Sum-Product-Attention Networks: Leveraging Self-Attention in Energy-Based Probabilistic Circuits

Zhongjie Yu[1] **Devendra Singh Dhami**[1,2] **Kristian Kersting**[1,2,3]

[1]Department of Computer Science, TU Darmstadt,
[2]Hessian Center for AI (hessian.AI),
[3]Centre for Cognitive Science, TU Darmstadt

Abstract

Energy-based models (EBMs) have been hugely successful both as generative models and likelihood estimators. However, the standard way of sampling for EBMs is inefficient and highly dependent on the initialization procedure. We introduce Sum-Product-Attention Networks (SPAN), a novel energy-based generative model that integrates probabilistic circuits with the self-attention mechanism of Transformers. SPAN uses self-attention to select the most relevant parts of Probabilistic circuits (PCs), here sum-product networks (SPNs), to improve the modeling capability of EBMs. We show that while modeling, SPAN focuses on a specific set of independent assumptions in every product layer of the SPN. Our empirical evaluations show that SPAN outperforms energy-based and classical generative models, as well as state-of-the-art probabilistic circuit models in out-of-distribution detection. Further evaluations show that SPAN also generates better quality images when compared to EBMs and PCs.

1 INTRODUCTION

Energy-based models (EBMs) have recently been very successful in various tasks ranging from image generation [Du and Mordatch, 2019, Gao et al., 2021, Song et al., 2021a], likelihood estimation [LeCun et al., 2006, Du and Mordatch, 2019] and out-of-distribution detection [Zhai et al., 2016]. Although quite successful in sampling from probability distributions, EBMs fall short when faced with the task of reasoning about these distributions. Deep probabilistic models take advantage of the deep learning model efficiency and also abstract the underlying model representation while enabling reasoning over uncertainty in the domain. This has resulted in several probabilistic models such as arithmetic circuits [Darwiche, 2003], sum-product networks [Poon and Domingos, 2011] and more recently, probabilistic generating circuits [Zhang et al., 2021] to name a few. These probabilistic models learn the underlying joint distribution using a network polynomial over evidence variables and network parameters.

Although probabilistic models are touted to be scalable, there are limitations when they are used for learning with real-world data. To alleviate these limitations, Einsum networks (EiNet) [Peharz et al., 2020], which are essentially a form of sum-product networks (SPN), have been proposed. They combine several arithmetic operations in a single einsum operation and thus lead to learning speedups. Typical PCs, like EiNet, model the mixture of probability distributions at the sum node, shown in Figure 1 (Left). In the realm of deep learning, there has recently been work on Transformers which are models that leverage self-attention for capturing long-term dependencies for sequence-to-sequence modeling of text [Vaswani et al., 2017].

In this work, we introduce Sum-Product-Attention Networks (SPAN) that incorporate the concept of self-attention in SPNs to select the most relevant sub-SPN while modeling a given data sample and also take advantage of the tractable power of probabilistic circuits. We assume that activating the most relevant child(ren) at the sum node of SPNs should be sufficient to represent the mixture, and show that while modeling, SPAN focuses on a specific set of independent assumptions via self-attention, in every product layer of the sum-product network. We specifically make use of the Transformer encoder in every product layer of SPAN. During the training phase of SPAN, the inputs are fed to both the Transformer encoders and the SPN. The encoder outputs the attention weights for the product nodes in the corresponding product layer. The outputs at the root of the SPN given the inputs are then dominated by the product nodes which have relatively higher attention weights, see Figure 1 (Right).

We make the following key contributions: (1) We introduce a new energy-based model class that leverages the power

Accepted for the 5th Workshop on Tractable Probabilistic Modeling at UAI (TPM 2022).

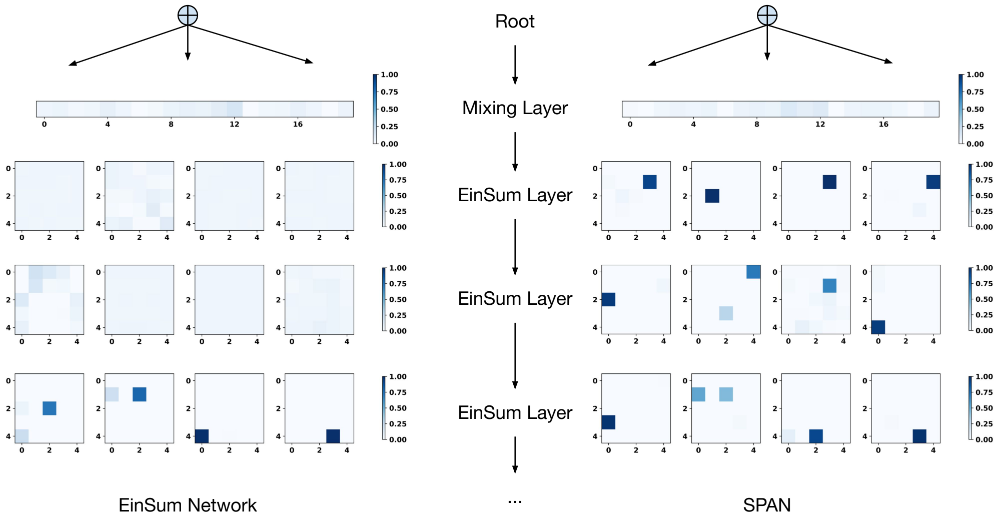


Figure 1: SPAN pays attention to only one or a few product nodes at each sum node, while EiNet treats all the product nodes from one sum node almost equally in the upper layers. The 5×5 matrices stand for the last 2 dimensions of the updated weights at a sum node. The 1×20 vector visualizes the weights of the 20 replicas of the EiNet, where no self-attention is employed. Weights from one sum node are all normalized.

of self-attention in order to identify and select the most relevant sub-SPN structure while learning. (2) Our model class is agnostic to the type of Transformer encoder as well as the SPNs being used and thus can model various data types. (3) Our model offers a principled way to connect self-attention to sum-product networks, without breaking tractability in likelihood estimation. (4) We show that SPAN acts as a generative model outperforming state-of-the-art EBM and PCs.

2 RELATED WORK

Recently, energy-based models became popular because of their generality and simplicity in likelihood modeling [Du and Mordatch, 2019, LeCun et al., 2006]. EBMs, which are also probabilistic models, are widely used in *e.g.* image generation [Du and Mordatch, 2019, Gao et al., 2021] and out-of-distribution (OOD) detection [Zhai et al., 2016]. More recently, based on EBM, score-based models [Song et al., 2021a,b] were proposed, which diffuse the data distribution towards a noise distribution using a stochastic differential equation (SDE), and the time reversal of this SDE is learned for sample generation.

Transformers were initially proposed for natural language processing (NLP) [Vaswani et al., 2017], but have been widely adopted in various fields, such as computer vision [Dosovitskiy et al., 2021], speech processing [Dong et al., 2018], databases [Thorne et al., 2021] and genomics [Ji et al., 2021].

Probabilistic circuits are popular probabilistic models which allow a wide range of exact and efficient inference routines. PCs were firstly developed from arithmetic circuits [Darwiche, 2003], and later on into SPNs [Poon and Domingos, 2011] and some other members such as cutset networks (CNets) [Rahman et al., 2014] and probabilistic sentential decision diagrams (PSDD) [Kisa et al., 2014]. SPNs are a family of tractable deep density estimators first presented in Poon and Domingos [2011]. SPNs can represent high-treewidth models [Zhao et al., 2015] and facilitate exact inference for a range of queries in time polynomial in the network size [Bekker et al., 2015].

3 SPAN

The Sum-Product-Attention Network leverages the attention mechanism for highlighting the most relevant sub-SPNs thus improving its modeling ability. SPAN consists of an SPN and Transformer encoders for each SPN product layer. We can now define the Sum-Product Attention Networks formally followed by introducing the architecture and the training and inference mechanisms.

Definition 1. ***Sum-Product-Attention Networks.*** *A sum-product-attention network $\mathcal{C}^A = (\mathcal{G}, \theta_\mathcal{G}, \mathcal{T}, \theta_\mathcal{T})$ over RVs $\mathbf{X} = \{X_i\}_1^N$ is an energy-based SPN $\mathcal{C} = (\mathcal{G}, \theta_\mathcal{G})$, connected to a set of Transformer encoders $\mathcal{T}$, which are parameterized by a set of graph parameters $\theta_\mathcal{G}$ and Transformer encoder parameters $\theta_\mathcal{T}$. Given datum x, the normalized weights $w_{\mathsf{S},\mathsf{P}} \in \theta_\mathcal{G}$ of a sum node S in $\mathcal{C}$ are re-parameterized by the attention weights $w^A_{\mathsf{S},\mathsf{P}}(x)$ which are outputs of $\mathcal{T}$. The output of $\mathcal{C}^A$ is the output computed at the root of $\mathcal{C}$ and represents the* ***negative energy****: $-E_\theta(\mathbf{X})$,* *where $\theta = \{\theta_{\mathcal{G}}, \theta_{\mathcal{T}}\}$. The negative energy can also be seen as the* ***likelihood score*** *from SPAN.*

Given datum x, SPAN uses Transformer *encoder* to produce the **attention weight** $w^A_{\mathsf{S},\mathsf{P}}(x)$ for the product node P where $\mathsf{P} \in \mathbf{ch}(\mathsf{S})$. The Transformer encoder consists of an embedding layer, followed by several stacks of multi-head attention sub-layers and feed-forward sub-layers. Similar to Transformer decoder, a feed-forward layer and a softmax layer are employed to connect the Transformer encoder output to the product nodes P. To this end, each product node in one product layer will be provided with an attention weight from the Transformer corresponding to this product layer.

3.1 TRANSFORMER EMBEDDING

In order to model various types of random variables, different embeddings can be used in the Transformer encoder.

Bernoulli distribution To model Bernoulli random variables rather than tokens as in sequence transduction models, we propose the following embedding strategy for the Transformer encoder. For a binary random variable X_i, value True is embedded as $\mathsf{01}$ and value False as $\mathsf{10}$:

$$\text{Embedding}(X_i) = \begin{cases} 10 & \text{if } X_i = \text{True} \\ 01 & \text{if } X_i = \text{False} \end{cases}. \qquad (1)$$

Therefore, the dimension of embedding d_{model} for a binary random variable is 2.

Modeling images On the other hand, when modeling images, Vision Transformer (ViT) [Dosovitskiy et al., 2021] can be employed and thus the "Patch + Position" embedding can be used to embed the image. That is, the image $X \in \mathbb{R}^{H \times W \times C}$ is reshaped into a sequence of flattened patches $X_p \in \mathbb{R}^{p \times p \times C}$, where (H, W) is the resolution of the image, C is the number of (color) channels, and p is the patch size. Here, each patch works as token and the flattened patch naturally becomes the embedding of the token.

3.2 ATTENTION IN SPAN

Existing PCs use a sum node S to model the mixture of its children N, *i.e.*, $\mathsf{S} = \sum_{\mathsf{N} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}} \mathsf{N}$, where $\sum_{\mathsf{N} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}} = 1$. In SPAN, we follow the same mechanism but pay more attention to the product nodes P that fit more the distribution of the subset of corresponding data instances. To achieve this, each product node P is assigned with an extra attention weight $w^A_{\mathsf{S},\mathsf{P}}(x)$ provided by the Transformer encoder given x, working as a gate. To be more specific, a sum node S in SPAN computes the convex combination of its re-weighted children:

$$\mathsf{S}(x) = \sum_{\mathsf{P} \in \mathbf{ch}(\mathsf{S})} \frac{w_{\mathsf{S},\mathsf{P}} w^A_{\mathsf{S},\mathsf{P}}(x) \mathsf{P}}{\sum_{\mathsf{P} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{P}} w^A_{\mathsf{S},\mathsf{P}}(x)}, \qquad (2)$$

where $\sum_{\mathsf{P} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{P}} = 1$ and $\sum_{\mathsf{P} \in \mathbf{ch}(\mathsf{S})} w^A_{\mathsf{S},\mathsf{P}}(x) = 1$. Here, although both weights $w_{\mathsf{S},\mathsf{P}}$ and $w^A_{\mathsf{S},\mathsf{P}}(x)$ are normalized, the weights at a sum node S depend on the input x. Hence, the output at the root of SPAN is no more normalized and is defined as the negative energy $-E_\theta(\mathbf{X})$. Note that this operation is different from the neural CSPN [Shao et al., 2020], where the weights of a sum node are directly determined by the output of the feed-forward neural network.

SPAN can work with various types of sum-product networks. We illustrate here SPAN with EiNet, a novel implementation design for PCs [Peharz et al., 2020], as an example. In EiNet, assume a sum node S has one child that is a product node P, and P has two children N' and N''. Therefore, the output at S is given in *Einstein notation* as:

$$\mathsf{S}_k = \mathbf{W}_{kij} \mathsf{N}'_i \mathsf{N}''_j, \qquad (3)$$

where $\mathbf{W}$ is a $K \times K \times K$ tensor, and N'_i, N''_j are outputs of the children. $\mathbf{W}$ is normalized over its last two dimensions, *i.e.*, $\mathbf{W}_{kij} \geq 0$, $\sum_{i,j} \mathbf{W}_{kij} = 1$. In order to provide the attention weights for the product node (here in the eimsum layer), the attention weight tensor $\mathbf{W}^A(x)$ from the Transformer encoder should have size $b \times K \times K \times K$, given input data with batch size b. The softmax is then applied over the last two dimensions, to ensure $\sum_{i,j} \mathbf{W}^A_{bkij}(x) = 1$. Following Equation (2), the updated weights $\mathbf{W}^{\mathsf{S}}(x)$ are:

$$\mathbf{W}^{\mathsf{S}}_{bkij}(x) = \frac{\mathbf{W}_{kij} \times \mathbf{W}^A_{bkij}(x)}{\sum_{i,j} \mathbf{W}_{kij} \times \mathbf{W}^A_{bkij}(x)}. \qquad (4)$$

3.3 TRAINING AND INFERENCE

SPAN is trained by minimizing its energy $E_\theta(\mathbf{X})$, which can also be seen as maximizing the likelihood score from the root. The training procedure of SPAN mainly consists of 3 phases. In the first phase which we call SPN warm-up, the trainable parameters of the SPN are updated with ep_1 epochs while the Transformer parameters remain fixed. In the second phase, the SPN parameters and the Transformer parameters are updated iteratively, in a coordinate descent fashion for ep_2 epochs. In the last phase, the Transformer trainable parameters are again fixed and the SPN parameters are fine-tuned by maximizing the root output for ep_3 epochs.

Inference of SPAN is similar to SPNs for taking advantage of SPNs' tractable inference property. More details are in the appendix.

4 EXPERIMENTAL EVALUATION

In order to investigate the benefits of SPAN compared to EBMs and other probabilistic models, we aim to answer the following research questions: **(Q1)** Can SPAN capture out-of-distribution data better than energy-based and classical generative models, as well as state-of-the-art probabilistic

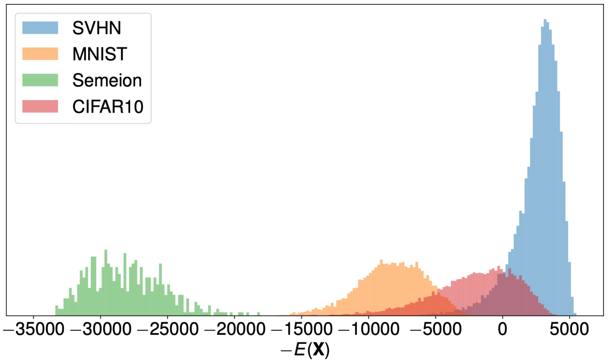


Figure 2: Sample-wise SPAN log-likelihood scores of *SVHN* test set are much higher than the test sets of MNIST, Semeion and CIFAR10. Bins of Semeion data set are re-scaled for better visualization.

Table 1: SPAN outperforms EBM and deep generative models measured with AUC scores of out-of-distribution classification on different data sets, and is on a par with EiNet. All models trained on *SVHN* training set.

Data set	SPAN	EBM	EiNet	MAF	VAE
CIFAR10	0.9270	0.8425	0.9273	0.9517	**0.9889**
MNIST	**0.9992**	0.5416	0.9989	0.9921	0.9296
Semeion	**1.0000**	0.8734	**1.0000**	0.9965	0.9994

Table 2: SPAN reconstructs images better than baselines by providing the best reconstruction error.

	Filling top half		Filling left half	
	mean	std.	mean	std.
SPAN	$\mathbf{8.76 \times 10^{-3}}$	6.84×10^{-3}	$\mathbf{6.73 \times 10^{-3}}$	5.29×10^{-3}
EiNet	9.28×10^{-3}	7.90×10^{-3}	7.08×10^{-3}	6.13×10^{-3}
EBM	1.35×10^{-2}	7.29×10^{-3}	1.40×10^{-2}	6.90×10^{-3}

circuit models? **(Q2)** Does SPAN work well as a generative model for images?

To answer these questions, we evaluate the performance of SPAN on the image data set *SVHN* [Netzer et al., 2011] which is a real-world image data set with house numbers in Google Street View images, incorporates over 600,000 digit images and comes from a significantly harder, unsolved, real-world problem [Netzer et al., 2011]. *SVHN* contains 32 × 32 RGB images of digits.

(Q1) Better OOD detector. We train SPAN on the full *SVHN* data set to evaluate its modeling ability for images. SPAN employs an EiNet with number of entries $K = 5$, split-depth $D = 4$, number of replica $R = 100$, with a random binary tree structure. SPAN also uses a Vision Transformer to encode the image inputs, which has patch size 8, depth 2, and 16 heads and the embedding has dimension 1024 with feed-forward layers of dimension 512. SPAN training follows the 3-phased training with $ep_1 = 5$ for warm-up training of SPN, $ep_2 = 10$ for coordinate descent and $ep_3 = 15$ for fine-tuning the SPN weights with fixed Transformer weights. As for comparisons, we employed 1) the unconditional EBM from Du and Mordatch [2019] and used its default hyper-parameters, 2) the vanilla EiNet with the same structure as the SPAN components, trained with EM for 30 epochs, 3) MAF [Papamakarios et al., 2017] and 4) VAE [Choi et al., 2021] with their default hyper-parameters. To estimate the running time, we ran both SPAN and EiNet for 5 times, and the average running time for SPAN is 11341.9s and for EiNet is 5547.5s.

SPAN can detect the OOD test samples better than EBM, which is a property of PCs. As shown in Figure 2, the negative energy of MNIST [LeCun et al., 1998] test set has almost no overlap with the *SVHN*, mainly because MNIST has grayscale images. CIFAR10 [Krizhevsky et al., 2009] has color images thus distributing closer to *SVHN*, while the overlap happens at the lower negative energy values of *SVHN*. The Semeion [Buscema, 1998] data set contains binary images and thus has extremely low negative energy. Hence, SPAN trained on *SVHN* can successfully distinguish the outliers from MNIST and Semeion data sets, and also provide much lower negative energy given images from CIFAR10. We employed the area under the curve (AUC) to quantitatively measure the OOD classification quality [Hendrycks and Gimpel, 2017]. In Table 1, SPAN performs significantly better OOD detection than unconditional EBM, and overall better than the other baselines. Therefore, we can answer **Q1** affirmatively: SPAN captures out-of-distribution data better than energy-based (EBM) and classical generative models (VAE), as well as state-of-the-art probabilistic circuit models (EiNet).

(Q2) Generative model for images. Table 2 presents the mean squared error (MSE) of the reconstruction by the 3 methods on *SVHN* test set. SPAN produces the best reconstructions with the smallest reconstruction error, both in filling the top half and the left half of the image. The reconstructed images and more details are in the appendix. Thus, **Q2** can be answered affirmatively: SPAN can model the image distribution, providing samples as good as baselines, and producing better-visualized reconstructions.

5 CONCLUSION

We presented SPAN, a new model that incorporates attention in the SPN architecture. This results in selection of the most relevant sub-SPN structures during learning while taking advantage of the tractable power of PCs. We show that SPAN is a better generative model in OOD detection and image generation. Future works include reducing the number of embeddings while modeling Bernoulli random variables. Extending our model to handle missing data and marginalization with the Transformer is a natural next step.

References

Rupam Acharyya. *Approximating Partition Functions of Spin Systems and Its Applications*. PhD thesis, University of Rochester, 2020.

Jessa Bekker, Jesse Davis, Arthur Choi, Adnan Darwiche, and Guy Van den Broeck. Tractable learning for complex probability queries. In *NeurIPS*, pages 2242–2250, 2015.

Massimo Buscema. Metanet*: The theory of independent judges. *Substance use & misuse*, 1998.

Jaemoo Choi, Changyeon Yoon, Jeongwoo Bae, and Myungjoo Kang. Robust out-of-distribution detection on deep probabilistic generative models. *arXiv preprint arXiv:2106.07903*, 2021.

Adnan Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM (JACM)*, 50(3): 280–305, 2003.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *ICASSP*, pages 5884–5888. IEEE, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *NeurIPS*, volume 32, pages 3608–3618, 2019.

Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. In *ICLR*, 2021.

Shahrzad Haddadan, Yue Zhuang, Cyrus Cousins, and Eli Upfal. Fast doubly-adaptive mcmc to estimate the gibbs partition function with weak mixing time bounds. *NeurIPS*, 34, 2021.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *KR*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning, 2011. In NIPS Workshop.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *NeurIPS*, 30, 2017.

Robert Peharz, Steven Lang, Antonio Vergari, Karl Stelzner, Alejandro Molina, Martin Trapp, Guy Van den Broeck, Kristian Kersting, and Zoubin Ghahramani. Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *ICML*, pages 7563–7574, 2020.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *UAI*, pages 337–346, 2011.

Tahrima Rahman, Prasanna Kothalkar, and Vibhav Gogate. Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of chow-liu trees. In *ECML-PKDD*, pages 630–645, 2014.

Xiaoting Shao, Alejandro Molina, Antonio Vergari, Karl Stelzner, Robert Peharz, Thomas Liebig, and Kristian Kersting. Conditional sum-product networks: Imposing structure on deep probabilistic architectures. In *PGM*, pages 401–412, 2020.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *NeurIPS*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021b.

James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. From natural language processing to neural databases. In *VLDB*, volume 14, pages 1033–1039, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *ICML*, pages 1100–1109. PMLR, 2016.

Honghua Zhang, Brendan Juba, and Guy Van den Broeck. Probabilistic generating circuits. *arXiv*, abs/2102.09768, 2021.

Han Zhao, Mazen Melibari, and Pascal Poupart. On the relationship between sum-product networks and bayesian networks. In *ICML*, pages 116–124, 2015.

A SPAN ARCHITECTURE

Figure 3 (Left) shows the overall SPAN architecture, where each product layer is equipped with one Transformer encoder. The structure of Transformer encoder in SPAN is depicted in Figure 3 (Right).

B SPAN IMAGE RECONSTRUCTION

To qualitatively evaluate SPAN as a generative model, we visualize the samples and reconstructions from SPAN in Figure 4. The reconstructions are from the approximated most probable explanation (approximated MPE) query. That is, $\arg\max$ is employed in the top-down pass at each sum node of the PC component. Similar to the experimental settings in **(Q1)**, SPAN, EBM and EiNet are trained on one cluster from 100 k-means clusters from *SVHN* data set. Figure 4 (b), (c) and (d) show samples from SPAN, EBM and EiNet, respectively. SPAN produces image samples as good as EiNet, while EBM samples are smoother and have twisted shapes of digits. There are no “stripy” artifacts as both SPAN and EiNet employed a binary tree structure, instead of the PD architecture [Poon and Domingos, 2011, Peharz et al., 2020]. On the other hand, the sampled images contain more noise due to the blurry images in the data set (see Figure 4 (a)).

In most cases, SPAN, EBM and EiNet all successfully reconstructed the digit given half of the image, as shown in Figure 4 (e), (f) and (g). Overall, the SPAN reconstructions show less noise than EiNet, *e.g.*, digit “7” in the 2^{nd} column, digit “9” in the 3^{rd} column, and digit “2” in 4^{th} and 5^{th} columns. Furthermore, SPAN better reconstructs digit “2” in the 7^{th} column, especially in the case of left-half-missing. The reconstruction from SPAN is left half of digit “2”, while the EiNet reconstruction is not recognisable. The reconstruction of this image is challenging as there is also another digit “3” in the image, which inputs additional noise to the models. The EBM reconstructions are initialized from missing pixels all being 0.5 (gray), in order not to include the randomness of the initial pixel values. EBM reconstructions are again smoother but can not reconstruct the digits as good as SPAN and EiNet.

C SPAN INFERENCE

Inference of SPAN is similar to SPNs for taking advantage of SPNs’ tractable inference property. When computing the likelihood score, the Transformer encoder is activated. In order to obtain a normalized probability from the likelihood score, it is in most cases efficient to calculate the partition function $Z(\theta)$. A large number of RVs makes the exact calculation of $Z(\theta)$ infeasible and therefore MCMC approaches can be applied to approximate $Z(\theta)$ [Haddadan et al., 2021, Acharyya, 2020]. On the other hand, when computing the bottom-up pass from SPAN with RVs marginalized, the Transformer encoder is deactivated, as the Transformer encoder is not trained with missing values. Moreover, for the top-down pass *e.g.* sampling, the Transformer encoder is also deactivated. That is, with marginalization and data generation, SPAN inference degenerates to SPN inference.

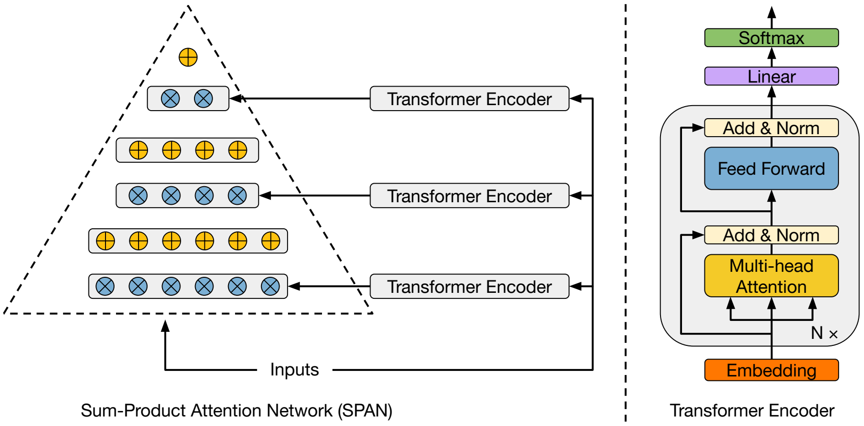
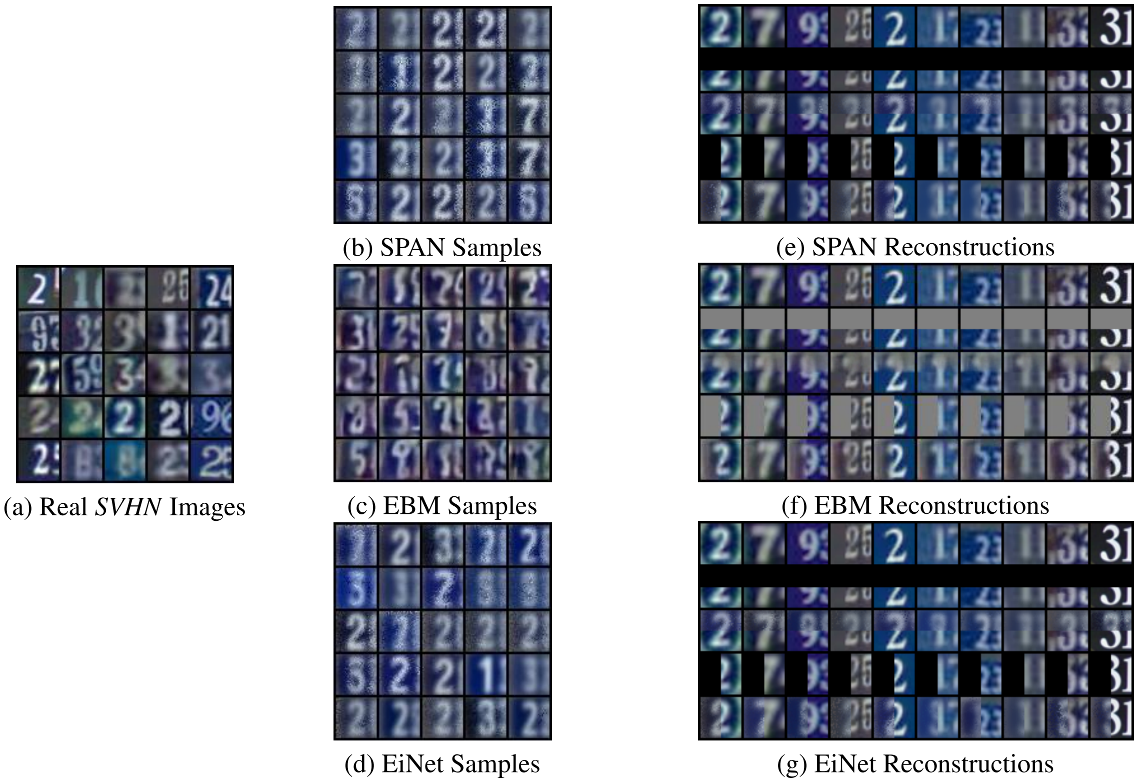


Figure 3: Illustration of Transformer encoder in SPAN. Inputs are fed to both the Transformer encoders and the SPN. The Transformer encoder outputs the attention weights for the product nodes in the corresponding product layer. The outputs at the root of the SPN given the inputs are then dominated by the product nodes which have relatively higher attention weights.


(b) SPAN Samples

(e) SPAN Reconstructions

(a) Real *SVHN* Images

(c) EBM Samples

(f) EBM Reconstructions

(d) EiNet Samples

(g) EiNet Reconstructions

Figure 4: SPAN generates samples as good as EiNet, while EBM samples capture better contrast and smoothness of images, but worse shapes of digits. SPAN also reconstructs better-visualized digits than both EiNet and EBM. For a fair comparison, EBM samples are generated from pixel values being 0.5, rather than uniformly random initialization.