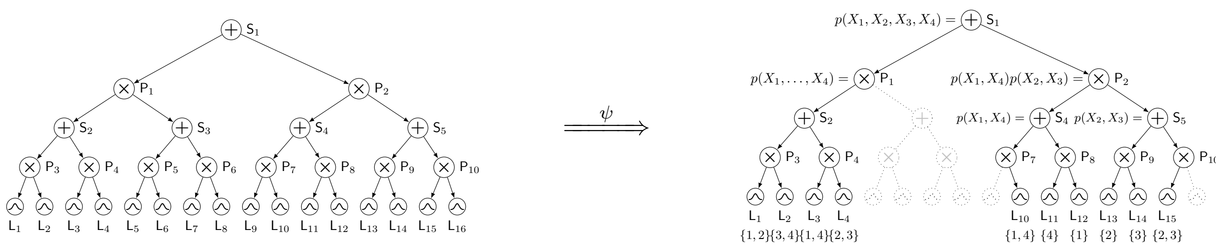

Figure 1: An example of a computational graph $\mathcal{G}$ (left) and a sum-product-network (SPN) structure (right), defined by the scope function ψ, discovered using posterior inference on $\mathbf{y}$. The resulting SPN might contain only a subset of the nodes in $\mathcal{G}$ as some sub-trees might be allocated with an empty scope during inference (dotted) – evaluating to constant 1. The graph $\mathcal{G}$ only encodes the topological layout of nodes, while the "effective" SPN structure is encoded via ψ. Example will be included in the supplementary.

General remarks **We want to thank all reviewers for their constructive feedback and for reviewing our work.** Fig. 1 depicts our proposed decomposition into computational graph and scope function. See caption for details.

Reviewer 1 **Running times for Gibbs sampling:** We will report detailed running times in the revised paper. For now, we report total running times on the cross-validated computational graphs, for a diverse selection of datasets. *Audio* ($N = 17000$, $D = 100$): 13h, 17m 49s; *EachMovie* ($N = 5526$, $D = 500$): 106h 54m 04s; *BBC* ($N = 1895$, $D = 1085$): 12h 49m 45s. These times were measured for an i7-6900k CPU @ 3.2 GHz.

Reviewer 2 **Related work:** We will restructure Sections 1 & 2 to provide a related work section, and incorporate related tractable probabilistic models. In particular, we will add (i) (extremely randomized) cutset networks [Rahman et al. 2014, Di Mauro et al. 2017], (ii) probabilistic sentential decision diagrams [Kisa et al. 2014, Liang et al. 2017] and (iii) mixtures of trees [Meila 2000].

Region-graph and Fig. 1: Section 3.2 contains a brief description of how to construct computational graphs from region graphs, which is admittedly quite terse. We will augment this description with a detailed description in the supplementary. Reference to Fig. 1 was accidentally removed in one of our paper iterations. This will be fixed in the revised paper.

SOTA results: We will add results of other tractable probabilistic models into the results table. In particular, we will list the latest results reported for: (i) cutset networks, (ii) probabilistic sentential decision diagrams and (iii) sum-product networks. Also, we will add a column listing the best results (considering all published results on tractable probabilistic models) for each dataset and drop the arrows in the tables.

Missing values: We selected k-NN imputation because it arguably provides a stronger baseline than simple mean imputation (while being computationally more demanding). Pairing structure learning with EM + MPE would be a possible avenue. However, using EM as an inner loop within a structure search would be computationally quite demanding. Using (approximate) MPE inference within a structure search is heuristic. Our Bayesian SPN framework is, as far as we know, the first method that allows coherent structure learning under missing data.

Learning only scope functions: We indeed focus on learning the scope function, as it is clearly the more challenging part – the computational graph has only the requirement to be acyclic, which could be addressed with tools from neural architecture search, AutoML, or, in future work, with Bayesian inference over graphs using more flexible priors.

Reviewer 3 **Computational graph identification:** We indeed decompose the sum-product network (SPN) structure learning problem into two parts, namely (i) determining a computational graph and (ii) learning the scope function. In our paper, we emphasise the latter aspect as it is far more challenging, due to SPNs' structural constraints – completeness and decomposability. Determining the computational graph is far simpler, and can be tackled with cross-validation (as in this paper), or as suggested by the reviewer using AutoML techniques or neural structural search (NAS). The reviewer is right that these directions are natural, but we leave them to future work.

Fixing computational structure: We do not have an iterative procedure to switch to a better computational graph, we only perform cross-validation over 24 different computational graphs. Within the validation loop, the computational graph $\mathcal{G}$ remains fixed. Fixing the computational graph structure is only unfair towards our approach.

Existing structure learners: Embedding our approach into existing structure learners is non-trivial, as all existing methods learn the computational graph and the scope function in an entangled way. As our paper focuses on introducing a new way of thinking about structure learning and stimulating research on Bayesian formulations, we leave those directions to future work.