{N} \delta(x_p - x_p^{(n)})$, so that $F_p(\mathcal{Z}) = \frac{1}{N} \sum_{n=1}^{N} f_p\Big(\mathcal{Z}|x_p^{(n)}\Big)$.

The RPM is a normalised model with a proper likelihood function. Recognition parameters can be learnt by optimising a free energy bound (Neal and Hinton, 1998) as discussed in Appendix A.4.

### 2.2. Tree-Structured Graphical Models

We consider undirected, tree-structured, graphical models. To simplify presentation, we focus on models with pairwise (and singleton) factors, but the key ideas we develop generalise to any tree-structured factor graph (see Appendix A.6).

Let $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be a tree with nodes $\mathcal{V}$ and (undirected) edges $\mathcal{E}$. We identify the nodes with variables in a probabilistic model, partitioned into *latent variables* $\mathcal{Z}$ and *observed variables* $\mathcal{X}$; $\mathcal{V} = \mathcal{Z} \cup \mathcal{X}$, and refer to the nodes interchangeably either by the variable index (e.g. $k \in \{1 \dots K\}$) or by the associated variable label (e.g. $z_k$). Without loss of generality[1], we take the observations to be leaf

1. If an observation were at an internal node it would partition $\mathcal{T}$ into two or more subtrees within which inference and learning would proceed independently.

nodes of the tree; implying that $\mathcal{E}$ is partitioned into sets linking two latents ($\mathcal{E}_{\text{zz}}$), or a latent to an observation ($\mathcal{E}_{\text{zx}}$); $\mathcal{E} = \mathcal{E}_{\text{zz}} \cup \mathcal{E}_{\text{zx}}$. This graphical structure defines a family of joint probability distributions over observed and latent variables, satisfying the following factorisation:

$$p(z_{1:K}, x_{1:P}) = \frac{1}{Z} \prod_k \phi_k(z_k) \prod_p \psi_p(x_p) \times \prod_{(k,k')\in\mathcal{E}_{\text{zz}}} \Phi_{kk'}(z_k, z_{k'}) \prod_{(k,p)\in\mathcal{E}_{\text{zx}}} \Psi_{kp}(z_k, x_p) \tag{2}$$

### 2.2.1. Trees and Conditional Independence

In a tree-structured graph, the removal of a latent node $z_k$ with degree $\deg(k)$ partitions the graph $\mathcal{T}$ into a collection of $\deg(k)$ disjoint sub-trees, one for each edge at $k$ (see Fig. 1). We label the sub-trees by the neighbour to which the corresponding edge connects: $\{\mathcal{T}_j^k\}_{j\in\partial_k}$, where $\partial_k$ is the neighbourhood of $z_k$ (note that $j$ may index an observed variable, in which case $\mathcal{T}_j^k$ contains the single node $x_j$). As $\mathcal{T}_j^k$ is defined by the removal of $z_k$ from the graph, $\mathcal{T}_j^k \neq \mathcal{T}_k^j$.

Let $\mathcal{X}_j^k$ be the set of observed variables in the tree $\mathcal{T}_j^k$. The graphical model defined by $\mathcal{T}$ implies that the sets $\mathcal{X}_j^k$ for $j \in \partial_k$ are conditionally independent given $z_k$. We refer to this property as the $z_k$-"nodewise" conditional independence partition of the observations. These conditional independence properties provide the principal signal for learning within a nonlinear tree-structured model: $z_k$ is defined by the need to uniquely model the dependence amongst the different subsets of observations $\{\mathcal{X}_j^k\}$. The induced partition depends on where the latent variable sits within the tree. For example, in the graph shown in Fig. 1, $x_2$ is conditionally independent of $x_3$ given $z_2$, but not $z_1$. From this point of view, a latent variable is only discoverable on the basis of the conditional independence it induces amongst observations—to paraphrase Firth (1957), "you shall know a latent by the observables it keeps". Thus, we focus on trees in which each latent variable induces a unique partition of the observations. This is captured by the following assumption:

**Assumption 1.** *For every pair $z_i$, $z_j$ $(i \neq j)$ in $\mathcal{T}$, there exist $x, x' \in \mathcal{X}$ that belong to the same sub-tree of $z_i$ but to different sub-trees of $z_j$.*

We have focused on conditional independence induced on the observed (leaf) nodes, but independence amongst latents is an equally important aspect of the model. For example, in a Markovian state-space model, these independencies define the Markov chain structure of the latent series $z_t$. Fortunately, as the following lemma shows, a model defined by all nodewise conditional independence relationships on observations necessarily also captures conditional independence amongst the latents.

**Lemma 2.** *Let $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ satisfy Assumption 1. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathcal{G}})$ be a connected graph on the same nodes, satisfying the nodewise conditional independence properties implied by $\mathcal{T}$. Then, $\mathcal{E}_{\mathcal{G}} = \mathcal{E}_{\mathcal{T}}$.*

*Proof.* We show that $\mathcal{E}_{\mathcal{G}} \subseteq \mathcal{E}_{\mathcal{T}}$; equality follows from the assumptions that $\mathcal{T}$ is a tree and $\mathcal{G}$ is connected. Let $(i,j) \notin \mathcal{E}_{\mathcal{T}}$. Because $\mathcal{T}$ is a tree with $\mathcal{X}$ at its leaves, there must be a path from $z_i \rightsquigarrow z_j$ passing through at least one additional node $z_k$ ($k \notin \{i,j\}$). By Assumption 1 there exist distinct $x_a, x_b$ such that $x_a \perp\!\!\!\perp x_b | z_i$ but $x_a \not\perp\!\!\!\perp x_b | z_k$. Similarly, there are $x_c, x_d$ separated by $z_j$, but not by $z_k$. As $\mathcal{G}$ satisfies the nodewise conditional independencies of $\mathcal{T}$, there must be a path $x_a \rightsquigarrow x_b \in \mathcal{E}_{\mathcal{G}}$ that passes through $z_i$ but not $z_k$. Similarly, there is a path $x_c \rightsquigarrow x_d \in \mathcal{E}_{\mathcal{G}}$ through $z_j$ but not $z_k$. Now, by construction, $x_a$ and $x_d$ belong to different sub-trees partitioned by $z_k$, and so $x_a \perp\!\!\!\perp x_d | z_k$ in $\mathcal{T}$, from which it follows that $x_a \perp\!\!\!\perp x_d | z_k$ also in $\mathcal{G}$. But if $\mathcal{G}$ contained the edge $(i,j)$, there would be a path $x_a \rightsquigarrow z_i \to z_j \rightsquigarrow x_d$ which does not pass through $z_k$, violating this separation. So $(i,j) \notin \mathcal{E}_{\mathcal{T}} \Rightarrow (i,j) \notin \mathcal{E}_{\mathcal{G}}$ and $\mathcal{E}_{\mathcal{G}} \subseteq \mathcal{E}_{\mathcal{T}}$ as required. □

### 2.2.2. Message passing

It is well known that exact inference is tractable for tree-structured graphs (Koller and Friedman, 2009), and can be implemented efficiently by belief propagation (BP; Pearl, 1988) *provided* that

integrals over local beliefs can be computed exactly. Defining potentials $\alpha_{j\to k}(z_k) = p(\mathcal{X}_j^k|z_k)$ at each latent, BP provides the recursive updates

$$\alpha_{j\to k}(z_k) = \int dz_j\, p(z_j|z_k) \prod_{l\in\partial_j\setminus k} \alpha_{l\to j}(z_j) \quad \text{if } j\in\mathcal{Z}; \qquad \alpha_{j\to k}(z_k) = p(x_j|z_k) \quad \text{if } j\in\mathcal{X}, \tag{3}$$

where $p(z_j|z_k)$ and $p(x_j|z_k)$ are derived from the factors of Eq. (2). To compute $p(z_k|\mathcal{X})$, we follow the "inward" path from each $x_p$ to $z_k$, computing $\alpha_{r\to s}(z_s)$ for each edge $(r,s)$ encountered. Then

$$p(z_k|\mathcal{X}) = \frac{1}{L_k} p(z_k,\mathcal{X}) = \frac{1}{L_k} p(z_k) \prod_{j\in\partial_k} \alpha_{j\to k}(z_k)\,, \tag{4}$$

where $L_k$ is easily computed by integration over the single variable $z_k$. Furthermore, $L_k = p(\mathcal{X})$ is the model likelihood, and so, provided messages are computed exactly, is the same at every $k$. As will be seen, RAMP optimises all of these likelihoods together. Fortunately, BP allows efficient computation of every nodewise likelihood: a single "inward-outward" pass of messages with respect to any one node in the tree (taking order $|\mathcal{V}|$ steps) is sufficient to compute them all.

In practice, exact messages can only be computed for a limited class of generative conditionals. More general (nonlinear) dependence between variables requires approximation (Wainwright and Jordan, 2008), often combined with Monte-Carlo or numerical integration (e.g., Ranganath et al., 2014).

## 3. RAMP

We are now ready to define RAMP. Let $\mathcal{X}$ be a set of observed variables, and $\{\mathcal{X}^{(n)}\}_{n=1}^N$ an observed dataset, with a joint captured by a tree-structured model $\mathcal{T} = (\mathcal{Z}\cup\mathcal{X}, \mathcal{E})$ satisfying Assumption 1.

### 3.1. Amortised Message-Passing Network

We define the RAMP model in terms of message-passing-based recognition, in the spirit of the RPM (Eq. (1)). The latent variables $z_k$ in Eq. (2) can be transformed by an arbitrary bijection, and the relevant factors composed with the inverse, without altering the joint distribution. Thus, without loss of generality, we fix marginal priors $f_0^k(z_k)$. Let $f_j^k(z_k|\mathcal{X}_j^k)$ be a recognition factor that defines a normalised message representing a belief over $z_k$ based on $\mathcal{X}_j^k$. The tree-based factorisation implies that this belief should depend on messages to $z_j$ in a manner analogous to Eq. (3),

$$f_j^k(z_k|\mathcal{X}_j^k) = \int dz_j\, p(z_k|z_j) f_0^j(z_j) \prod_{l\in\partial_j\setminus k} \frac{f_l^j(z_j|\mathcal{X}_l^j)}{f_0^j(z_j)}\,. \tag{5}$$

Eq. (5) requires parametrisation of $p(z_k|z_j)$ (or of the factors in Eq. (2) from which it is derived), and depends on evaluation of the integral over $z_j$. Thus, graph factors are often constrained to a simple tractable form. In RAMP, we instead amortise the belief-to-belief transformation directly, defining

$$f_j^k(z_k|\mathcal{X}_j^k) = \mathcal{G}_{j\to k}\left( f_0^j(z_j) \prod_{l\in\partial_j\setminus k} \frac{f_l^j(z_j|\mathcal{X}_l^j)}{f_0^j(z_j)} \right), \tag{6}$$

in terms of a generic functional $\mathcal{G}_{j\to k}$ for latent-to-latent messages. Note that the tree structure implies that $\mathcal{X}_j^k = \bigcup_{l\in\partial_j\setminus k} \mathcal{X}_l^j$, so that the distributions on the two sides of Eq. (6) are conditioned on the same observations – the functional $\mathcal{G}$ transforms a belief on $z_j$ (given $\mathcal{X}_j^k$) into a belief on $z_j$ (given $\mathcal{X}_j^k$). Finally, in the case of a singleton observation group, observation-to-latent messages are defined directly by $f_p^k(z_k|x_p)$ as in the RPM.

Thus, the functions $\{f_p^k(z_k|x_p)\}_{(k,p)\in\mathcal{E}_{\text{zx}}}$ and functionals $\{\mathcal{G}_{j\to k}\}_{(j,k)\in\mathcal{E}_{\text{zz}}}$ together define a single amortised message-passing network which computes posterior beliefs over each $z_k$ given the observations.

### 3.2. Nodewise RPMs

The key to RAMP is to learn the amortised message-passing factors by simultaneously training multiple nodewise RPMs, one for each $z_k$. These models take the form

$$\mathsf{P}_k(z_k, \mathcal{X}) = f_0^k(z_k) \prod_{j \in \partial_k} \frac{f_j^k(z_k|\mathcal{X}_j^k) p_{\text{emp}}(\mathcal{X}_j^k)}{F_j^k(z_k)}, \tag{7}$$

where $f_0^k(z_k)$ is the marginal prior on $z_k$, $f_j^k(z_k|\mathcal{X}_j^k)$ are recognition factors defined by amortised message passing (Eq. (6)), $p_{\text{emp}}(\mathcal{X}_j^k)$ is the empirical distribution of $\mathcal{X}_j^k$, and $F_j^k(z_k)$ are defined by

$$F_j^k(z_k) = \int d\mathcal{X}_j^k \, f_j^k(z_k|\mathcal{X}_j^k) p_{\text{emp}}(\mathcal{X}_j^k) = \frac{1}{N} \sum_{n=1}^{N} f_j^k\left(z_k|\mathcal{X}_j^{k(n)}\right), \tag{8}$$

as in the RPM. The second equality in Eq. (8) follows from the atomic choice for $p_{\text{emp}}$.

The nodewise RPM at (each) $z_k$ is a model of the observed data that respects the conditional independences induced by $z_k$. Because $z_k$ is latent, fitting the parameters of this model requires optimising a variational free energy (or ELBO). As the nodewise RPMs have the canonical form introduced by Walker et al. (2023), we can directly adopt the form of the free energy derived there:

$$\mathcal{F}_k^{(n)}\left(q^{(n)}(z_k), \theta\right) = \left\langle \log f_0^k(z_k) + \sum_{j \in \partial_k} \log \frac{f_j^k(z_k|\mathcal{X}_j^k)}{F_j^k(z_k)} \right\rangle_{q^{(n)}} + \mathbf{H}\left[q^{(n)}\right], \tag{9}$$

where $\theta$ are the RAMP parameters, $q^{(n)}$ is a variational distribution, and $\mathbf{H}[\cdot]$ is the entropy.

### 3.3. Joint optimisation

Even though the recognition factors of all the nodewise RPMs are defined by a single amortised message-passing network, optimisation of a single $\mathcal{F}_k$ is not sufficient for accurate learning. Most obviously, only factors $\mathcal{G}_{i \to j}$ that lie along inward paths to $k$ appear in the definitions of $f_{()}^k$. More subtly, the $z_k$-nodewise RPM is defined in terms of $p_{\text{emp}}(\mathcal{X}_j^k)$, and so does not by itself induce the additional tree-based condition that the different observations within $\mathcal{X}_j^k$ be conditionally independent given the latents in $\mathcal{T}_j^k$. Thus, RAMP instead optimises the summed objective

$$\mathcal{F} = \frac{1}{K} \sum_k \mathcal{F}_k = \frac{1}{KN} \sum_{k,n} \mathcal{F}_k^{(n)}. \tag{10}$$

This approach is justified by the following straightforward theorem.

**Theorem 3.** *Let the distribution $p(\mathcal{X})$ be compatible with a tree-structured latent model $\mathcal{T}$ satisfying Assumption 1, and let $\mathcal{Z}^* = \{z_k^*\}$ be minimal latent variables in $\mathcal{T}$ such that the beliefs $p(z_k^*|\mathcal{X}_j^k)$ are each minimal sufficient statistics of $\mathcal{X}_j^k$ for $\bigcup_{l \in \partial_k \setminus j} \mathcal{X}_l^k$. Define an amortised message passing network on $\mathcal{T}$ according to Eq. (6). Then, if $\{f_0^k(z_k)\}$, $\{f_p^k(z_k|x_p)\}$ and $\{\mathcal{G}_{j \to k}\}$ all optimise the objective of Eq. (10) in the limit $p_{emp}(\mathcal{X}_j^k) = p(\mathcal{X}_j^k)$, we have:*

1. *each $\mathcal{F}_k$ attains its optimal value,*
2. *the true beliefs $p(z_k^*|\mathcal{X}_j^k)$ are each a function of $f_j^k(z_k|\mathcal{X}_j^k)$.*

*Proof.* As $p(\mathcal{X})$ is compatible with $\mathcal{T}$, the set of possible amortised message-passing networks includes one that computes the true beliefs $p(z_k^*|\mathcal{X}_j^k)$. That is, there exist functions $\{f_0^k(z_k)\}$, $\{f_p^k(z_k|x_p)\}$ and

$\{\mathcal{G}_{j\to k}\}$ defining accurate beliefs in a model that renders the sets $\{\mathcal{X}_j^k\}_{j\in\partial_k}$ conditionally independent given $z_k$ for all $k$. Thus there is at least one solution that optimises all the $\mathcal{F}_k$ and (1) follows. As each nodewise RPM is then an accurate model of the joint over $\{\mathcal{X}_j^k\}_{j\in\partial_k}$, the amortised belief $f_j^k(z_k|\mathcal{X}_j^k)$ is a sufficient statistic of $\mathcal{X}_j^k$ for $\bigcup_{l\in\partial_k\setminus j}\mathcal{X}_l^k$. Claim (2) then follows as $p(z_k^*|\mathcal{X}_j^k)$ is a minimal sufficient statistic by assumption, and so a function of any other sufficient statistic. □

### 3.4. Exponential-family parametrisation

To implement the amortised message passing of Eq. (6) we must represent, multiply, and flexibly transform beliefs $f_j^k(z_k|\mathcal{X}_j^k)$; and to optimise Eq. (9), we must compute integrals over $z_k$. A natural parametrisation that simplifies both steps is given by a tractable exponential family. Specifically, we choose $f_0^k(z_k)$ and $f_j^k(z_k|\mathcal{X}_j^k)$ at each $z_k$ to be instances of the same exponential family. The choice of family may differ for different $z_k$. Furthermore, the choice of an exponential family for $f_j^k(z_k|\mathcal{X}_j^k)$ does not restrict the form of joint factor implied by the amortised $\mathcal{G}_{j\to k}$. Instead, the learnt mapping may correspond to an exponential-family projection of an arbitrary joint potential. In effect, this choice approximates the marginal posterior over $z_k$ with a tractable parametric form, while allowing the parameters of that posterior to depend on complicated learnt functional mappings from the observations. Although Theorem 3 no longer applies directly, parametric optimisation minimises the sum of nodewise Kullback-Leibler divergences from the optimal model of that result.

Exponential family distributions can be represented by natural parameters. In this representation, the product of distributional factors is easy to compute: the result is another distribution in the same family, with natural parameters equal to the sum of the factor natural parameters.

A remaining challenge is to handle the $F_j^k(z_k)$, which are mixtures of $f_j^k(z_k|\mathcal{X}_j^k)$. We adopt the "interior variational" bound introduced by Walker et al. (2023). This replaces $F_j^k$ with an exponential family approximation, adding terms to the free energy that ensure it remains a well-behaved lower bound. Furthermore, based on the observation that $F_j^k \to f_0^k$ as $N\to\infty$ in a well-specified model, we define this bound using $f_0^k$ as the approximating functions. Further details appear in Appendix A.

The effect of these assumptions is to make message propagation in RAMP particularly straightforward. If $\eta_{i\to j}$ are the natural parameters of $f_i^j(z_j|\mathcal{X}_i^j)$ then

$$\eta_{j\to k} = g_{j\to k}\Big(\sum_{i\in\partial_j\setminus k}\eta_{i\to j}\Big), \tag{11}$$

where $g_{j\to k}$ is a learned function (e.g., a neural network), mapping natural parameters to natural parameters. Different latent variables may belong to different exponential families: the only requirement is that all the $g_{i\to j}$ functions for fixed $j$ return natural parameters of the family of $z_j$.

## 4. Related Work

There is a long history of methods for approximate inference (Sudderth et al., 2003; Winn and Bishop, 2005; Ihler and McAllester, 2009; Lienart et al., 2015) and learning in intractable models. Rather than an extensive review, we discuss key approaches that are most relevant to RAMP.

Black Box Variational Inference (BBVI; Ranganath et al., 2014) optimises the parameters of a variational distribution by following a stochastic gradient of the free-energy function, employing the REINFORCE trick to compute Monte-Carlo gradients (Williams, 1992). This method requires the joint model to be specified explicitly, and the variational family is often fully factorised ("naive" mean-field). In contrast, inference in RAMP more directly resembles belief propagation—the recognition factors learn to amortise a marginalisation operator of an implied factor. This makes our method better suited to discover structured or hierarchical latent representations (see Section 5.1 and Appendix B.3).

Structured variational autoencoders (Johnson et al., 2016; Zhao and Linderman, 2023) combine nonlinear recognition factors with either tractable exact or mean-field variational message passing on a latent graph. Similarly, standard applications of the RPM combine nonlinear recognition factors with tractable or variational inference between latents (Walker et al., 2023; Hromadka et al., 2025). Again, RAMP combines these nonlinear recognition factors at the leaves with learnt amortised message passing to capture more complex relationships between latents.

In some ways, the exponential-family approximation of messages in RAMP most closely resembles the framework of expectation propagation (EP, Minka, 2001). However, as with variational methods, EP depends on a parametrisation of generative model factors that are then approximated. For flexibly defined potentials, these approximations may themselves require integrals that are analytically intractable, and numerical approximation introduces error and potential bias (Boyen and Koller, 1999; Yu et al., 2006). RAMP avoids these challenges by *defining* the model in terms of the projected exponential family belief propagation, exploiting the RPM framework to ensure that the resulting models still have well-defined likelihoods.

Leveraging the "nodewise" conditional independences for learning latent representations has been explored in some contrastive methods, particularly in the context of temporal sequences (van den Oord et al., 2018; Eysenbach et al., 2022). Unlike contrastive methods, RAMP builds an explicit probabilistic model linking observations and latent variables, that can be estimated by (approximate) maximum likelihood. The probabilistic framework also makes our method directly applicable to graphical structures that are not necessarily temporal sequences (see Sections 5.1 and 5.3 and Appendix B.3), where contrastive methods have been less explored, and are likely to require ad-hoc adjustments.

The idea of composing neural networks in a graph-defined structure has been explored extensively. In the context of sequences, intuition about the forward-backward structure of inference has inspired Bidirectional RNN models (Schuster and Paliwal, 1997; Baldi et al., 1999; Graves and Schmidhuber, 2005; Arisoy et al., 2015). Applied to time series, RAMP can also be seen as a bidirectional RNN, where the hidden state has direct probabilistic and inferential semantics (see Section 5.2). Beyond sequences, Graph Neural Networks (for review, see Sanchez-Lengeling et al., 2021) have been proposed for representation learning from richly structured data. However, these typically rely on a synchronised message-passing computation, limiting their applicability for inference problems (Solodova et al., 2024). Rather than iterating a global "graph layer", in RAMP each edge has its own neural network, and the message-passing flow is asynchronous, as in classic belief-propagation.

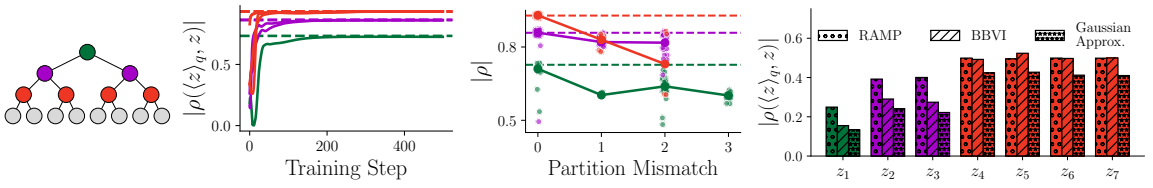
## 5. Results

### 5.1. Hierarchical Models

We begin with a simple motivating example of hierarchical latents models, with the graphical structure shown in Fig. 2a. We use this family of problems to exemplify the key components of our method, as well as its robustness to structural and distributional misspecification.

**Linear Gaussian Trees** Jointly Gaussian data were generated from a tree by sampling a root latent from $\mathcal{N}(0, 1)$, and children from $z|\text{pa}(z) \sim \mathcal{N}(a \cdot \text{pa}(z), \sigma^2)$. Leaf nodes were observed. Joint Gaussianity means that exact inference is tractable (given the true parameters of the model). Nonetheless, learning is still non-trivial due to the existence of latent variables; moreover, the natural parameter transformations of message passing are nonlinear, providing a nontrivial test of RAMP. Trained on this data, RAMP recovered the exact solution with high fidelity. (Fig. 2b).

**Structural Misspecification** We next consider the scenario in which the graphical structure assumed in the model does not perfectly match that of the true data generating process. The main learning signals in RAMP are the nodewise free energies, which encourage learned latents to capture information that is shared between the variables in different associated subtrees (thereby




(a) (b) (c) (d)

Figure 2: (a) Hierarchical structure with 3 levels of latent variables (color coded) arranged on a binary tree and observations on the leaves. (b) Under a linear-Gaussian data generating process, Pearson correlations between the posterior means inferred by RAMP and the ground truth latents (solid lines) reach the optimal bound achieved by exact inference with the true generative model (dashed lines). (c) Data as in (b), but RAMP is trained using misspecified trees. The quality of representations learned at individual latents degrade with the amount by which they are perturbed (see main text). Crucially, latents that are structurally intact (partition mismatch 0) maintain their unperturbed performance (dashed lines), even though their sub-trees are misspecified. Solid lines trace the median level of correlations within each group of latents. (d) Non-linear data generating process. RAMP learns more accurate representations compared to a black box method and a (supervised) Gaussian approximation, measured by the Spearman correlation of inferred posterior means and true latents.

rendering those subtrees conditionally independent; Theorem 3). That optimisation pressure should remain accurate at any node that implies a partition on observations that is correct, and useful when violations relative to the true structure are few, even if each subtree (internal) structure is misspecified.

To test this claim, we fit RAMP models based on randomly perturbed trees to data generated from a balanced base tree (see Appendix B.2). As the labeling of latents is arbitrary, for analysis each latent in a perturbed tree was matched to the best candidate in the original tree, by minimising the partition mismatch metric. This metric counts how many observations are routed to the wrong sub-tree. In general, the ability of RAMP to recover the latent variable degraded as the partition mismatch increased. However, latents which were structurally intact (partition mismatch 0) *within* a perturbed tree maintained an almost identical correlation with the corresponding ground-truth latent as that obtained in the well-specified tree (Fig. 2c).

**Distributional Misspecification** Another form of model misspecification is distributional. In RAMP the (marginal) posteriors are constrained to be of an exponential family (here we use Gaussians), which in general would be an approximation. We tested the quality of this approximation by making the data generating process non-linear, thus making the true posteriors non-Gaussian. Here, RAMP outperformed Black Box methods applied to parametrised generative models (Fig. 2d).

Additional details and results for this section appear in Appendices B.1 to B.3, C.1 and C.2.

### 5.2. Nonlinear Dynamical Systems

An important application of latent variable models is to time-series data, under the general assumption of a Markovian state-space model; the sequence of temporally correlated observations $x_{1:T}$ is assumed to be generated by a hidden Markov process, $z_{1:T}$, and an instantaneous emission process. The graph thus comprises a chain of latent variables, each connected to a single (unique) observation. Conditioning on the latent variable at $z_t$ should make $\{x_{1:t-1}\}$, $\{x_t\}$, and $\{x_{t+1:T}\}$ independent. In other words, knowing the hidden state of the system at time $t$ decouples past, present, and future observations. The corresponding message-passing process has a forward-backward structure: each node $z_t$ receives a message from the observation directly connected to it, $x_t$, and from $z_{t-1}$ and $z_{t+1}$.[2]

2. The node at the first (last) time-step does not receive an incoming forward (backward) message.

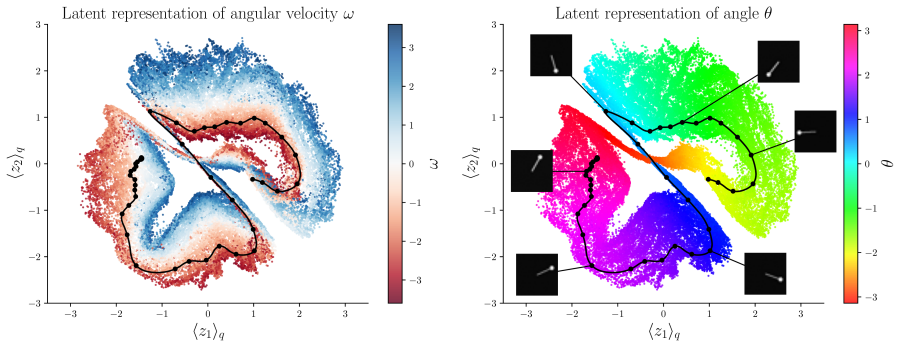



Figure 3: Inferred posterior means of learnt 2D latents for the pendulum dataset. Left: points coloured by ground-truth angular velocity. Right: points coloured by ground-truth angle. The black line tracks a single trajectory, with representative observations displayed. RAMP learns a latent representation of both dynamical state dimensions, reliably separating angles and angular velocities.

Time-series models often assume time invariance: i.e., that the transition probability $p(z_{t+1}|z_t)$, and the emission probability $p(x_t|z_t)$, do not depend on $t$. In RAMP this implies that the parameters of the neural networks propagating "forward", "backward", and "observation" messages are tied:

$$g_{t\to t+1} \equiv g_{\text{fwd}}, \quad g_{t\leftarrow t+1} \equiv g_{\text{bwd}}, \quad g_{x_t\to t} \equiv g_{\text{obs}}.$$

With this, the amortised message-passing computation becomes a bidirectional RNN, processing the sequence of observation potentials (i.e., $g_{\text{obs}}(x_t)$). However, instead of arbitrary hidden-state activations, these RNNs communicate probabilistic beliefs over $z_t$ represented by natural parameters.

We trained a RAMP model on image observations generated by a nonlinear dynamical system. We chose a simple physical pendulum with the dynamics $\dot{\theta} = \omega$; $\dot{\omega} = -\sin(\theta)$, where $\theta$ is the angle of the pendulum and $\omega$ its angular velocity. The observation $x(t)$ at each time was taken to be a rendered image of a pendulum at angle $\theta(t)$. Two sources of nonlinearity in this problem pose a challenge: the mapping between angle and image, and the dynamical law of the hidden state. The first requires recognition factors $g_{\text{obs}}$ expressive enough to extract angle information from the image. But angle alone is not enough to make past, future and present conditionally independent. Evolution depends on the angular velocity, which cannot be read from a static frame. Thus the model must learn to infer it by temporal integration. This makes the ability to learn nonlinear transformations between latent variables essential. As the underlying system is nonlinear, linear state-space methods typically rely on a higher-dimensional latent embedding(Lusch et al., 2018; Brunton et al., 2021; Nayak et al., 2025; Hromadka et al., 2025; Johnson et al., 2016). When limited to the true two-dimensional latent space, both Contrastive Predictive Coding (CPC; van den Oord et al., 2018) and SVAE failed to reliably recover angular velocity information. In contrast, RAMP recovered the phase space of the system within a two-dimensional latent space (Table 1). This embedded phase-space can be directly visualised: Fig. 3 shows posterior means inferred by a trained model for all frames in the dataset, coloured by the corresponding true angle and/or angular velocity.[3]

Table 1: $R^2$ values (top: train; bottom: test) Kernel Ridge Regression from inferred latents to ground truth latents.

| | $\cos\theta$ | $\sin\theta$ | $\omega$ |
|---|---|---|---|
| RAMP | 0.9868<br>0.9857 | 0.9434<br>0.9459 | 0.7127<br>0.6444 |
| SVAE | 0.9983<br>0.9908 | 0.9976<br>0.9882 | 0.2021<br>-0.1856 |
| CPC | 0.8131<br>0.8047 | 0.1208<br>0.1001 | 0.0396<br>0.0097 |

3. The use of ground truth hidden state is for visualization alone; the model is trained only from image observations.

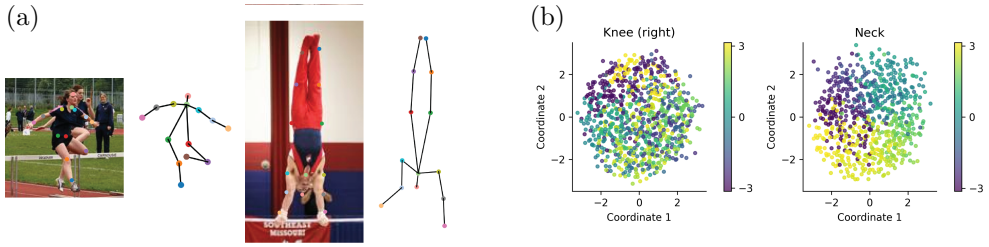



Figure 4: Pose reconstruction. (a) Images from LSP (left) were used to create simplified stick-figure representations, and local patches extracted based on labelled joint positions (coloured circles). By regressing the inferred posterior means at each joint to edge orientations, a sketch of the underlying pose was reconstructed (right). (b) 2D projections of the 6D posterior means inferred at the right knee (left) and neck (right), coloured by the angle (in radians) of the edge connecting the joint to the right ankle (left) and right hip (right), respectively. Each dot represents a different image.

### 5.3. Pose Estimation

We next considered the problem of human pose estimation, where dependencies are tree-structured in space rather than time. The goal is to infer body configuration from real-world observations. Since the body forms a tree-like structure with joints connected by limbs, we used RAMP to model their dependencies and estimate pose. The graph expresses the kinematic layout of the body. Each node represents a joint, with edges drawn between nodes whose joints are connected by limbs; for example, the left knee node has an edge to the left hip node and the left ankle node (see Fig. 4a). Each node is associated with its own local observation: a small $32 \times 32$ image patch centred on the associated joint. Thus, RAMP does not observe the entire image nor the joint locations, just their local appearances. The model must estimate pose by piecing together image fragments.

We extracted naturalistic poses from the Leeds Sports Pose (LSP) benchmark (Johnson and Everingham, 2011). Due to variation in body pose and viewpoint, joints are often occluded, making knowledge of body structure important for effective inference. For simplicity, each image was converted to a stick-figure representation, with black lines drawn between joints on a white background. Analysis of the latent variables inferred by RAMP revealed a strong correlation between the inferred posterior means at each joint and the orientations of the edges connecting that joint to its neighbours (Appendix C). This information, which is critical for pose estimation, ensures that the observation at each joint is independent of the observations at neighbouring joints; the information that is shared by a joint and its neighbours is the orientation of the edges that connect them. Edge orientation encoding was also observed in low-dimensional projections of the inferred posterior means at each joint, obtained by multidimensional scaling (Figure 4b; additional examples in Appendix C). Based on the orientations encoded at each joint, we reconstructed a sketch of the underlying pose (Figure 4a). Despite the left ankle being occluded in the left image (modelled as a missing observation), RAMP uses the learned tree-structured prior to accurately infer the pose of the left lower leg.

## 6. Discussion

Statistical regularity is key to learning latent representations. RAMP exploits this signal to learn a complex, non-linear latent tree. It does so by constructing latents that each model a different correlation pattern in the data, but through beliefs that are coupled by a coherent inference process.

The tree structure of the underlying graphical model, as assumed in this work, is essential both to ensure identifiability, and to guarantee the coherence of inference. Nevertheless, belief propagation on *loopy* graphs has been demonstrated to be effective empirically, and some convergence properties are known (Pearl, 1988; Murphy et al., 1999; Yedidia et al., 2000; Wainwright et al., 2001). Future work will explore the possibility of adapting RAMP to learn loopy graphical models.

## 7. Acknowledgments

This work was supported by the Gatsby Charitable Foundation (GAT3850, GAT4058), the Simons Foundation (SCGB543039) and the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/S021566/1 for the UCL Centre for Doctoral Training in Foundational Artificial Intelligence.

## References

Ebru Arisoy, Abhinav Sethy, Bhuvana Ramabhadran, and Stanley Chen. Bidirectional recurrent neural network language models for automatic speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5421–5425. IEEE, 2015.

Pierre Baldi, Søren Brunak, Paolo Frasconi, Giovanni Soda, and Gianluca Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 11 1999. ISSN 1367-4803. doi: 10.1093/bioinformatics/15.11.937. URL https://doi.org/10.1093/bioinformatics/15.11.937.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 561–578, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46454-1.

Xavier Boyen and Daphne Koller. Approximate learning of dynamic models. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.

Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 0035-9246.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 35603–35620. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e7663e974c4ee7a2b475a4775201ce1f-Paper-Conference.pdf.

John Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32, 1957.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.06.042. URL https://www.sciencedirect.com/science/article/pii/S0893608005001206. IJCNN 2005.

Samo Hromadka, Kai Biegun, Lior Fox, James Heald, and Maneesh Sahani. Maximum likelihood learning of latent dynamics without reconstruction, 2025. URL https://arxiv.org/abs/2505.23569.

Alexander Ihler and David McAllester. Particle belief propagation. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5, pages 256–263. PMLR, 2009.

Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/7d6044e95a16761171b130dcb476a43e-Paper.pdf.

Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of Computer Vision and Pattern Recognition (CVPR) 2011*, 2011.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009. ISBN 9780262013192.

Thibaut Lienart, Yee Whye Teh, and Arnaud Doucet. Expectation particle belief propagation. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/a00e5eb0973d24649a4a920fc53d9564-Paper.pdf.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *UAI*, pages 362–369, 2001.

Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1561–1570, 2017. doi: 10.1109/CVPR.2017.170.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, page 467–475, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606149.

Indranil Nayak, Ananda Chakrabarti, Mrinal Kumar, Fernando L Teixeira, and Debdipta Goswami. Temporally-consistent koopman autoencoders for forecasting dynamical systems. *Scientific Reports*, 15(1):22127, 2025.

Radford M. Neal and Geoffrey E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 355–370. Kluwer Academic Press, 1998.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 1558604790.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B Wiltschko. A gentle introduction to graph neural networks. *Distill*, 6(9):e33, 2021.

M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.

Olga Solodova, Nick Richardson, Deniz Oktay, and Ryan P Adams. Graph neural networks gone hogwild. *arXiv preprint arXiv:2407.00494*, 2024.

E.B. Sudderth, A.T. Ihler, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I, 2003. doi: 10.1109/CVPR.2003.1211409.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. ISSN 1935-8237, 1935-8245.

Martin J Wainwright, Tommi Jaakkola, and Alan Willsky. Tree-based reparameterization for approximate inference on loopy graphs. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/9185f3ec501c674c7c788464a36e7fb3-Paper.pdf.

William I. Walker, Hugo Soulat, Changmin Yu, and Maneesh Sahani. Unsupervised representation learning with recognition-parametrised probabilistic models. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 4209–4230. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/walker23a.html.

Chunyu Wang, Yizhou Wang, Zhouchen Lin, Alan L. Yuille, and Wen Gao. Robust estimation of 3D human poses from a single image. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2369–2376. IEEE Computer Society, 2014. doi: 10.1109/CVPR.2014.303. URL https://doi.org/10.1109/CVPR.2014.303.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

John Winn and Christopher M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(23):661–694, 2005. URL http://jmlr.org/papers/v6/winn05a.html.

Jonathan S Yedidia, William Freeman, and Yair Weiss. Generalized belief propagation. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/61b1fb3f59e28c67f3925f3c79be81a1-Paper.pdf.

Byron M. Yu, Krishna V. Shenoy, and Maneesh Sahani. Expectation propagation for inference in non-linear dynamical models with Poisson observations. In *Proceedings of the Nonlinear Statistical Signal Processing Workshop*. IEEE, 2006.

Yixiu Zhao and Scott Linderman. Revisiting structured variational autoencoders. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42046–42057. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zhao23c.html.

# A. Model Definition and Free Energy

We briefly review the RPM of Walker et al. (2023) and tree-based message passing, and then provide additional details about the RAMP model definition, free energy, and interior variational bound. Finally, we describe the extension of RAMP to general factor trees.

## A.1. RPM Review

Suppose we have $N$ observations of a collection of $P$ variables $\mathcal{X} = \{x_1 \dots x_P\}$. Label each observed value $\mathcal{X}^{(n)} = \{x_1^{(n)} \dots x_P^{(n)}\}$, and the complete data set $\mathbb{X}^{(N)} = \{\mathcal{X}^{(1)}, \dots \mathcal{X}^{(N)}\}$. We seek to learn a model for these data in which the dependence amongst the observations in $\mathcal{X}$ is captured by one or more latent variables $\mathcal{Z} = \{z_1 \dots z_K\}$. In other words, the model should render observations conditionally independent given the latents, implying a joint distribution of the form

$$p(\mathcal{Z}, \mathcal{X}) = p(\mathcal{Z}) \prod_p p(x_p|\mathcal{Z}) \,. \tag{S1}$$

The usual (directed) generative modelling approach is to parametrise each of the factors in Eq. (S1) and fit these parameters by, for example, maximising the likelihood on the data $\mathbb{X}^{(N)}$.

Now, by Bayes rule each conditional could instead have been written in the recognition form $p(x_p|\mathcal{Z}) = p(\mathcal{Z}|x_p)p(x_p)/p(\mathcal{Z})$. However, this is not helpful for a fully parametric model: explicitly parametrising $p(x_p)$ may limit the model marginals to simple distributions; and, moreover, the parameters of the recognition conditionals $p(\mathcal{Z}|x_p)$ must be constrained to ensure consistency with the marginals $p(x_p)$ and $p(\mathcal{Z})$.

The insight of Walker et al. (2023) was to instead define a *semiparametric* model, taking $p(x_p)$ to be a nonparametric summary of the corresponding observed distribution $p_{\text{emp}}(x_p)$, and setting the normalising denominator in the Bayes rule expression to be the corresponding mixture of recognition factors. Following those authors' use of the symbol $f$ for the parametrised recognition factors (which map from observations to parametric—typically exponential family—distributions on the latent) and $F$ for the normaliser, the RPM joint is

$$p(\mathcal{Z}, \mathcal{X}) = p(\mathcal{Z}) \prod_p \frac{f_p(\mathcal{Z}|x_p)}{F_p(\mathcal{Z})} p_{\text{emp}}(x_p) \qquad \text{with} \qquad F_p(\mathcal{Z}) = \int dx_p \, f_p(\mathcal{Z}|x_p) p_{\text{emp}}(x_p) \,. \tag{S2}$$

As $F$ is defined in terms of $f$ and $p_{\text{emp}}$, the only parametrised distributions in this model are the latent prior $p(\mathcal{Z})$ and recognition factors $\{f_p(\mathcal{Z}|x_p)\}_{p=1}^P$; hence the name "recognition-parametrised model".

As Walker et al. (2023) point out, the RPM is well defined for many alternative nonparametric choices of $p_{\text{emp}}$. In practice, however, they—and we—limit implementations to the atomic empirical measure $p_{\text{emp}}(x_p) = \frac{1}{N} \sum_{n=1}^N \delta(x_p - x_p^{(n)})$. In this case $F$ takes the form

$$F_p(\mathcal{Z}) = \frac{1}{N} \sum_{n=1}^N f_p\Big(\mathcal{Z}|x_p^{(n)}\Big) \,. \tag{S3}$$

The RPM is a properly normalised model with a well-defined likelihood function on the parameters. This may be maximised for parameter estimation following the standard development for latent variable models (Dempster et al., 1977; Neal and Hinton, 1998). We discuss the variational free energy and approximations necessary to handle the mixture terms $F$ in Appendix A.4.

### A.2. Tree-Structured Models and Message Passing

Consider a graphical model with observed variables $\mathcal{X}$, latent variables $\mathcal{Z}$ (i.e., nodes $\mathcal{V} = \mathcal{X} \cup \mathcal{Z}$) and edge set $\mathcal{E} = \mathcal{E}_{\mathrm{zz}} \cup \mathcal{E}_{\mathrm{zx}}$. The general form of the joint is

$$p(z_{1:K}, x_{1:P}) = \frac{1}{Z} \prod_k \phi_k(z_k) \prod_p \psi_p(x_p) \prod_{(k,k') \in \mathcal{E}_{\mathrm{zz}}} \Phi_{kk'}(z_k, z_{k'}) \prod_{(k,p) \in \mathcal{E}_{\mathrm{zx}}} \Psi_{kp}(z_k, x_p)\,,$$

where $\phi_k$, $\psi_k$, $\Phi_{kk'}$ and $\Psi_{kp}$ are (typically parametrised) non-negative factors and $Z$ is a normalising constant. If the graph is tree-structured, it is possible to write the same distribution in terms of its normalised singleton and pairwise marginals,

$$p(z_{1:K}, x_{1:P}) = \prod_k p(z_k) \prod_p p(x_p) \prod_{(k,k') \in \mathcal{E}_{\mathrm{zz}}} \frac{p(z_k, z_{k'})}{p(z_k)p(z_{k'})} \prod_{(k,p) \in \mathcal{E}_{\mathrm{zx}}} \frac{p(z_k, x_p)}{p(z_k)p(x_p)}\,,$$

or, defining an arbitrary $z_k$ as the root node of the tree, in terms of directed conditionals,

$$p(z_{1:K}, x_{1:P}) = p(z_k) \prod_{k' \neq k} p(z_{k'}|\mathrm{pa}(z_{k'})) \prod_p p(x_p|\mathrm{pa}(x_p))\,.$$

Belief propagation (BP; Pearl, 1988) is an efficient message-passing algorithm to compute various quantities in the graph: these include singleton and pairwise marginals, posterior marginals and the likelihood. In particular, defining potentials at each latent $\alpha_{j\to k}(z_k) = p(\mathcal{X}_j^k|z_j)$ (with $\mathcal{X}_j^k$ representing the leaves of the sub-tree rooted at $x_j$ and not including $x_k$, as in Section 2.2.1), BP provides the recursive updates

$$\begin{aligned} \alpha_{j\to k}(z_k) &= \int dz_j\, p(z_j|z_k) \prod_{l \in \partial_j \setminus k} p(\mathcal{X}_l^j|z_j) = \int dz_j\, p(z_j|z_k) \prod_{l \in \partial_j \setminus k} \alpha_{l\to j}(z_j) && \text{if } v_j \in \mathcal{Z} \\ \alpha_{j\to k}(z_k) &= p(x_j|z_k) && \text{if } v_j \in \mathcal{X}\,. \end{aligned} \tag{S4}$$

To compute the marginal posterior $p(z_k|\mathcal{X})$, we follow every "inward" path from each $x_p$ to $z_k$, computing each message $\alpha_{r\to s}(z_s)$ for edges $(r, s) \in x_p \rightsquigarrow z_k$. Then

$$p(z_k|\mathcal{X}) = \frac{1}{L_k} p(z_k, \mathcal{X}) = \frac{1}{L_k} p(z_k) \prod_{j \in \partial_k} \alpha_{j\to k}(z_k)\,, \tag{S5}$$

where $L_k$ is easily computed by integration over the single variable $z_k$. Furthermore, $L_k = p(\mathcal{X})$ and so corresponds to the likelihood of the model parameters. Important to the development of RAMP is that this likelihood does not depend on the choice of node $k$, and so, as long as the messages $\alpha$ can be computed exactly, the same computation can be performed at any latent node to yield the same value. In fact, for reasons discussed below, RAMP optimises all of these likelihoods together. Fortunately, BP allows efficient computation of every nodewise likelihood: a single "inward-outward" pass of messages with respect to any one node in the tree (taking order $|\mathcal{V}|$ steps) is sufficient to compute them all.

In practice, exact message computation is only possible for a limited class of generative conditionals. More general (and nonlinear) dependence between variables in the model necessitates approximation (Wainwright and Jordan, 2008), often combined with Monte-Carlo or numerical integration (e.g., Ranganath et al., 2014).

### A.3. RAMP

Although the RPM formulation of Walker et al. (2023) allowed for the model to comprise multiple latents $\mathcal{Z}$ with their own conditional independence structure, the recognition parametrisation applied

only to the leaf node conditionals from $x_p$ to (possibly a subset of) $\mathcal{Z}$. The joint over variables in $\mathcal{Z}$ needed either to be tractable, or to be approximated by standard approaches from the graphical models literature (cf. Wainwright and Jordan, 2008).

The key insight of RAMP is that the nodewise BP-derived models of Eq. (S5) can each be recognition parametrised in a coherent way by directly parametrising the message passing operations of Eq. (S4) without explicit reference to the generative conditionals. That is, we can parametrise the functional $\mathcal{G}_{j\to k}$ of Eq. (6) (or—for exponential family forms—the natural parameter mapping $g_{j\to k}$, as discussed further below), in effect learning an amortised computation of the integrals needed for belief propagation. Just as the recognition factors (and prior) *define* the RPM likelihood, these parametrised messages (and priors and leaf recognition factors) *define* (an approximation to) each nodewise RAMP likelihood.

The RAMP model parameters are learnt by maximising a lower bound to the product of the nodewise likelihoods (Eq. 10), even though in principle each evaluates the same joint probability, as shown above. This design is necessitated by the recognition parametrisation of the nodewise RPMs.

Recognition factors in a single-latent RPM are optimised to infer a belief over the latent variable, capturing information that is shared amongst the different observations. Thus, a model trained on a single $z_k$-nodewise likelihood would learn amortised messages within each subtree $\mathcal{T}_j^k$ that carry information shared by variables in $\mathcal{X}_j^k$ with variables in a *different set* $\mathcal{X}_l^k$, but not necessarily the information shared amongst the variables *within* $\mathcal{X}_j^k$. This information content is a feature of the learning objective rather than of the topology of message passing. To ensure the RAMP model captures information linking two variables $x_r, x_s \in \mathcal{X}_j^k$ but not shared with any variables in $\mathcal{X}_l^k, l \neq j$, it is necessary to also optimise the nodewise RPM likelihood based at a latent node that lies along the direct path $x_r \rightsquigarrow x_s$.

In this way, each nodewise RPM likelihood is optimised to create conditional independence within the corresponding partition $\{\mathcal{X}_j^k : j \in \partial_k\}$. As shown by Lemma 2, the collection of all these nodewise observation conditional independencies is sufficient to ensure consistency with the entire tree.

### A.4. Free Energy and Interior Bound

The variational free energy (sometimes called the ELBO) is a lower bound on a model likelihood in terms of the parameters $\theta$ that define the model $p(\mathcal{Z}, \mathcal{X})$, and a variational distribution over the latent variables, $q(\mathcal{Z})$. It has the form

$$\mathcal{F}(\theta, q) = \langle \log p(\mathcal{Z}, \mathcal{X}) \rangle_q + \mathbf{H}[q] \,, \tag{S6}$$

where angle brackets indicate expectation and $\mathbf{H}[\cdot]$ is the entropy. The maximum of $\mathcal{F}$ with respect to $q$ is achieved when $q(\mathcal{Z}) = p(\mathcal{Z}|\mathcal{X})$ and equals the log-likelihood $\log p(\mathcal{X})$.

Applied to the $z_k$-nodewise RPM of Eq. (7), and dropping terms independent of $\theta$ and $q$ we have:

$$\begin{aligned}
\mathcal{F}_k(q(z_k), \theta) &\overset{+c}{=} \left\langle \log f_0^k(z_k) + \sum_{j\in\partial_k} \Big( \log f_j^k(z_k|\mathcal{X}_j^k) - \log F_j^k(z_k) \Big) \right\rangle_q + \mathbf{H}[q] \\
&= -\mathbf{KL}\big[q \big\| f_0^k(z_k)\big] - \sum_{j\in\partial_k} \mathbf{KL}\big[q \big\| f_j^k(z_k|\mathcal{X}_j^k)\big] + \sum_{j\in\partial_k} \mathbf{KL}\big[q \big\| F_j^k(z_k)\big] \,,
\end{aligned}$$

where $\mathbf{KL}[\cdot\|\cdot]$ is the Kullback-Leibler (KL) divergence.

Direct optimisation of this free energy is made difficult by the $F_j^k(z_k)$ factors in two ways. First, they complicate the form of the optimal variational distribution $q$. Second, even if $q$ is restricted to a tractable class, as is common in variational learning, exact computation of the final KL term remains challenging.

While Walker et al. (2023) also investigated sampling and numerical approximation techniques for the KL computation, here we adopt a variant of those authors' "interior variational bound". Introducing variational functions $\tilde{f}_j^k(z_k)$ we have

$$\begin{aligned}\left\langle \log F_j^k(z_k) \right\rangle_q &= \left\langle \log F_j^k(z_k) \frac{\tilde{f}_j^k(z_k)}{q(z_k)} \right\rangle_q + \left\langle \log \frac{q(z_k)}{\tilde{f}_j^k(z_k)} \right\rangle_q \\ &\leq \log \Gamma_j^k + \left\langle \log \frac{q(z_k)}{\tilde{f}_j^k(z_k)} \right\rangle \qquad \text{with } \Gamma_j^k = \int dz_k \, F_j^k(z_k) \tilde{f}_j^k(z_k) \,. \end{aligned} \tag{S7}$$

Thus we obtain a lower bound on the free energy

$$\mathcal{F}_k(q(z_k), \theta) \geq \widetilde{\mathcal{F}}_k\Big(q(z_k), \theta, \{\tilde{f}_j^k(z_k)\}\Big) = \left\langle \log f_0^k(z_k) \right\rangle_q + \sum_{j \in \partial_k} \left( \left\langle \log f_j^k(z_k | \mathcal{X}_j^k) + \log \frac{\tilde{f}_j^k(z_k)}{q(z_k)} \right\rangle_q - \log \Gamma_j^k \right) + \mathbf{H}[q] \,. \tag{S8}$$

Now, if $\tilde{f}_j^k(z_k)$ is chosen to have the form of an unnormalised distribution in the same family as $f_0^k(z_k)$ and $f_j^k(z_k|\mathcal{X}_j^k)$, then the distribution $q(z_k)$ which maximises $\tilde{\mathcal{F}}_k$ will lie within that same tractable family, while $\Gamma_j^k$ becomes a straightforward sum of exponential family normalisers.

The bound of Eq. (S8) will be tight when $\tilde{f}_j^k(z_k) \propto q(z_k)/F_j^k(z_k)$. Furthermore, as $F_j^k(z_k)$ is an average posterior, we expect its optimal form over learning to approach the prior $f_0^k(z_k)$. Thus, rather than optimising $\tilde{\mathcal{F}}_k$ with respect to parametric functions $\tilde{f}_j^k(z_k)$ we take the approach of setting $\tilde{f}_j^k(z_k) = q(z_k)/f_0^k(z_k) \equiv \hat{f}_j^k(z_k)$ after each computation of the optimal variational posteriors $q^*(z_k)$. This choice yields the bound

$$\mathcal{F}_k(q(z_k), \theta) \geq \widetilde{\mathcal{F}}_k\Big(q(z_k), \theta, \{\hat{f}_j^k(z_k)\}\Big) = \left\langle \log f_0^k(z_k) \right\rangle_q + \sum_{j \in \partial_k} \left( \left\langle \log \frac{f_j^k(z_k|\mathcal{X}_j^k)}{f_0^k(z_k)} \right\rangle_q - \log \widehat{\Gamma}_j^k \right) + \mathbf{H}[q]$$

$$\text{with } \widehat{\Gamma}_j^k = \int dz_k \, F_j^k(z_k) \hat{f}_j^k(z_k) \,,$$

for which the optimal $q$ has the form

$$q^*(z_k) \propto f_0^k(z_k) \prod_{j \in \partial_k} \frac{f_j^k(z_k|\mathcal{X}_j^k)}{f_0^k(z_k)} \,. \tag{S9}$$

### A.4.1. Exponential-Family Messages

A practical choice, assumed throughout this paper, is for all the parametrised potentials over latent variable $z_k$ to lie in the same exponential family with sufficient statistic function $T^k$ and density

$$f(z_k) = e^{\eta^\intercal T^k(z_k) - \Phi^k(\eta)}$$

with respect to a suitable base measure. Here $\eta$ is the natural parameter and $\Phi^k$ is the log-normalising function. The family (and therefore functions $T^k$ and $\Phi^k$) may differ for different $z_k$.

Now, multiplying the relevant factors (as in Eq. S9) reduces to summing their natural parameters $\eta$. Note that in both Eqs. (6) and (S9), the recognition factors always appear divided by the corresponding prior. In practice, we parametrise this ratio directly, letting $\eta_j^k$ represent the difference in natural parameters between the recognition factor $f_j^k(z_k|\mathcal{X}_j^k)$ and the prior $f_0^k(z_k)$. With this choice, Eq. (S9) becomes (c.f. Eq. (11)):

$$\eta_k^* = \eta_0^k + \sum_{j \in \partial_k} \eta_j^k(\mathcal{X}_j^k) = \eta_0^k + \sum_{j \in \partial_k} \eta_{j \to k} \tag{S10}$$

Working in natural parameter space also makes it easier to parametrise the functionals $\mathcal{G}_{j\to k}$—these should transform natural parameters of $z_j$ into natural parameters of $z_k$.

In our experiments, the prior terms $f_0^j(z_j)$ were fixed for each model. Therefore rather than including a constant $\eta_0^j$, we define $\mathcal{G}_{j\to k}$ by a parametric function $g_{j\to k}$ such that (as in Eq. (11)):

$$\eta_{j\to k} = g_{j\to k}\left(\sum_{l\in\partial_j\setminus k} \eta_{l\to j}\right)$$

The constant offset of $\eta_0^j$ is then implicit in the effective "bias" of the neural network defining $g_{j\to k}$.

### A.5. Complete Algorithm

The overall structure of RAMP learning is given by Algorithm 1, along with pseudocode for the free energy computation in Listing 1. Learning follows the gradient of the free energy with respect to the parameters defining the amortised messages[4] Messages are computed by inward-outward propagation along the edges of the tree defining the model. These messages are sufficient to compute $q_k(z_k)$ at each node, and so evaluate the (interior variational bound to the) free energy.

Efficient computation of the messages and free energy for exponential family beliefs requires an exponential family object that supports: **a)** Multiplicative combination (i.e., summing natural parameters); **b)** Computation of the KL divergence between two distributions from their natural parameters, $\mathsf{KL}[\eta_1\|\eta_2]$; **c)** Computation of the log-normaliser function, $\Phi(\eta)$; and **d)** automatic differentiation through all of the above methods.

**Algorithm 1** RAMP

```
Initialise the parameters of the message-passing networks θ
for t in T training iterations do
  Sample a minibatch of observations 𝕏^(t) ⊆ 𝕏
  for all samples 𝒳^(n) ∈ 𝕏^(t) do
    Compute messages for data sample 𝒳^(n)
  end for
  Compute the free energy ℱ given the batch of messages (see Listing 1)
  Compute the gradient ∇_θ ℱ and update the parameters
end for
```

4. And potentially parameters of $f_0^k(z_k)$; although these were held to fixed values in our experiments without loss of generality.

```
def node_free_energy(prior, messages):
    # prior:    (unbatched)      natural parameters of f_0
    # messages: batch of [J x N] natural parameters of f_j/f_0
    # q:        batch of [N]     natural parameters of q
    # aux:      batch of [J x N] natural parameters of fhat

    f[j]   = prior + messages[j]
    q      = prior + messages.sum(axis=0)
    aux[j] = messages.sum(axis=0)

    gamma[j,n,m]  = Phi(aux[j,n]+f[j,m]) - Phi(aux[j,n]) - Phi(f[j,m])
    logGamma[j,n] = gamma[j,n,n] - logsumexp(gamma[j,n])

    F = -kl(q,prior) -kl(q,f) + logGamma
    return F

def free_energy(params, model, X):
    # X: batch of N samples of all P observations

    # compute all the messages (parallelized over batch),
    # and get the marginal priors at each z_k:
    messages = model.apply(params, X)
    priors   = model.apply(params, method='prior')

    F[k] = node_free_energy(priors[k], messages_into_k)

    return F.mean()
```

Listing 1: Pseudocode for the free energy calculation

### A.6. RAMP For General Factor Trees

Section 3 presented the form of RAMP for tree-structured graphical models with pairwise factors. Here, we give a brief overview of how RAMP may be generalised to arbitrary tree-structured factor graphs.

A factor graph is an undirected bipartite graph $\mathcal{T} = (\mathcal{V}, \mathcal{U}, \mathcal{E})$ where $\mathcal{V}$ is a set of *variable* nodes, $\mathcal{U}$ is a set of *factor* nodes, and $\mathcal{E}$ is a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{U}$. As before, $\mathcal{V}$ is partitioned into observation nodes $\mathcal{X}$ and latent nodes $\mathcal{Z}$, and we assume that $\mathcal{T}$ has a tree structure with observations on the leaves. The graphical structure defines a factorisation of the joint distribution (over latents and observations) according to:

$$p(\mathcal{X}, \mathcal{Z}) = \frac{1}{Z} \prod_{u \in \mathcal{U}} \phi_u(\partial_u),$$

where $\phi_u$ are factor potentials, $\partial_u$ is the set of variable nodes connected to the factor $u$, and $Z$ is a normalisation constant. If $\deg(u) = k$, then the factor potential $\phi_u$ is a $k$-way function. An example is shown in Fig. S1.

Belief propagation in factor graphs involves passing variable-to-factor and factor-to-variable messages. The message from a variable node $v$ to one of its neighbouring factors $u \in \partial_v$ is given by the belief formed at $v$ by aggregating the messages from all the *other* factors connected to it. Note that this message is a belief *over the variable* $v$:

$$m_{v \to u}(v) \propto \prod_{u' \in \partial_v \setminus u} m_{u' \to v}(v). \tag{S11}$$

The message from a factor node $u$ to one of its neighbouring variables $v$ is the belief over $v$ derived by marginalising the factor potential with respect to the messages to $u$ from the *other* nodes connected to it:

$$m_{u \to v}(v) \propto \int \left( \prod_{v' \in \partial_u \setminus v} dv' \right) \phi_u(\partial_u) \prod_{v' \in \partial_u \setminus v} m_{v' \to u}(v'). \tag{S12}$$

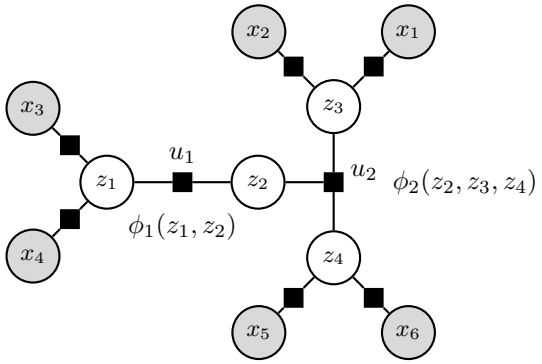



Figure S1: A tree-structured factor graph with 7 pairwise factors and one 3-way factor. Two factor nodes are labelled with their corresponding potentials.

The factor-based RAMP model is based on amortisation of the factor-to-variable messages of Eq. (S12). For a $k$-way factor $u$, these could be written generically as a $(k-1)$-way functional that transforms the collection of all incoming messages from variables besides $v$ into a belief over $v$ (c.f. Eq. (6)). The form of this mapping in a general factor graph is potentially arbitrary; in particular, it is not restricted to a multiplicative combination of the incoming messages. In practice, for natural-parameter messages, the mapping may be implemented by a neural network acting on the concatenated incoming belief natural parameters to produce the output belief parameters.

To give a concrete example, consider the nodewise RPM at $z_2$ in the graph of Fig. S1. This RPM has two recognition factors, one for $\{x_3, x_4\}$, and the other for $\{x_1, x_2, x_5, x_6\}$, corresponding to the sets of all observation nodes downstream from each factor $z_2$ connects to (i.e., $u_1$ and $u_2$). The recognition factors are then of the form:

$$f^2_{u_1}(z_2|x_3, x_4) = \mathcal{G}_{u_1 \to z_2}(\mathsf{P}_1(z_1|x_3, x_4))$$
$$f^2_{u_2}(z_2|x_1, x_2, x_5, x_6) = \mathcal{G}_{u_2 \to z_2}(\mathsf{P}_3(z_3|x_1, x_2), \mathsf{P}_4(z_4|x_5, x_6))$$

## B. Details of Experiments

### B.1. Gaussian Tree

The hierarchical linear-Gaussian model is defined by the following sequential sampling process:

$$z_1 \sim \mathcal{N}(0, 1)$$
$$v|\mathrm{pa}(v) \sim \mathcal{N}(a_v \cdot \mathrm{pa}(v), \sigma_v^2),$$

where $z_1$ is the node at the root of the tree, $v$ is any other node in the tree (either a leaf $x_i$ or an internal node $z_i$) and $\mathrm{pa}(v)$ is the parent of $v$.

This model definition holds for general trees. In our experiments, we used a complete binary tree of depth 4 (including the root). We take the leaves to be observations and the rest of the nodes to be latent variables, yielding 8 observations and 7 latents, as follows:

We take the conditional factors to be spatially uniform across the tree, with $a_v \equiv a = 1$ and $\sigma_v^2 \equiv \sigma^2 = 1$.

The linear-Gaussian links imply that the full joint distribution $p(\mathbf{z}, \mathbf{x})$ is multivariate Gaussian. This multivariate distribution has mean zero and a covariance matrix with a recursive structure, reflecting that of the tree (Fig. S3).

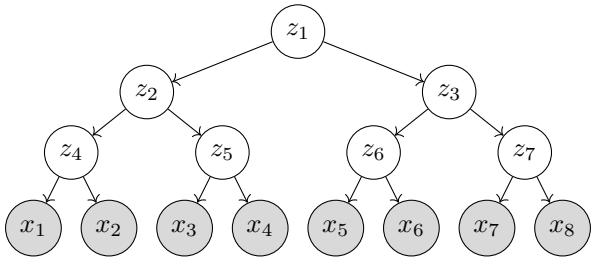



Figure S2: The tree structure of for the hierarchical linear-Gaussian model of Section 5.1

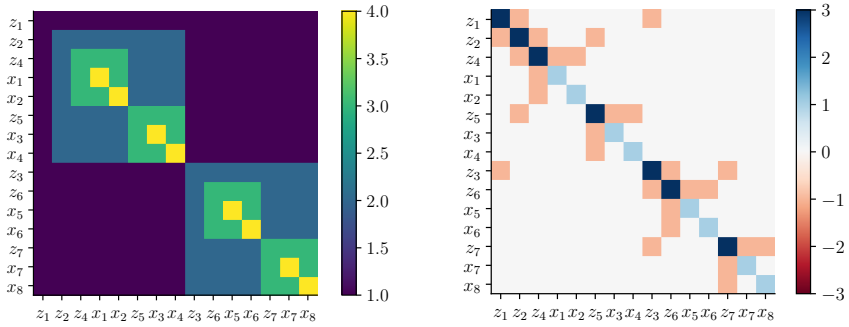



Figure S3: The covariance (left) and precision (right) matrices of the hierarchical linear-Gaussian model of Section 5.1 (with $a = 1$ and $\sigma^2 = 1$). Non-zero off-diagonal elements in the precision matrix correspond to edges in the graphical model.

By choosing an appropriate permutation of the variables, the covariance and precision matrices can be written in block form:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L_{zz}} & \mathbf{L_{zx}} \\ \mathbf{L_{xz}} & \mathbf{L_{xx}} \end{bmatrix}, \qquad \mathbf{\Lambda} = \mathbf{L}^{-1} = \begin{bmatrix} \mathbf{\Lambda_{zz}} & \mathbf{\Lambda_{zx}} \\ \mathbf{\Lambda_{xz}} & \mathbf{\Lambda_{xx}} \end{bmatrix},$$

which, using the rules of Gaussian conditioning, give a simple expression for the posterior $p(\mathbf{z}|\mathbf{x})$. This posterior is also a multivariate Gaussian, with the statistics (in terms of mean and covariance):

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z};\ -\mathbf{\Lambda}_{\mathrm{zz}}^{-1}\mathbf{\Lambda}_{\mathrm{zx}}\mathbf{x}, \mathbf{\Lambda}_{\mathrm{zz}}^{-1}\big)$$

Naturally, this posterior depends on the prior over $\mathbf{z}$. However because these are all latent variables, they can be rescaled arbitrarily, while compensating by the likelihood terms. From the perspective of RAMP, we have kept the marginal priors ($f_0^k(z_k)$) of all latent variables fixed at a standard Gaussian with zero mean and unit variance. Therefore, in order to quantitatively compare the RAMP results with those of the analytic theory, we computed a re-scaled version of the matrix $\mathbf{L}$ (and $\mathbf{\Lambda}$), reflecting the same parametrisation. For that, define the scaling matrix $\mathbf{S}$ to be the diagonal matrix with:

$$S_{ii} = \begin{cases} \frac{1}{\sqrt{L_{ii}}} & i \in \mathbf{z} \\ 1 & i \in \mathbf{x} \end{cases},$$

and set

$$\widetilde{\mathbf{L}} = \mathbf{SLS}^{\top}$$

Respectively we define $\widetilde{\mathbf{\Lambda}} = \widetilde{\mathbf{L}}^{-1}$ to get the re-scaled posterior:

$$\widetilde{p}(\mathbf{z}|\mathbf{x}) = \mathcal{N}\Big(\mathbf{z};\ -\widetilde{\mathbf{\Lambda}}_{\mathrm{zz}}^{-1}\widetilde{\mathbf{\Lambda}}_{\mathrm{zx}}\mathbf{x}, \widetilde{\mathbf{\Lambda}}_{\mathrm{zz}}^{-1}\Big) \tag{S13}$$

The exact posterior (in either parametrisation) has a non-diagonal covariance matrix, reflecting posterior correlations between different latent variables. RAMP only gives explicit access to a set of marginal posteriors, with the interactions between different latents being implicitly encoded in the message-passing factors. To demonstrate that these interactions are learned by RAMP, we also compared its results to those achieved by the "mean-field" variational posterior for the true generative model, i.e. the fully factored distribution $q(\mathbf{z}) = \prod_i q(z_i)$ that minimises $\mathbf{KL}[q(\mathbf{z})\|\widetilde{p}(\mathbf{z}|\mathbf{x})]$. This mean-field solution can be computed analytically for multivariate Gaussians. It recovers the correct posterior mean, but (generically) underestimates the marginal variances:

$$\widetilde{p}_{\mathrm{mf}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}\Big(\mathbf{z};\ -\widetilde{\mathbf{\Lambda}}_{\mathrm{zz}}^{-1}\widetilde{\mathbf{\Lambda}}_{\mathrm{zx}}\mathbf{x}, \mathrm{diag}(\widetilde{\mathbf{\Lambda}}_{\mathrm{zz}})^{-1}\Big) \tag{S14}$$

Figure 2 shows the correlation between the inferred posterior means and the true latents, for each one of the models: RAMP, variational mean-field, and exact inference. The mean-field and exact inference models predict constant marginal variances (independent of the observation). While this constraint could be imposed in RAMP, we have kept the parametrisation general, and so for the comparison of marginal variances we have averaged the posterior variances predicted by RAMP, per latent variable, over the dataset.

Hyperparameter settings for the experiment can be found in Table S1.

| Dataset | $N$ | Message-passing networks | | Prior | LR | Batch | Train Iters. | Optim. |
|---|---|---|---|---|---|---|---|---|
| | | Architecture | Nonlinearity | | | | | |
| Linear tree | 10,000 | [32,32] | tanh | $\mathcal{N}(0,1)$ | 0.001 | $N$ | 500 | Adam |
| Nonlinear tree | 10,000 | [64,64] | tanh | $\mathcal{N}(0,1)$ | 0.001 | $N$ | 500 | Adam |

Table S1: Hyperparameters for the linear and nonlinear tree experiments (Section 5.1 and Appendix B.3

### B.1.1. Black Box Variational Inference

In addition to comparing RAMP results to exact inference (which requires knowledge of the true generative model), we also compared to a Black-Box Variational Inference (BBVI) baseline (Ranganath et al., 2014). BBVI enables (variational) inference in analytically intractable models, by following a stochastic, samples-based, estimate of the gradient of the free energy. The method requires a choice of the variational posterior family that is easy to sample from, as well as the ability to evaluate the log joint-probability over latent and observed variables.

The original method focused on inference in a known generative model. However, it can naturally be extended to learning model parameters, either by EM, or by jointly optimising variational *and* model parameters by gradient ascent on the free energy. Here we take the latter approach which more closely resembles learning in RAMP. We defined the model parametrisation to follow the graphical structure of the true generative model (Fig. S2), but with conditional factors defined as

$$v = f_v(\mathrm{pa}(v)) + \xi_v,$$

where $f_v$ is a neural-network function and $\xi_v$ is independent Gaussian noise with unit variance. Each $f_v$ was parametrised as a (separate) neural network with one hidden layer of 16 units. The variational distribution was taken to be a fully-factorised Gaussian, $q(\mathbf{z}) = \prod_{i=1}^{7} q_i(z_i)$ with each $q_i$ a univariate Gaussian.

These choices were made as to resemble the "structural" prior of RAMP: that the joint factorises according to the tree structure, that no specific functional/distributional form of the factors themselves is assumed *a priori*, and that the marginal posteriors should be (approximated as) Gaussian distributions.

BBVI was trained with full-batch iterations. We found that BBVI was slower to train compared to RAMP. To reach convergence, we trained for 2,500 training iterations (Fig. S9). For each training iteration and for each example, 100 samples from the (current) variational posterior were used to form a Monte-Carlo estimate of the free energy.

The same setup was used for the nonlinear tree experiment, with 3,000 training iterations.

### B.2. Structural Misspecification with Perturbed Trees

To create perturbed trees, we started from the true tree (Fig. S2), and applied a random sequence of $k$ Nearest Neighbour Interchange (NNI) steps. We repeated this procedure for $k = 0, \ldots, 4$, with 10 random seeds for each value of $k > 0$ (note that $k = 0$ is, by construction, the true unperturbed tree). An illustration of the procedure with $k = 2$ is depicted in Fig. S4.

Because latents are only defined relative to the tree structure defined on the observations (consider, for example, a perturbation that would switch $z_4$ with $z_5$ – clearly the 'perturbed' tree is identical to the original one, up to relabeling of the latents), in each perturbed tree we have matched each latent to its best match in the original tree. For a given latent $z'_i$ in the perturbed tree, we have computed for each latent $z_j$ in the true tree the partition mismatch number $m(i, j)$ that counts how many observations are routed, in the perturbed tree, to the wrong sub-tree at $z'_i$, relative to the partition induced by $z_j$ in the true tree. We then identified $z'_i$ with $z_j$ that minimised $m(i, j)$. Note that this matching procedure is not necessarily a permutation – indeed, multiple latents in the perturbed tree could be matched to a single latent in the unperturbed tree. The calculation of partition mismatch index is illustrated in Fig. S5.

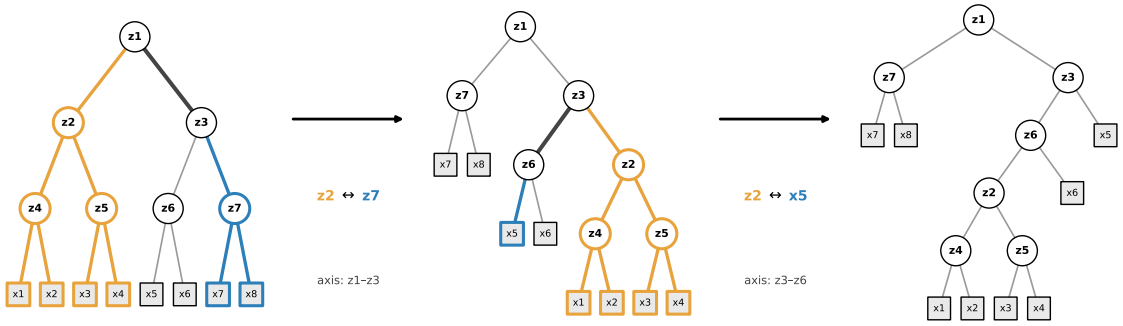



Figure S4: An example of a perturbed tree created by $k = 2$ NNI moves

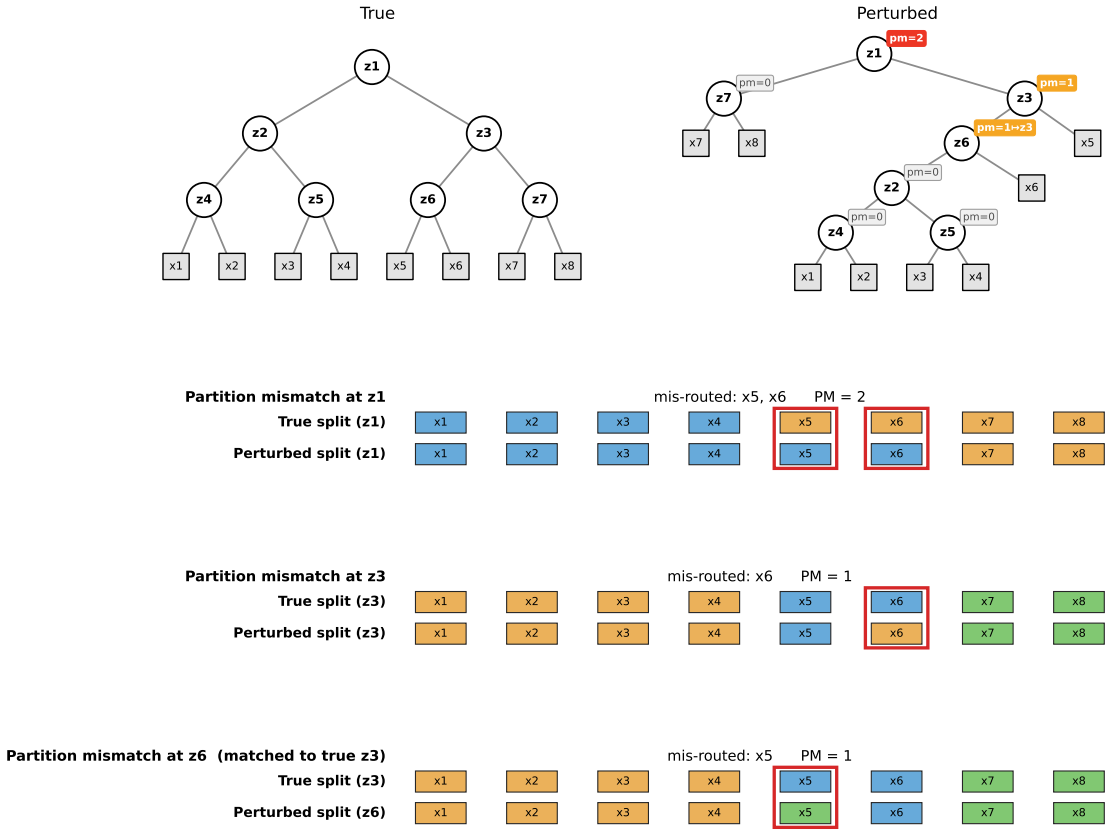



Figure S5: The perturbed tree from Fig. S4, with the partition mismatch index of each latent. Note that in the perturbed tree, both $z_3$ and $z_6$ are matched to the original latent $z_3$.

### B.3. Nonlinear Tree

We next examine a similar example, but with each variable sampled from a *nonlinear* transformation of its parent (a clipped inverse) with additive Gaussian noise.

For the non-linear tree model, we generated data from a hierarchical sampling process with the same tree structure, but with the conditional mean of each node given by a (noisy) non-linear transformation of its parent variable:

$$v = f(\mathrm{pa}(v)) + \sigma_v \xi_v.$$

We ensured that the latent variables all have marginal priors with zero mean and unit variance, by "scaling" the non-linearities.[5] Specifically, we set:

$$v = \sqrt{1-\alpha}\frac{f(\mathrm{pa}(v)) - \mu_{\mathrm{pa}(v)}}{\sigma_{\mathrm{pa}(v)}} + \sqrt{\alpha}\xi_v,$$

where $\mu_{\mathrm{pa}(v)}, \sigma_{\mathrm{pa}(v)}$ are the mean and standard-deviation of $\mathrm{pa}(v)$ (computed empirically over the sampling batch), $\xi_v$ is independent standard Gaussian noise, and $\alpha$ roughly controls the signal-to-noise ratio in $v$.

In the experiment we set $f(x) = \mathrm{clip}(\frac{1}{x}, [-15, 15])$ and $\alpha = 0.2$.

5. note that the marginal prior is *not* a Gaussian: we simply standardized the first two moments

For the non-linear model, the joint probability is not multivariate Gaussian, and exact inference is intractable. We therefore compared RAMP to Black-Box Variational Inference (BBVI), as well as to a Gaussian approximation which constructs an approximate posterior in the same way as in Eq. (S13), using the $2^{\text{nd}}$ order statistics of the model. Note that this Gaussian approximation is a "supervised" baseline, since access to the full covariance matrix (including the latent variables) is required.

The hyperparameter settings for the experiment can be found in Table S1. The BBVI model was the same as that described for the linear tree. Results from this experiment are reported in Appendix C.2.

### B.4. Pendulum

The state of a simple physical pendulum is characterized by its angle and angular velocity. We sampled trajectories by numerically integrating the dynamical system:

$$\dot{\theta} = \omega$$
$$\dot{\omega} = -\sin(\theta)$$

We set $T_{\text{sim}} = 10$ and $dt = 0.001$. Integration was performed by the `scipy.integrate.odeint` function with default settings. The initial conditions were sampled uniformly $\theta \in [-\pi, \pi]$, $\omega \in [-3, 3]$. Note that this generates both oscillating and rotating trajectories of the pendulum. For rotating trajectories, we have interpreted all angles to lie in the interval $[-\pi, \pi]$ for later analysis.

We sampled $N = 2,000$ trajectories with random initial conditions, and then sub-sampled each trajectory in intervals of 100 frames, resulting in $T = 100$ timesteps in each trajectory. For these 100 timesteps, we have rendered a $128 \times 128$ grayscale image depicting the pendulum at the corresponding angle (Fig. S6).

Figure S6: First 25 frames of one of the trajectories in the pendulum dataset

In RAMP, we set the marginal prior over each $z_t$ to a 2-dimensional Gaussian distribution (we have also explored different priors; see Appendix C.3). As explained in Section 5.2, parameters were shared for each group of "forward" ($g_{t \to t+1}$), "backward" ($g_{t \leftarrow t+1}$) and "observations" ($g_{x_t \to t}$) amortisers. Each of those was parametrised by a neural network with 1 hidden layer. We ensured that messages reflected valid Gaussian natural parameters by having the network generate the Cholesky factor of the precision matrix.

We performed a hyperparameter grid search over the width of these hidden layers and the learning rate (see also Fig. S14), with the reported run in Section 5.2 having 128 hidden units for all networks and a learning rate of 0.0005. The batch size was kept fixed across runs at 200 sequences per batch.

### B.5. Pose Estimation

Human pose estimation from a single image typically follows a two-step pipeline. In step one, the 2D position of the $N$ body joints are detected using a computer vision method. In step two, the 2D joint positions detected in step one are used to infer the 3D body pose. Step two of the pipeline is typically treated either as a supervised learning problem involving $2D$-to-$3D$ regression of the Cartesian joint coordinates (Moreno-Noguer, 2017) or as an optimization problem involving an explicit generative (camera) model of how a 3D pose projects to a 2D image (Bogo et al., 2016; Wang et al., 2014). In this work, we take a fundamentally different approach; pose estimation with RAMP

is entirely unsupervised and does not require a generative model to be specified due to recognition parameterisation.

We trained RAMP on a dataset of 10,000 poses derived from the LSP dataset (Johnson and Everingham, 2011). For each original image in the dataset, we created a simplified "stick-man" version by drawing the lines connecting adjacent joints, based on the labels. The observations for RAMP consist of local image patches centered around each joint in these (derived) images, but without global location information. An overview of the method is shown in Fig. S7.

We set the dimensionality of all the latent variables to 6, with a multivariate Gaussian prior of $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Message-passing factors between the latent variables were parametrised as fully-connected neural networks with one hidden layer of 64 units. The observation-to-latent recognition factors were parameterised as convolutional neural networks with three convolutional layers. The first layer employed 32 filters with a 3×3 kernel and a stride of 1, the second layer used 64 filters with a 3×3 kernel and a stride of 2, and the third layer used 32 filters with a 3×3 kernel and a stride of 2. The model was trained on minibatches of size 200 for 3,000 iterations with a learning rate of 0.0001.

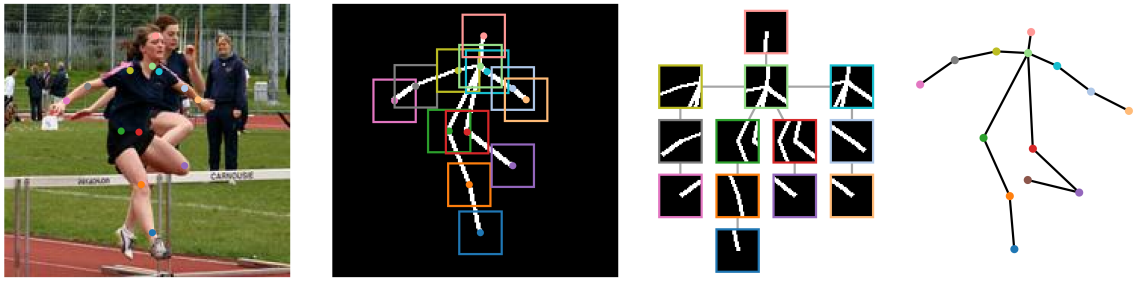


Figure S7: Pose estimation task. The LSP dataset consists of images (left) and joint positions in image space (overlaid as coloured circles for visualisation). The left ankle joint has no position label as it is occluded in this image. Each image was converted to a stick-figure representation (centre left) by drawing lines between joints. The left lower leg has not been drawn as the left ankle joint position is unknown. Patches of images (32 × 32) centred at each joint (coloured bounding boxes) were used as local joint observations (centre right). Note that the observation at the left ankle joint is missing. The posterior means inferred by RAMP were used to construct a sketch of the pose (right).

## C. Additional Experimental Results

### C.1. Linear tree

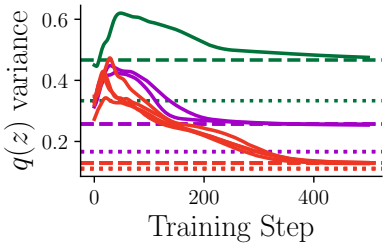



Figure S8: Posterior marginal variances (average over the dataset) inferred by RAMP, as a function of training step. Despite having no explicit model of the posterior pairwise marginals, the marginal variances learned by RAMP converge to the exact posterior variances (dashed lines), not those of the "naive" Mean-Field (dotted lines) achieved by restricting the posterior to factorise over the latents, indicating that correlations among latent variables are implicitly encoded in the model.

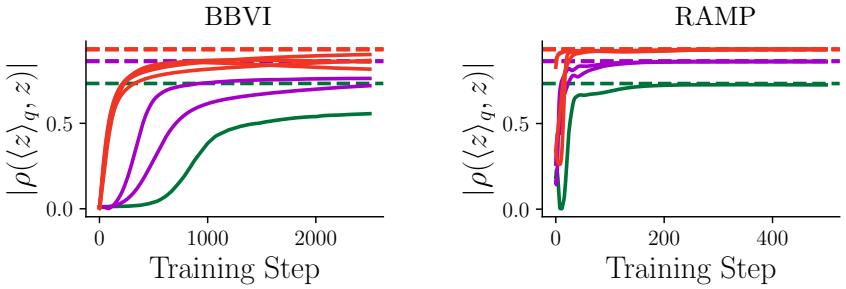



Figure S9: Pearson correlation between inferred posterior means and true latent in the linear tree model as a function of training step for the BBVI (left) and RAMP (right; panel reproduced from Fig. 2 for reference). Dashed lines denote optimal values of the exact posterior. Note the differences in training time axes, as well as the converged performance level, between BBVI and RAMP.

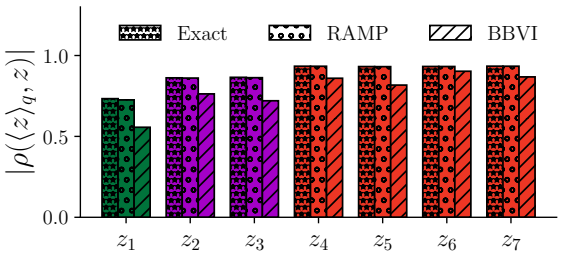



Figure S10: Pearson correlation between inferred posterior means (at the end of training) and true latents for RAMP, BBVI, and Exact inference in the linear tree model.

### C.2. Nonlinear tree

The flexible recognition of RAMP enables it to discover an appropriate representational scheme for the latent variables. In this case, the model learns to reliably represent and infer the reciprocals of the latents, $\frac{1}{z_k}$. RAMP is able to learn a better model of the latents compared to Black Box methods that depend on an explicitly parametrised generative model, and also outperforms a supervised baseline that assumes access to the covariance matrix of the true model (Fig. S11).

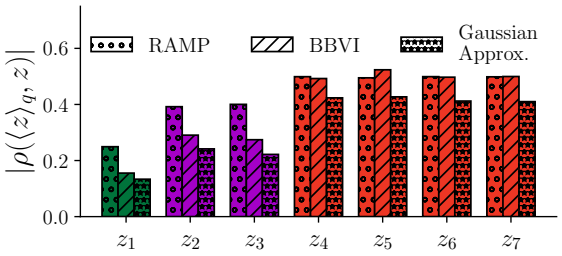



Figure S11: Hierarchical nonlinear tree-structured model. Bars show Spearman correlation between inferred means and true latents (colours indicate variable depth as in Fig. 2). RAMP learns to infer latents more accurately than either BBVI, or a Gaussian approximation based on the full covariance matrix (requiring knowledge of the true latents), particularly for latents "further away" from the observations.

The sampling process of the nonlinear tree model (Appendix B.3) suggests that a natural solution for the problem would be to represent $z^{-1}$ (instead of $z$). Note that at least in the last layer of latent variables, the observations are indeed linear-Gaussian in $z^{-1}$. We hypothesized that the flexible

recognition scheme used in RAMP might pick up this representation. Indeed, as shown in Figs. S12 and S13 we found high *Pearson* correlation between the posterior means inferred by RAMP and the (clipped) *reciprocals* of the true latents.

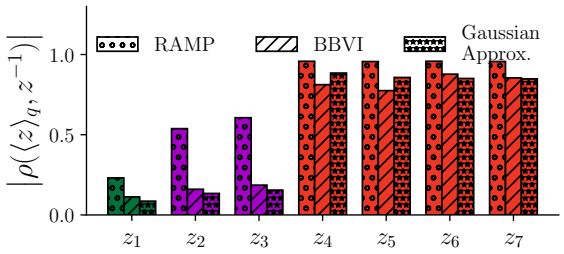



Figure S12: The Pearson correlation between inferred posterior means and the reciprocals (clipped at $[-15, 15]$) of the true latents for RAMP, BBVI, and the Gaussian approximation.

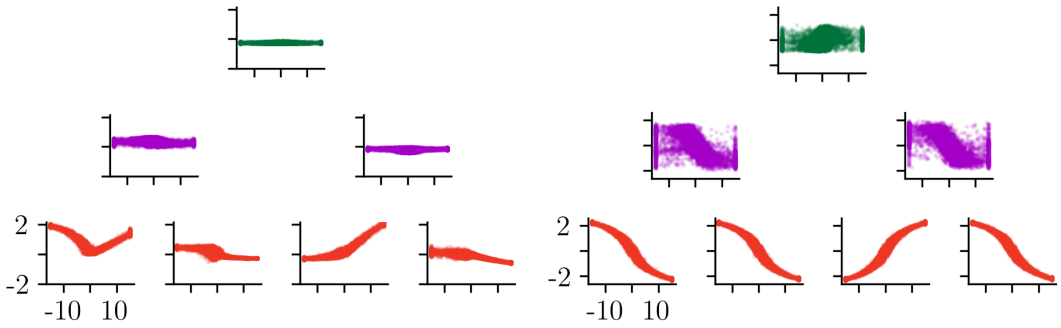



Figure S13: Scatter plot of the reciprocal of true latents (clipped at $[-15, 15]$ against the posterior means inferred by RAMP before (left) and after (right) learning, in the nonlinear tree dataset.

### C.3. Pendulum

To explore the robustness of RAMP's ability to learn informative latents to hyperparameters, we examined the learned latent spaces from all runs in the hyperparameter sweep (detailed in Appendix B.4) in Fig. S14.

Because message-passing in RAMP is amortised through a set of neural-networks, the method is not restricted to using only Gaussian distributions for the latent variables. Fig. S15 displays the learned latent space on the pendulum dataset from a model in which the prior and recognition factors of each $z_k$ are a product of two Beta distributions. With this choice, we are able to impose a proper uniform prior (on the square $[0, 1] \times [0, 1]$) for the latents representations.

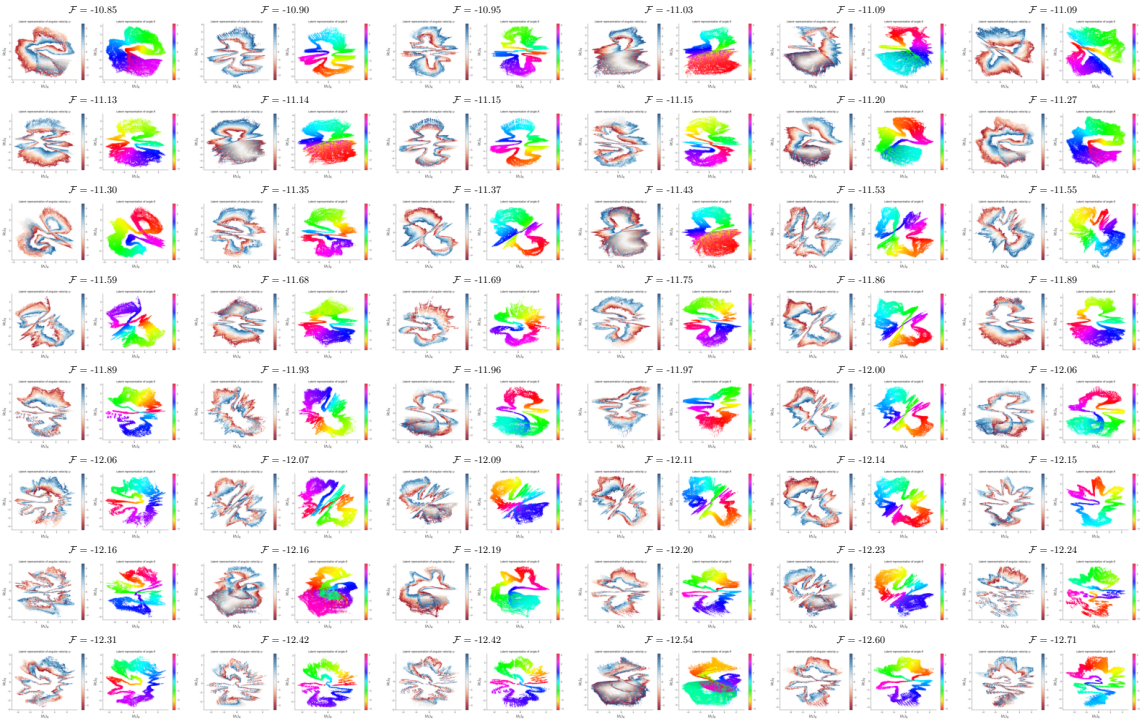


Figure S14: Latent space visualisations of all RAMP models from the pendulum hyperparameter sweep with Gaussian prior, ordered by increasing free energy. RAMP consistently and reliably learns a clean latent representation that distinguishes between different angles and angular velocities.

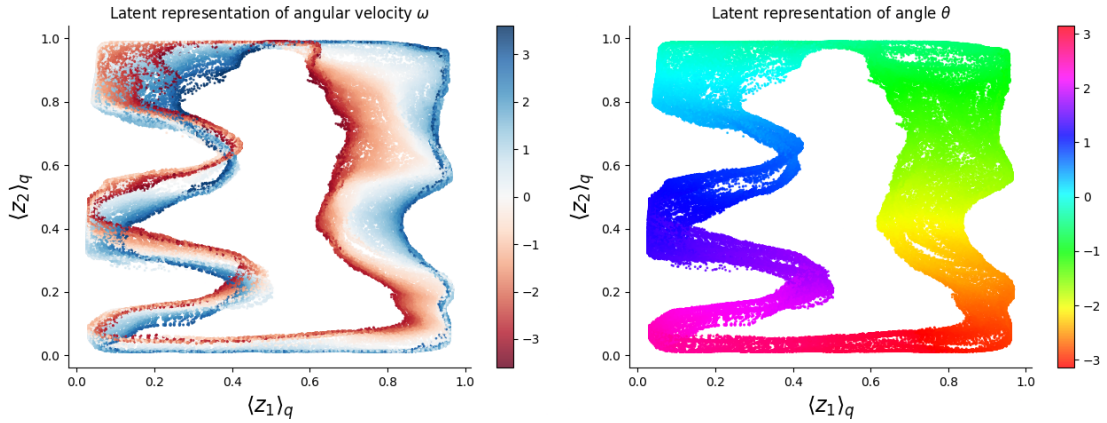



Figure S15: The learned latent space from a RAMP model with bivariate beta prior on the pendulum dataset.

### C.4. Pose Estimation

We performed linear regression from the 6D inferred posterior mean at each joint to the 2D cosine and sine of the angle of the edge (limb) connecting the joint to a neighbouring joint. This revealed a clear encoding of limb orientation in the inferred latent variables (Figure S16). We also performed nonlinear kernel ridge regression with an RBF kernel using scikit-learn with default hyperparameters. The resulting $R^2$ values are shown in Table S2 and show a strong correlation between the inferred latent variables and limb orientation.

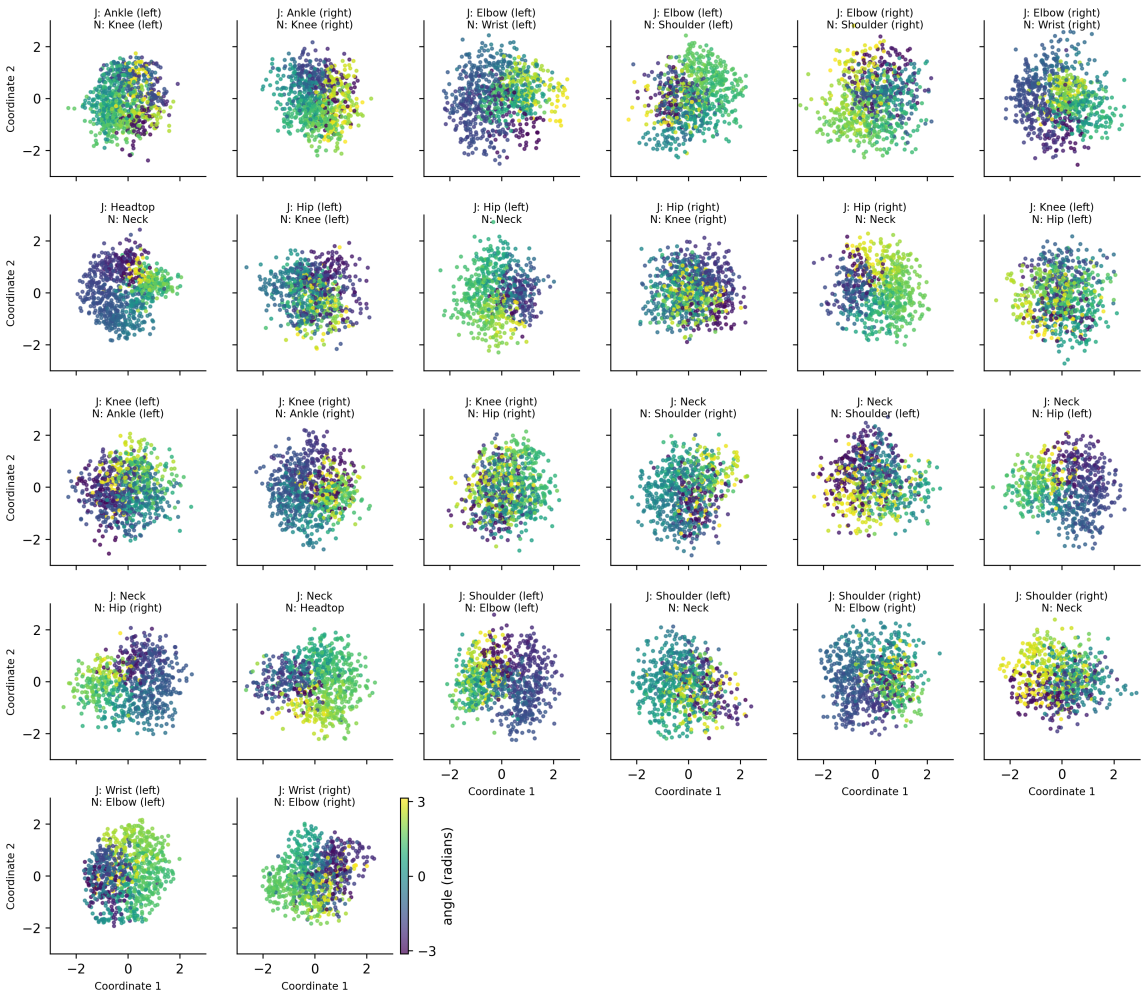



Figure S16: The 6D posterior means inferred at joint J are shown projected onto the two (orthonormalised) coefficient vectors obtained by performing linear regression from those means to the cosine and sine of the angle of the edge connecting joint J to neighbouring joint N. Each dot corresponds to a single image and is coloured based on the true angle of the edge connecting joint J to joint N.

Table S2: $R^2$ values from kernel ridge regression predicting the orientation of edges between each joint and each of its neighbours, based on the inferred posterior means at the joint.

| Joint | Neighbour | $R^2$ |
|---|---|---|
| Ankle (left) | Knee (left) | 0.83 |
| Ankle (right) | Knee (right) | 0.84 |
| Elbow (left) | Wrist (left) | 0.67 |
| Elbow (left) | Shoulder (left) | 0.56 |
| Elbow (right) | Shoulder (right) | 0.53 |
| Elbow (right) | Wrist (right) | 0.65 |
| Head (top) | Neck | 0.77 |
| Hip (left) | Knee (left) | 0.46 |
| Hip (left) | Neck | 0.68 |
| Hip (right) | Knee (right) | 0.50 |
| Hip (right) | Neck | 0.71 |
| Knee (left) | Hip (left) | 0.44 |
| Knee (left) | Ankle (left) | 0.73 |
| Knee (right) | Ankle (right) | 0.79 |
| Knee (right) | Hip (right) | 0.44 |
| Neck | Shoulder (right) | 0.40 |
| Neck | Shoulder (left) | 0.41 |
| Neck | Hip (left) | 0.79 |
| Neck | Hip (right) | 0.81 |
| Neck | Head (top) | 0.62 |
| Shoulder (left) | Elbow (left) | 0.65 |
| Shoulder (left) | Neck | 0.44 |
| Shoulder (right) | Elbow (right) | 0.56 |
| Shoulder (right) | Neck | 0.40 |
| Wrist (left) | Elbow (left) | 0.77 |
| Wrist (right) | Elbow (right) | 0.74 |