Appendix to Faster Attend-Infer-Repeat with Tractable Probabilistic Models

Karl Stelzner Robert Peharz Kristian Kersting

1. Details on the Prior $p(z)$

Here, we specify the exact parameters used for $p(z)$. For $p(N)$, we assume a truncated geometric prior with a fixed success probability of 0.7. For the x and y coordinates of z^i_{where}, specifying the top left corner of the bounding box of object i, we assume a uniform distribution over the range $[0, 0.9B]$, where B is the size of the quadratic canvas. This ensures that objects cannot exceed the bounds of the canvas to an excessive degree. The scaling factor for the object's x dimension s_x, indicating the width of the bounding box relative to the object SPN's patch size $A = 28$, is similarly drawn uniformly from the interval $[0.3, 0.9]$. To ensure that object dimensions are not overly skewed, s_y is drawn from $[0.75s_x, 1.25s_x]$. Excessive overlap between objects is discouraged via an unnormalized penalty term on $p(z^i_{\text{where}})$, modelled as a Gamma distribution with $\alpha = 1, \beta = 120$ over each object's *occlusion ratio*, the ratio of its pixels which is occluded and will thus be marginalized.

2. Details on the SPN Structures

RAT-SPNs are based on the notion of a *region graph* (Dennis & Ventura, 2012; Peharz et al., 2018), a bipartite directed acyclic graph the nodes of which are either *regions* or *partitions*. A region node represents a certain sub-scope of the modeled random variables x, and partition nodes represent a decomposition of a parent region into two sub-regions. To construct an SPN from a region graph, each of the leaf regions is equipped with I distributions defined over the region's scope. In this paper, we simply assume Gaussians with isotropic covariance, i.e. products of single-dimensional Gaussians. Furthermore, each of the non-leaf regions is equipped with K sum nodes. Finally, each partition is equipped with the outer product of the distributions contained in the partition's two child regions, and these products are then passed on as inputs to the parent region. It can be shown that this process produces a complete and decomposable SPN (Peharz et al., 2018).

RAT-SPNs use a randomly generated region graph instead of learning the structure from data. This way, SPNs can be scaled to sizes similar to neural networks, and can fit high-dimensional densities, despite using a random structure. To construct a random region graph, the overall scope x (the root region) is repeatedly divided into two random sub-scopes of equal size, until a certain depth D, which is a hyperparameter of the model. This generates a binary region graph with 2^D leaf regions. The random splitting process is repeated R times, generating R parallel binary region graphs, all with the same root region. The random region graph generated in this way is then converted into an SPN.

Table 1. Hyperparameters for background and object networks.

	Object SPN	**BG SPN**
Split Depth D	2	1
Split Repetitions R	6	3
Sums per Region K	20	6
Min. variance σ^{min}	0.12	0.002
Max. variance σ^{max}	0.35	0.12

The structural parameters used for both the object and background network are given by Table 1. At their roots, both networks end in a single sum node. Also given are the bounds on the variance parameters of the SPNs' Gaussian leaf nodes. We deviate slightly from those values for the experiments on the sprite dataset and the noisy MNIST dataset: in the former case, we use $\sigma^{\text{max}}_{\text{bg}} = 0.16$, in the latter, we set $\sigma^{\text{max}}_{\text{bg}} = 0.06$ and $\sigma^{\text{min}}_{\text{obj}} = 0.25$ to ensure that the two networks' domains of expertise are sufficiently separated. For the ablation experiments, we employ a fixed background model instead of using the background SPN. Specifically, similarly to AIR, we express the expectation of the background being black by assuming a pixelwise normal distribution $\mathcal{N}(0, 0.35)$ over the background pixels.

3. Details on Datasets

The Multi-MNIST dataset was generated using the *observations* library[1] using a canvas size of 50×50. The sprite dataset was generated in a similar manner, using code available in our repository. In order to generate the noisy version of the dataset, each pixel x in Multi-MNIST was transformed in the following way:

$$x' = 0.8x + 0.1 + \epsilon,$$
$$\epsilon \sim \mathcal{N}(0.0, 0.2).$$

[1] https://github.com/edwardlib/observations

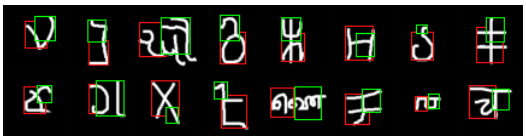

Figure 1. Segmentations obtained by SuPAIR when applied to the Omniglot dataset. The model learns to detect the outlines of the characters.

The result is then clipped to the interval $[0, 1]$.

The grids for the structured background were generated as follows:

$$x_0, y_0 \sim \text{Uniform}(0, 4)$$

$$bg_{ij} = \begin{cases} 0.4 & \text{if } i \bmod 5 = x_0 \text{ or } j \bmod 5 = y_0 \\ 0 & \text{else} \end{cases}$$

The Multi-MNIST scenes x were then overlaid to obtain the final dataset $x' = \max(x, bg)$.

4. Results on Omniglot

In addition to our other experiments, we also applied SuPAIR to the Omniglot handwritten character dataset introduced by Lake et al. (2015). While there is no clear ground truth as to how objects should be segmented in this dataset, it is still interesting to observe the model's behavior (Fig. 1).

References

Dennis, A. and Ventura, D. Learning the architecture of sum-product networks using clustering on variables. In *Proceedings of NIPS*, pp. 2042–2050, 2012.

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

Peharz, R., Vergari, A., Stelzner, K., Molina, A., Trapp, M., Kersting, K., and Ghahramani, Z. Probabilistic deep learning using random sum-product networks. *CoRR*, abs/1806.01910, 2018. URL http://arxiv.org/abs/1806.01910.