Collapsed Inference for Bayesian Deep Learning

Zhe Zeng [1] **Guy Van den Broeck** [1]

Abstract

Bayesian deep learning performs well at providing prediction accuracy and calibrated uncertainty. Current research has been focused on scalability by imposing simplistic assumptions on posteriors and predictive distributions, which harms the prediction performances. While an accurate estimation of the posterior is critical to performance, doing so is computationally expensive and prohibitive in practice since it would require running a long Monte Carlo chain. In this paper, we explore a trade-off between reliable inference and algorithm scalability. The main idea is to use collapsed samples: while doing full Bayesian inference, we sample some of the stochastic weights and maintain tractable conditional distributions for the others, which are applicable to exact inference. This is possible by encoding the Bayesian ReLU neural networks into probabilistic Satisfiability Modulo Theories models and further leveraging a recently proposed tool that is able to perform exact inference for such models. We illustrate our proposed collapsed Bayesian deep learning algorithm on regression tasks. Empirical results show significant improvements over the existing Bayesian deep learning approaches.

1. Introduction

Bayesian inference with neural networks by Bayesian model averaging (BMA) (Fragoso et al., 2018) is particularly compelling in Bayesian deep learning community. However, to compute BMA is distinctly challenging. We will show that even with approximate posteriors in a simple form and a low-dimensional parameter space, doing BMA involves integration over highly non-convex distributions. In general, BMA requires to integrate complex and multi-modal posteriors over high dimensional parameter space, which has observed to have some unusual topological properties such as mode-connectivity.

How to accurately approximate the true BMA has been an attractive topic since it can potentially achieve significant performance gains (Izmailov et al., 2021). Existing Bayesian deep learning approaches mainly focus on computational convenience such as MCMC-based ones that could potentially give heavily biased estimation in practice for posterior expectations, and variational-inference-based approaches that typically use unimodal Gaussian approximations for posterior estimation. Their strong performance on benchmark problems does not imply that the algorithm accurately approximate the true BMA.

In this work, we are interested in using *collapsed samplers*, also known as *cutset* or *Rao-Blackwellised* samplers for BMA, which improve over classical particle-based methods by limiting sampling to a subset of the variables while pairing it with some closed-form representation of a conditional distribution over the rest. The accuracy of BMA computation is improved by performing exact marginalization on the conditional distributions while the efficiency is guaranteed by the sampling part of the inference algorithm.

The exact marginalization is possible thanks to recent advances in probabilistic inference tools under algebraic constraints. Inspired by recent work where ReLU neural networks are compiled into Satisfiability Modulo Theories (SMT) formulas and analyzed by SMT solvers, we observe that BNNs with ReLU activation functions can be encoded as probabilistic SMT models, over which BMA can be solved by leveraging the weighted model integration (WMI), an inference framework for doing marginalization over probabilistic SMT models (Belle et al., 2015; Morettin et al., 2017; Zeng et al., 2020b). Various WMI solvers have been built (Morettin et al., 2019; de Salvo Braz et al., 2016; Kolb et al., 2018; Zuidberg Dos Martires et al., 2019; Zeng and Van den Broeck, 2019; Zeng et al., 2020a;b). With the above insights, we make two main contributions: **1)** we propose to perform BMA using collapsed samples to increase the accuracy of prediction and uncertainty estimation while maintaining scalability; **2)** we propose an algorithm name CIBER for collapsed BMA based on an encoding of the BNN in probabilistic SMT models over which exact inference can be performed by the WMI solvers.

[1] Computer Science Department, University of California, Los Angeles, USA. Correspondence to: Zhe Zeng <zhezeng@cs.ucla.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

2. Proposed Approach

Notation. Uppercase letters (e.g., X) denote random variables and lowercase letters (e.g., x) denote their assignments. We use bold for sets of variables or a vector of variables (e.g., $\boldsymbol{X}$) and their joint assignments (e.g., $\boldsymbol{x}$). We use capital Greek letters (e.g., Δ) for logical formulas. Let $\boldsymbol{x} \models \Delta$ denote the satisfaction of a formula Δ by an assignment $\boldsymbol{x}$, with a corresponding indicator function $[\![\boldsymbol{x} \models \Delta]\!]$.

We start by formalizing the predictive distribution given by *Bayesian model averaging (BMA)*:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \int p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \, p(\boldsymbol{w} \mid \mathcal{D}) \, d\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{w}$ are the parameters of a neural network model denoted by $f_{\boldsymbol{w}}$, $\mathcal{D}$ denotes the observed data, and $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ is a succinct representation for $p(\boldsymbol{y} \mid f_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{x}, \boldsymbol{w})$. In this work, we are interested in regression tasks. With the above predictive distribution, an expected regression prediction is given by $\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$.

To obtain an accurate estimation of the predictive uncertainty $p(\boldsymbol{y} \mid \boldsymbol{x})$ is hard. First, computing the exact posterior $p(\boldsymbol{w} \mid \mathcal{D})$ is intractable in general and it requires approximation. Current Bayesian inference methods approximate the posterior either by *variational inference*, to learn some approximate distributions of simple forms for example Gaussian distributions that are easy to sample from, or by *SGD approximate inference*, to sample from SGD trajectories that is believed to be a good approximation for the true posterior. Further, even with approximate posteriors being Gaussian, there is no analytical solution for $p(\boldsymbol{y} \mid \boldsymbol{x})$ or the marginalized predictions $\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$, where most of existing work require Monte Carlo estimation for the marginalization.

Instead, we are interested in providing an accurate estimation of the predictive uncertainty $p(\boldsymbol{y} \mid \boldsymbol{x})$ by computing the marginalization in Equation 1 exactly under some approximate posteriors. While the exact computation might be costly especially when the parameter space is large, we propose a *collapsed BMA* scheme that allows a trade-off between accuracy and scalability.

Definition 2.1. *(Collapsed BMA) Let $(\boldsymbol{W}_s, \boldsymbol{W}_c)$ be a partition for network parameters $\boldsymbol{W}$. A collapsed sample from the posterior for parameters $\boldsymbol{W}$ takes the form of a tuple $(\boldsymbol{w}_s, p(\boldsymbol{W}_c \mid \boldsymbol{w}_s, \mathcal{D}))$, where $\boldsymbol{w}_s$ is an assignment for the sampled parameter $\boldsymbol{W}_s$ and $p(\boldsymbol{W}_c \mid \boldsymbol{w}_s, \mathcal{D})$ is a conditional posterior over the* collapsed set $\boldsymbol{W}_c$. *With M collapsed samples, the* collapsed BMA *gives the predictive distribution and marginalized prediction as below*

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{M} \sum_{\boldsymbol{w}_s} \int p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \, p(\boldsymbol{w}_c \mid \boldsymbol{w}_s, \mathcal{D}) \, d\boldsymbol{w}_c,$$

$$\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}] = \frac{1}{M} \sum_{\boldsymbol{w}_s} \int \boldsymbol{y} \, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \, p(\boldsymbol{w}_c \mid \boldsymbol{w}_s, \mathcal{D}) \, d\boldsymbol{w}_c \, d\boldsymbol{y}.$$

To develop an algorithm to compute collapsed BMA, we are faced with a few design choice questions: **(Q1)** how to sample $\boldsymbol{w}_s$ from the posterior? **(Q2)** what should be the closed-form representation for the collapsed set such that the integrals in collapsed BMA can be computed exactly? Next, we will provide our answer to the two questions that leads to our proposed solution CIBER.

The answer to **(Q1)** is more straightforward than **(Q2)**. To find an approximation to posteriors, we follow Maddox et al. (2019) where we sample from the SGD trajectory and use the information contained in the SGD trajectory to efficiently approximate the posterior distribution over the parameters of the neural network, leveraging the interpretation of SGD as approximate Bayesian inference (Mandt et al., 2017; Chen et al., 2020). Given a set of parameter samples $\mathcal{W}$, the sample set is defined as $\mathcal{W}_s = \{\boldsymbol{w}_s \mid \boldsymbol{w} \in \mathcal{W}\}$. For each assignment $\boldsymbol{w}_s$, an approximation $q(\boldsymbol{W}_c)$ to the conditional posterior $p(\boldsymbol{W}_c \mid \boldsymbol{w}_s, \mathcal{D})$ is necessary since the posteriors induced by SGD trajectories are implicit. Next, we focus on our solution to **(Q2)**.

2.1. Exact BMA via WMI

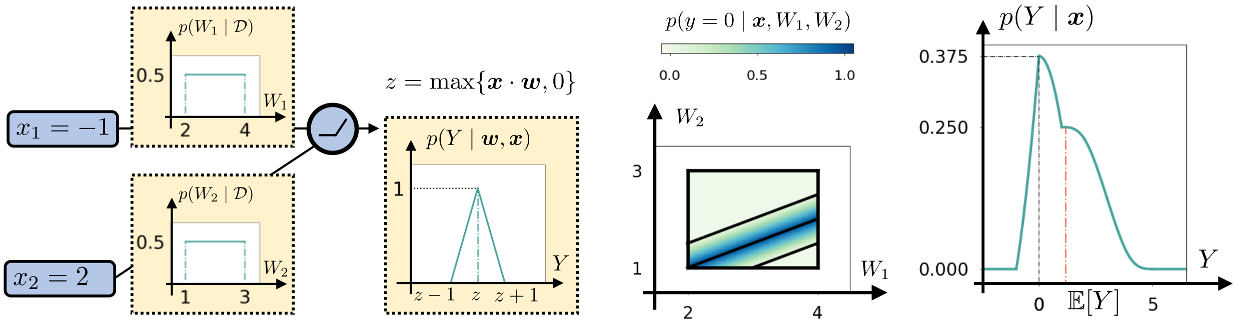
One key insight in our work is that the Bayesian neural network models can be encoded as a so-called *probabilistic Satisfiability Modulo Theories (SMT)* model, which is amenable to advanced exact inference tool called *weighted model integration (WMI)* solvers such that the integral in BMA can be solved exactly. We propose to use the probabilistic SMT model as the closed-form representation for the approximate conditional posterior on the collapsed set as our answer to **(Q2)**. We first introduce the probabilistic SMT models and WMI solvers and then show how BMA can be exactly solved by WMI solvers.

An SMT formula is an expression containing both propositional and *theory* atoms connected with the usual logical connectives (Barrett et al., 2010). The theory atoms encode algebraic constraints over $\mathbf{X}$, often restricted to the theory of *linear algebra over reals* (LRA), where atoms have form $(\mathbf{c}^T\mathbf{X} \leq b)$. The encoding of neural networks into SMT formulas has been explored to enable formal verification on behaviours of neural networks (Katz et al., 2017; Huang et al., 2017).

Example 2.2. *The* ReLU *activation function $Z = \mathsf{ReLU}(\boldsymbol{x}; \boldsymbol{W})$ with $\Rightarrow$ denoting logical implications, can be encoded by a set of SMT constraints as below,*

$$\boxed{\Delta_{\mathsf{ReLU}} = \left\{ \begin{array}{l} \boldsymbol{x} \cdot \boldsymbol{W} > 0 \Rightarrow Z = \boldsymbol{x} \cdot \boldsymbol{W} \\ \boldsymbol{x} \cdot \boldsymbol{W} \leq 0 \Rightarrow Z = 0 \end{array} \right\}}$$

Definition 2.3. *(Probabilistic SMT model) Let $\boldsymbol{X}$ be a set of continuous random variables. A probabilistic SMT model is a pair $\mathcal{M} = (\Delta, \boldsymbol{W})$, where Δ is an SMT formula over $\boldsymbol{X}$*



(a) (b) (c)

Figure 1: **Exact Bayesian Model Averaging** for a Bayesian ReLU model. Figure 1a shows the choice of the approximate posterior to be uniform distributions and the likelihood function to be triangular distribution (cf. Section 2.2); Figure 1b shows the parameter space divided into four polytopes by logical constraints, each of which is weighted by some polynomial induced by the likelihood $p(y = 0 \mid \boldsymbol{x}, \boldsymbol{W})$, over which an integral would give predictive uncertainty $p(y = 0 \mid \boldsymbol{x})$; Figure 1c shows the full predictive distribution $p(y \mid \boldsymbol{x})$, which is piecewise and highly non-convex, as well as the marginalized prediction $\mathbb{E}[Y]$, both of which are computed by WMI solvers (cf. Section 2.1).

and a set of per-literal weights $\Phi = \{\phi_\ell\}_{\ell \in \mathcal{L}}$, where $\mathcal{L}$ are a set of SMT literals and each ϕ_ℓ are non-negative function defined over variables in literal ℓ. The unnormalized density of the probabilistic SMT model $\mathcal{M}$ is defined as the product of per-literal weights as below

$$p_{\mathcal{M}}(\boldsymbol{x}) = \prod_{\ell \in \mathcal{L}} \phi_\ell(\boldsymbol{x})^{[\![\boldsymbol{x} \models \ell]\!]}, \quad \boldsymbol{x} \models \Delta. \qquad (2)$$

Intuitively, the unnormalized density of a probabilistic SMT model is a piecewise functions where each piece is defined by a truth assignment to the literals in both SMT formula Δ and the literal set $\mathcal{L}$.

WMI is a recently proposed probabilistic inference framework that allows to perform marginalization over probabilistic SMT models. To develop WMI solvers has been an active research topic and several WMI solvers have been proposed for delivering exact and efficient inference, and WMI-based inference has been receiving increasing interest.

Definition 2.4. *(Weighted Model Integration) Let $\mathcal{M} = (\Delta, \Phi)$ be a probabilistic SMT model over real variables $\boldsymbol{X}$, the task of weighted model integration (WMI) is to compute*

$$\mathsf{WMI}(\Delta, \mathcal{W}) = \int_{\boldsymbol{x} \models \Delta} \prod_{\ell \in \mathcal{L}} \phi_\ell(\boldsymbol{x})^{[\![\boldsymbol{x} \models \ell]\!]} \, d\boldsymbol{x}. \qquad (3)$$

That is, the task is to integrate over the weighed assignments of $\boldsymbol{X}$ that satisfy the SMT formula Δ.

The encoding of Bayesian ReLU f into a probabilistic SMT model inspires us to cast the BMA problem as a *weighted model integration (WMI)* problem and leverage exact WMI solvers to provide accurate computation of the integration as in the definition of BMA in Equation 1.

We observe that if a Bayesian neural network model can be encoded as a probabilistic SMT model, the predictive uncertainty in BMA can be solved exactly by WMI solvers.

Proposition 2.5. *Given a Bayesian neural network model, if the neural network $f_{\boldsymbol{w}}$ can be encoded as an SMT formula $\Delta_{f_{\boldsymbol{w}}}$, and the likelihood function $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ as well as the approximate posterior $q(\boldsymbol{w})$ can be encoded as probabilistic SMT models $\mathcal{M}_{pred} = (\Delta_{pred}, \Phi_{pred})$ and $\mathcal{M}_{pos} = (\Delta_{pos}, \Phi_{pos})$ respectively, then the BMA problem can be solved by WMI solvers as shown below,*

$$\int p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \, q(\boldsymbol{w}) \, d\boldsymbol{w} = \frac{\mathsf{WMI}(\Delta \wedge (\boldsymbol{Y} = \boldsymbol{y}), \mathcal{W})}{\mathsf{WMI}(\Delta, \mathcal{W})},$$

$$\mathbb{E}_{p(\boldsymbol{y} \mid \boldsymbol{x})}[\boldsymbol{y}] = \frac{\mathsf{WMI}(\Delta, \mathcal{W}_{pos} \cup \mathcal{W}^*_{pred})}{\mathsf{WMI}(\Delta, \mathcal{W})},$$

with the SMT formula $\Delta = \Delta_{f_{\boldsymbol{w}}} \wedge \Delta_{pos} \wedge \Delta_{pred}$, weights $\mathcal{W} = \mathcal{W}_{pos} \cup \mathcal{W}_{pred}$, and weights $\Phi^*_{pred} = \{\phi^*_\ell(Y, \boldsymbol{W}_c) = Y \cdot \phi_\ell(Y, \boldsymbol{W}_c) \mid \phi_\ell \in \Phi_{pred}\}$.

2.2. Algorithmic Choice

Now the challenge is, how to choose the likelihood function $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ and approximate posterior $q(\boldsymbol{w})$ such that they are amenable to probabilistic SMT model encoding. One limitation of the existing WMI solvers is that they are only applicable when the weights in a probabilistic SMT model are (piecewise) polynomials, meaning that our choice are restricted to piecewise polynomials. As will show in the empirical results, the piecewise polynomial is sufficient to deliver surprisingly good empirical performance.

For an approximation $q(\boldsymbol{w}_c)$ to the conditional posterior $p(\boldsymbol{w}_c \mid \boldsymbol{w}_s, \mathcal{D})$, we choose it to be a uniform distribution which can be encoded into a probabilistic SMT

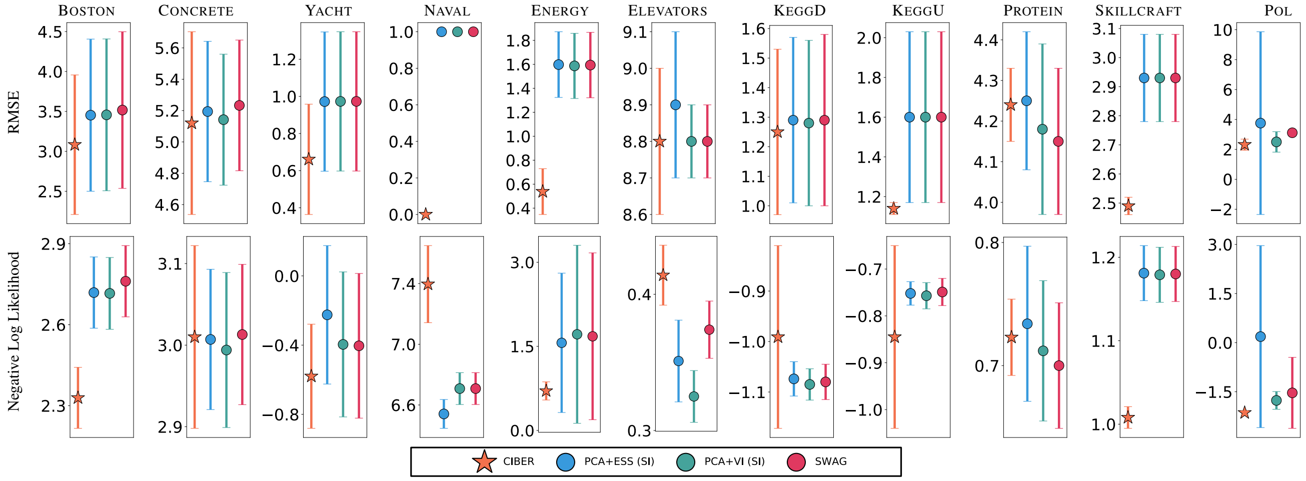


Figure 2: **Average test performance for UCI regression tasks** The first row reports root mean squared errors (RMSE) with standard deviations, where the results for dataset NAVAL are scaled by $1e3$, ELEVATORS scaled by $1e2$, and the four datasets, KEGGD, KEGGU, PROTEIN and SKILLCRAFT are scaled by $1e1$. The second row reports the negative log likelihood and standard deviation. The baseline results are reported by Izmailov et al. (2020).

model as $\mathcal{M}_{pos} = (\Delta_{pos}, \Phi_{pos})$ with the SMT formula being $\Delta_{pos} = \wedge_{i \in c}\, l_i \leq W_i \leq u_i$ and weights being $\Phi_{pos} = \{\phi_\ell(\boldsymbol{W}_c) = 1 \mid \ell = \texttt{true}\}$, where l_i and u_i are domain lower and upper bounds.

For the choice of likelihood function $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$, a common choice in existing literature is Gaussian distributions. To make the marginalization in collapsed BMA amenable to WMI solvers, we propose to use a triangular distribution as a piecewise polynomial approximation to Gaussian density, whose specific form is as follows,

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \frac{1}{r\sigma(\boldsymbol{x})} - \frac{|\boldsymbol{y} - f_{\boldsymbol{w}}(\boldsymbol{x})|}{r^2\sigma^2(\boldsymbol{x})}, |\boldsymbol{y} - f_{\boldsymbol{w}}(\boldsymbol{x})| \leq r\sigma(\boldsymbol{x})$$

where the constant r determines the shape of the triangular distribution and it is solved by minimizing the $L2$ distance between a standard Gaussian distribution and a symmetric triangular distribution parameterized by r; the $\sigma(\boldsymbol{x})$ is the variance estimation which might be independent from the input depending on whether the BNN is homoscedastic or heteroscedastic. The probabilistic SMT model encoding of the likelihood function $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ would then be as below.

$$\boxed{\begin{array}{l}
\Delta_{pred} = \left\{ \begin{array}{l} \boldsymbol{Y} - f_{\boldsymbol{w}}(\boldsymbol{x}) \leq r\sigma(\boldsymbol{x}) \\ \boldsymbol{Y} - f_{\boldsymbol{w}}(\boldsymbol{x}) \geq -r\sigma(\boldsymbol{x}) \end{array} \right\} \\
\Phi_{pred} = \left\{ \begin{array}{l} \phi_{\ell_1}(\boldsymbol{Y}, \boldsymbol{W}_c) = \frac{1}{r\sigma(\boldsymbol{x})} - \frac{\boldsymbol{Y} - f_{\boldsymbol{w}}(\boldsymbol{x})}{(r\sigma(\boldsymbol{x}))^2} \\ \phi_{\ell_2}(\boldsymbol{Y}, \boldsymbol{W}_c) = \frac{1}{r\sigma(\boldsymbol{x})} - \frac{f_{\boldsymbol{w}}(\boldsymbol{x}) - \boldsymbol{Y}}{(r\sigma(\boldsymbol{x}))^2} \end{array} \right\} \\
\textit{with } \ \ell_1 = \boldsymbol{Y} > f_{\boldsymbol{w}}(\boldsymbol{x}), \ell_2 = f_{\boldsymbol{w}}(\boldsymbol{x}) > \boldsymbol{Y}
\end{array}}$$

3. Experiments: UCI Regression

We evaluate our approach on standard datasets from the BNN literature: small and large UCI regression datasets, following the set-up of Izmailov et al. (2020). For the five small UCI datasets, they are *Boston*, *Concrete*, *Yacht*, *Naval* and *Energy*. and use a fully-connected network with a single hidden layer with 50 units with ReLU activation and two outputs parameterizing prediction and heteroscedastic variance. For the six large UCI datasets, they are *elevators*, *keggdirected*, *keggundirected*, *pol*, *protein* and *skillcraft*. On all datasets except *skillcraft* we use a feedforward network with five hidden layers of sizes $[1000, 1000, 500, 50, 2]$ with ReLU activation and two outputs parameterizing prediction and heteroscedastic variance.

Baselines. We compare our proposed collapsed inference algorithm called CIBER to the state-of-the-art approximate BNN inference methods: SWAG (Maddox et al., 2019), PCA+ESS (SI) (Izmailov et al., 2020) and PCA+VI (SI) (Izmailov et al., 2020), where they derive approximate posteriors by sampling from SGD trajectories as we do.

We summarize the experimental results in Figure 2. We observe that our proposed CIBER performs surprising well on prediction accuracy: CIBER outperforms all baselines in 10 our of 11 datasets. This is consistent with our conjecture that exact inference over conditional approximate posteriors helps achieve accurate estimation of the true BMA and it boosts prediction performance. On uncertainty estimation, we outperforms all baselines in 6 out of 11 datasets and has comparable performance on the others.

References

Clark Barrett, Leonardo de Moura, Silvio Ranise, Aaron Stump, and Cesare Tinelli. The SMT-LIB initiative and the rise of SMT. In *Proceedings of the 6th international conference on Hardware and software: verification and testing*, pages 3–3. Springer-Verlag, 2010.

Vaishak Belle, Andrea Passerini, and Guy Van den Broeck. Probabilistic inference in hybrid domains by weighted model integration. In *Proceedings of IJCAI*, pages 2770–2776, 2015.

Xi Chen, Jason D Lee, Xin T Tong, and Yichen Zhang. Statistical inference for model parameters in stochastic gradient descent. *The Annals of Statistics*, 48(1):251–273, 2020.

Rodrigo de Salvo Braz, Ciaran O'Reilly, Vibhav Gogate, and Rina Dechter. Probabilistic inference modulo theories. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3591–3599. AAAI Press, 2016.

Tiago M Fragoso, Wesley Bertoli, and Francisco Louzada. Bayesian model averaging: A systematic review and conceptual classification. *International Statistical Review*, 86(1):1–28, 2018.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International conference on computer aided verification*, pages 3–29. Springer, 2017.

Pavel Izmailov, Wesley J Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. In *Uncertainty in Artificial Intelligence*, pages 1169–1179. PMLR, 2020.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International Conference on Machine Learning*, pages 4629–4640. PMLR, 2021.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer, 2017.

Samuel Kolb, Martin Mladenov, Scott Sanner, Vaishak Belle, and Kristian Kersting. Efficient symbolic integration for probabilistic inference. In *IJCAI*, pages 5031–5037, 2018.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32:13153–13164, 2019.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

Paolo Morettin, Andrea Passerini, and Roberto Sebastiani. Efficient weighted model integration via SMT-based predicate abstraction. In *Proceedings of IJCAI*, pages 720–728, 2017.

Paolo Morettin, Andrea Passerini, and Roberto Sebastiani. Advanced smt techniques for weighted model integration. *Artificial Intelligence*, 275:1–27, 2019.

Zhe Zeng and Guy Van den Broeck. Efficient search-based weighted model integration. *Proceedings of UAI*, 2019.

Zhe Zeng, Paolo Morettin, Fanqi Yan, Antonio Vergari, and Guy Van den Broeck. Scaling up hybrid probabilistic inference with logical and arithmetic constraints via message passing. In *Proceedings of the International Conference of Machine Learning (ICML)*, 2020a.

Zhe Zeng, Paolo Morettin, Fanqi Yan, Antonio Vergari, and Guy Van den Broeck. Probabilistic inference with algebraic constraints: Theoretical limits and practical approximations. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, dec 2020b. URL `http://starai.cs.ucla.edu/papers/ZengNeurIPS20.pdf`.

Pedro Miguel Zuidberg Dos Martires, Anton Dries, and Luc De Raedt. Exact and approximate weighted model integration with probability density functions using knowledge compilation. In *Proceedings of the 30th Conference on Artificial Intelligence*. AAAI Press, 2019.