Approximate Bayesian inference with exchangeable distributions for neurosymbolic AI

Lennert De Smet
Department of Computer Science
KU Leuven
Leuven, Belgium
lennert.desmet@kuleuven.be

Abstract

Approximate Bayesian inference has driven significant advances in probabilistic machine learning at the basis of modern generative modelling. However, many successful Bayesian methods, such as variational inference or sequential Monte Carlo, often struggle in discrete domains with complex structure. In particular, their application to the challenges of neurosymbolic AI, i.e. learning and reasoning under uncertainty, is largely unexplored. Exploring the potential and opportunities of approximate Bayesian inference for neurosymbolic AI forms our three main contributions. First, the introduction of FiNS, a principled and general-purpose approach to learning and reasoning at scale that maintains statistical guarantees at test time. Second, a theoretical understanding of what properties a reasonable approximation to a neurosymbolic problem should have and how to practically enforce these properties in FiNS. Third, an empirical evaluation showing that FiNS reaches state-of-the-art performance on large neurosymbolic benchmarks at a favourable computational cost. The validation confirms FiNS provides approximations with symbolic properties such as consistency with background knowledge and generalisation from smaller to larger domains.

1 Introduction

Bayesian inference provides a wide range of tools to deal with uncertainty that have driven progress in probabilistic machine learning [74] and probabilistic programming [12, 18, 35]. These tools have been extended to deal with deep generative distributions modelling high-dimensional data [91, 93, 8] to great effect; image generation [92], audio generation [57], and now world models [43] all utilise Bayesian inference to some degree, mainly approximate Bayesian inference. Taking a Bayesian perspective is more than positing everything is a random variable that has to be dealt with according to probability theory. It can be a principled and pragmatic approach to statistical modelling that aims to balance the need for statistical rigour with the need for computational tractability.

Approximate Bayesian inference struggles in structured and discrete domains. Its successes are mostly apparent in continuous domains where the target probability distribution provides dense information, e.g. gradient-based Hamiltonian Monte Carlo [33, 75] (HMC) or optimisation-based variational inference [82, 49] (VI). If information becomes sparse or even discontinuous, then Bayesian methods struggle to provide reasonable approximations at scale. Such sparsity is characteristic for many problems that involve reasoning or any other type of structured prediction, especially over discrete domains. More generally, neurosymbolic problems that involve learning and reasoning under uncertainty seem to be a particularly challenging area for approximate Bayesian inference. Inference and learning in such problems is computationally hard [88, 63], but approximate Bayesian inference has a history of providing reasonable approximations to hard inference problems through algorithmic advances [36, 9].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

As an example, consider an autonomous driving agent asked for its next action. The agent is provided with uncertain information from its imperfect sensors and so its predicted actions are also inherently uncertain. Uncertainty is not problematic as long as probability is only placed on actions satisfying the laws of traffic. The traffic law is a hard constraint, either an action is legal or it is not. It is also sparse as only a few of the many possible prior actions might be legal. Such problems of updating a neural prior (actions) based on structured and sparse observations (traffic law) are of interest.

Hence, this paper explores an initial effort to adapt the success of approximate Bayesian inference to learning and reasoning under uncertainty. Specifically, we contribute the following. **(C1)** A scalable and general-purpose method, called FiNS, that adapts VI to the challenges of posterior inference over neurosymbolic problems (Section 3.1) and provides approximations consistent with symbolic constraints (Section 3.2.1). **(C2)** A theoretical framework for designing approximations that can generalise from smaller to larger domains (Section 3.2.2). **(C3)** An empirical validation that FiNS indeed provides consistent approximations and can zero-shot generalise what it learns on small domains to larger domains. The validation shows FiNS balances efficiency versus performance better than problem-specific solutions and reaches state-of-the-art accuracy. FiNS is ground in statistical principles and evaluates positively in terms of scalability, consistency and generalisation on a range of neurosymbolic problems (Section 5). In a broader context, it is a step towards extending the inference boundaries of highly-structured universal probabilistic programs [41].

2 Preliminaries

2.1 Variational inference casts Bayesian inference as optimisation

Bayesian inference takes a prior distribution $p(\boldsymbol{X})$ over a random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ together with instances $\boldsymbol{o}$ of a random vector $\boldsymbol{O}$ of observations and aims to compute the posterior distribution $p(\boldsymbol{X} \mid \boldsymbol{O} = \boldsymbol{o})$. Via Bayes' rule, one finds that

$$p(\boldsymbol{X} \mid \boldsymbol{O} = \boldsymbol{o}) = \frac{p(\boldsymbol{O} = \boldsymbol{o} \mid \boldsymbol{X}) p(\boldsymbol{X})}{p(\boldsymbol{O} = \boldsymbol{o})}, \tag{1}$$

where it is assumed that the *likelihood* $p(\boldsymbol{O} \mid \boldsymbol{X})$ and *prior* $p(\boldsymbol{X})$ are easy to evaluate. Even then, the observation probability $p(\boldsymbol{O} = \boldsymbol{o})$ in the denominator generally becomes intractable.

Variational inference [49] (VI) avoids computing $p(\boldsymbol{O})$ by directly fitting a parametrised family of distributions $q_{\boldsymbol{\theta}}(\boldsymbol{X})$ to $p(\boldsymbol{X} \mid \boldsymbol{O})$. This family is called the *variational family* and its elements *variational distributions*. A variational distribution is fit to the posterior by minimising a statistical divergence. The most common divergence is the Kullback-Leibler (KL) divergence and minimising $\mathsf{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid\mid p(\boldsymbol{X} \mid \boldsymbol{O}))$ is equivalent to minimising the negated evidence lower bound (ELBO)

$$\mathsf{KL}\left(q_{\boldsymbol{\theta}}(\boldsymbol{X}) \mid\mid p(\boldsymbol{X})\right) - \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{X})}\left[\log\left(p(\boldsymbol{O} \mid \boldsymbol{X})\right)\right]. \tag{2}$$

The choice of variational family trades off between computational efficiency and approximation quality. Some choices, such as the mean-field family $q_{\boldsymbol{\theta}}(\boldsymbol{X}) = \prod_{i=1}^{n} q_{\boldsymbol{\theta}}(X_i)$, can allow for exact computation and differentiation of the ELBO depending on the form of the likelihood and prior. This analytic tractability does come at the cost of expressivity as the mean-field family is often too restrictive to capture the true posterior distribution. Alternatively, expressive variational distributions can be fit using black-box variational inference (BBVI) [87] at the cost of computational efficiency.

2.2 Neurosymbolic AI combines learning and reasoning

Neurosymbolic AI combines probabilistic machine learning with symbolic computation into models that learn to predict concepts from data and symbolically reason under uncertainty [37, 68]. Symbolic is used in the general sense of formal languages with well-defined semantics. Formal languages, say programs, represent knowledge in the form of structured sentences and can evaluate sentences in accordance with their semantics. Such semantic evaluations are represented in modern neurosymbolic models by structured computational graphs [19, 102]. Hence, neurosymbolic models can be seen as programmable computational graphs that perform both neural and symbolic computation (Figure 1).

A formal system consists of a language with a coherent syntax and semantics. Direct examples are logical languages such as first-order logic [6], but formal systems also include many programming languages. For instance, the language Lean [73] for expressing and proving mathematical theorems,

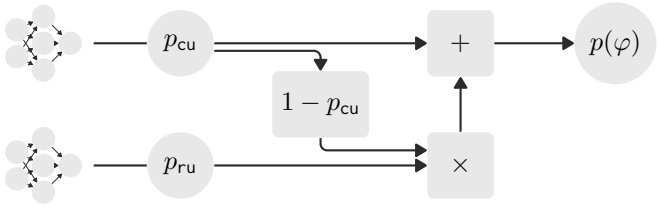


Figure 1: The computational graph of the probability of the sentence `column_unique` $\vee$ `row_unique`. The probabilities $p_{\texttt{cu}}$ and $p_{\texttt{ru}}$ of the leaf nodes are predicted by a neural network while the probability of the root node $p(\varphi)$ is computed by the symbolic (algebraic) computational graph.

but also more general-purpose languages, such as Python or Haskell [99, 34]. Even certain fragments of natural language can be considered formal systems [72]. In these broader systems, "semantics" define how a program executes given a specific state or set of inputs. Such general formal systems are considered in this work to align with recent efforts to define neurosymbolic AI in a broad sense [23].

We abstract both logical evaluation and programmatic execution into a unified mathematical notation. A language L is a set of sentences, e.g. logical formulas or program scripts, formed with a set of symbols $\mathcal{S}$ according to a well-defined syntax. Symbols are given meaning by an interpretation $\omega : \mathcal{S} \to \mathcal{T}$ that assigns a specific *semantic value* to each symbol from a target space $\mathcal{T} \subseteq \mathbb{R}^+$. For instance, an interpretation could assign truth values to logical propositions or concrete values to the input variables of a Python program. The set $\mathcal{T}^{\mathcal{S}}$ of all possible interpretations is denoted by Ω.

Interpretations of symbols form the basis of the semantics of the language. They determine a meaning for each symbol in the language that induces the meaning of more complex sentences. Formally, a semantics is defined as a function $\mu : L \times \Omega \to \mathcal{T}$ that outputs a semantic value for a given sentence under a specific interpretation. While μ is often defined inductively on the structure of the sentence, such as resolving logical connectives or executing a sequence of code, it acts mathematically as a fixed evaluation mapping (Example 2.1).

Example 2.1. Propositional sentences are formed by combining propositional symbols, such as `column_unique` (`cu`) or `row_unique` (`ru`) in a sudoku puzzle, using logical connectives like $\wedge$ (and) and $\vee$ (or). The Boolean semantics μ_B of propositional logic sentences is determined by a Boolean interpretation $\omega : \{\texttt{cu}, \texttt{ru}\} \to \{\texttt{true}, \texttt{false}\}$ of the propositions. If $\omega(\texttt{cu})$ is `true` and $\omega(\texttt{ru})$ is `false`, then $\mu_B(\texttt{cu} \wedge \texttt{ru}, \omega)$ is `false` while $\mu_B(\texttt{cu} \vee \texttt{ru}, \omega)$ is `true`. A simple propositional neurosymbolic model can predict probabilities $p_{\texttt{cu}}$ and $p_{\texttt{ru}}$ for the propositions `cu` and `ru` and use a symbolic computational graph to compute the probability of more complex sentences (Figure 1).

3 Probabilistic neurosymbolic computation as Bayesian inference

A probabilistic neurosymbolic model takes a formal system (L, μ) and a probability distribution $p_{\boldsymbol{\lambda}}(\omega)$ parametrised by $\boldsymbol{\lambda}$ to perform probabilistic inference over (unnormalised) joint distributions $p^{\mu}_{\boldsymbol{\lambda}}(\omega, \varphi) := \mu(\varphi, \omega) p_{\boldsymbol{\lambda}}(\omega)$ of interpretations ω and the semantic values of a sentence φ of L. One such type of inference is computing the posterior distribution $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$ over interpretations given some observed sentences. That is, computing the distribution over symbol interpretations that are consistent with the observed sentences. The distribution $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$ will be called the *neurosymbolic posterior* because it is the result of combining a neural prior $p_{\boldsymbol{\lambda}}(\omega)$ and a symbolic likelihood $\mu(\varphi, \omega)$.

Neurosymbolic posteriors are central to neurosymbolic AI. Existing neurosymbolic models often learn the parameters $\boldsymbol{\lambda}$ of $p_{\boldsymbol{\lambda}}(\omega)$ by maximising the marginal likelihood $p^{\mu}_{\boldsymbol{\lambda}}(\varphi) = \int_{\Omega} \mu(\varphi, \omega) p_{\boldsymbol{\lambda}}(\omega) \, \mathrm{d}\omega$ [104, 64, 58]. While this approach allows for learning through distant supervision, the resulting optimised $p_{\boldsymbol{\lambda}}(\omega)$ does not have the same interpretation as the posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$ of predicting interpretations ω consistent with the sentence φ. Additionally, it can cause reasoning shortcuts [66], because it leads to assigning high probability to a limited set of interpretations. This ambiguity is better captured by a Bayesian approach that models a distribution over valid interpretations subject to prior uncertainty [1, 67]. Consequently, probabilistic neurosymbolic models should focus on computing $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi) \propto \mu(\varphi, \omega) p_{\boldsymbol{\lambda}}(\omega)$ based on an explicit prior uncertainty $p_{\boldsymbol{\lambda}}(\omega)$ and a likelihood $p^{\mu}(\varphi \mid \omega) = \mu(\varphi, \omega)$ that encodes the semantics of the observed sentences.

Prior and semantics determine the probabilistic dependencies. The prior $p_{\boldsymbol{\lambda}}(\omega)$ encodes the dependencies between symbols before demanding any consistency with an observed sentence. It is often assumed to factorise over symbols [14] for tractability, but with consequences for expressivity [97]. Even if the prior factorises fully over symbols, the neurosymbolic posterior need not do so as the semantics μ may introduce dependencies between symbols. In fact, even simple priors and sentences can lead to complex neurosymbolic posteriors.

Example 3.1. Consider a logical language with symbols a, b and $\leftrightarrow$, and Boolean semantics μ_B that map to $\mathcal{T} = \{0, 1\}$. The sentence $\mathsf{a} \leftrightarrow \mathsf{b}$ is Boolean satisfied if and only if $\omega(\mathsf{a}) = \omega(\mathsf{b})$. If a and b are assumed independent and uniform, i.e. $p_{\boldsymbol{\lambda}}(\mathsf{a}, \mathsf{b}) = p_{\boldsymbol{\lambda}}(\mathsf{a}) \cdot p_{\boldsymbol{\lambda}}(\mathsf{b})$ and $p_{\boldsymbol{\lambda}}(\mathsf{a}) = p_{\boldsymbol{\lambda}}(\mathsf{b}) = 0.5$, then the neurosymbolic posterior $p_{\boldsymbol{\lambda}}^{\mu_B}(\mathsf{a}, \mathsf{b} \mid \mathsf{a} \leftrightarrow \mathsf{b})$ does not factorise. Indeed, it assigns probability 0.5 to the two interpretations where a and b have the same truth value and 0 otherwise.

3.1 Neurosymbolic posteriors challenge Bayesian inference

The complexity of computing neurosymbolic posteriors motivates the use of approximate Bayesian inference. Neurosymbolic posteriors will often have dependent variables due to sentences φ that involve many variables (Example 3.1). Such dependencies challenge exact inference methods in neurosymbolic AI, because they generally rely on compiling into a compact representation where inference is, ideally, linear in the size of the representation [19, 101]. The size of the representation is itself, unfortunately, exponential in the treewidth of the dependency graph of the posterior [3], which can roughly be interpreted as the maximum number of variables dependent on each other. Hence, any exact representation of the neurosymbolic posterior is infeasible for large problems. The same complexity issues apply to more general exact probabilistic inference methods outside of neurosymbolic AI [106, 27]. While it is possible to use exact inference in an approximate representation, it is not clear how to automatically construct a reasonable approximate representation for a given neurosymbolic posterior. Conversely, approximate inference methods in the exact neurosymbolic model are tailored to optimising $p_{\boldsymbol{\lambda}}^{\mu}(\varphi)$ instead of aiming for the posterior $p_{\boldsymbol{\lambda}}^{\mu}(\omega \mid \varphi)$. As a result, it merits exploring approximate Bayesian inference for neurosymbolic posteriors.

Variational inference with neurally parametrised distributions can capture posterior dependencies. The exponential scaling of exact representations of neurosymbolic posteriors motivates using neural networks as expressive approximators. Indeed, neural nets try to learn to disentangle dependencies in their high-dimensional parameter spaces [7, 59]. Applying neural approximations to Bayesian inference leads us to explore deep variational inference [105] for neurosymbolic posterior inference. The following paragraphs highlight the problems of applying VI and present potential solutions.

If $p_{\boldsymbol{\lambda}}^{\mu}(\omega \mid \varphi)$ is the target of a variational approximation $q_{\boldsymbol{\theta}}(\omega)$, then neurosymbolic posterior computation is equivalent to minimising the negated ELBO

$$\mathsf{KL}\left(q_{\boldsymbol{\theta}}(\omega) \,||\, p_{\boldsymbol{\lambda}}(\omega)\right) - \mathbb{E}_{q_{\boldsymbol{\theta}}(\omega)}\left[\log \mu(\varphi, \omega)\right], \tag{3}$$

which can be extended to allow for optimising the prior with respect to $\boldsymbol{\lambda}$ as well as a generative component (Appendix A). **Optimisation of Equation 3 introduces three important problems. (P1)** The expectation of the semantics over interpretations is intractable due to the size of the domain $\Omega = \mathcal{T}^{\mathcal{S}}$ and the fact that $\mu(\varphi, \cdot)$ generally does not decompose over symbols. **(P2)** The sparse semantics μ of hard constraints cause collapsing estimates that can prevent successful optimisation. For instance, Boolean semantics only return 1 for an often limited number of satisfying interpretations and 0 otherwise. **(P3)** The variational distributions $q_{\boldsymbol{\theta}}$ should be suited to the neurosymbolic context.

Problem (P1) is mitigated by applying black-box variational inference (BBVI). BBVI uses Monte Carlo sampling and black-box gradient estimators to approximate the intractable $\mathbb{E}_{q_{\boldsymbol{\theta}}(\omega)}\left[\log \mu(\varphi, \omega)\right]$ and its gradients. The choice of gradient estimator is crucial to the success of BBVI. We use either REINFORCE with a leave-one-out baseline [53] or Rao-Blackwellised [86, 13] estimators derived from the CatLog-derivative trick [26] for their state-of-the-art performance.

Problem (P2) is resolved through asymptotic relaxations of the semantics. When direct optimisation fails, relaxations of the semantics can be used as a surrogate objective. Ideally, any relaxation μ_{r} should at least match μ in expectation, i.e. $\mathbb{E}_{p_{\boldsymbol{\lambda}}(\omega)}\left[\mu(\varphi, \omega)\right] = \mathbb{E}_{p_{\boldsymbol{\lambda}}(\omega)}\left[\mu_{\mathrm{r}}(\varphi, \omega)\right]$, as a sufficient condition to have the resulting relaxed negated ELBO

$$\mathsf{KL}\left(q_{\boldsymbol{\theta}}(\omega) \,||\, p_{\boldsymbol{\lambda}}(\omega)\right) - \mathbb{E}_{q_{\boldsymbol{\theta}}(\omega)}\left[\log \mu_{\mathrm{r}}(\varphi, \omega)\right] \tag{4}$$

be an upper bound of $-\log \mathbb{E}_{p_{\boldsymbol{\lambda}}(\omega)}\left[\mu(\varphi, \omega)\right]$. For general programs, we use the algorithmic relaxations of Petersen et al. [81] that asymptotically converge to the true semantics. For logical languages with Boolean semantics, we introduce a simple inductive relaxation with a similar asymptotic guarantee as follows.

Assume without loss of generality that the sentence φ is in conjunctive normal form (CNF) [94], i.e. $\varphi \leftrightarrow \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} l_{ij}$, where each l_{ij} is a literal. Then, we relax the semantics of literals $\mu_r(l_{ij}, \omega)$ to ε if $\mu(l_{ij}, \omega) = 0$ and $1 - \varepsilon$ otherwise, for $0 < \varepsilon < 0.5$. These relaxations inductively propagate to yield the relaxed semantics of the sentence φ as $\mu_r(\varphi, \omega) = \bigotimes_{i=1}^{n} \bigoplus_{j=1}^{m_i} \mu_r(l_{ij}, \omega)$, where $\oplus$ and $\otimes$ are the algebraic operations of the product algebra [44] on $[0, 1]$ (Appendix B). The relaxed semantics μ_r converge to μ when $\varepsilon \to 0$ by construction, which however only ensure a negated ELBO that upper bounds $-\log \mathbb{E}_{p_{\boldsymbol{\lambda}}(\omega)}\left[\mu(\varphi, \omega)\right]$ in the limit of $\varepsilon \to 0$. This shortcoming is addressed jointly with the lack of guarantees of deep VI with respect to the true posterior $p_{\boldsymbol{\lambda}}^{\mu}(\omega \mid \varphi)$ (Section 3.2.1).

Problem (P3) on suitable variational distributions will be discussed in the next sections. Suitability will be assessed by the ability to emulate two key properties of symbolic computation; consistency with constraints and generalisation from smaller to larger domains.

3.2 Desirable symbolic properties of neural approximations

One might argue that a neural approximation of a neurosymbolic posterior defeats the purpose of incorporating symbols in the first place. Indeed, *symbolic computation comes with a number of benefits that are lost when implicitly encoded in the weights of a neural network.* **(i)** It guarantees that the output is consistent with the semantics and knowledge, and **(ii)** it provides relational capabilities that enable generalisation to unseen domains. For instance, not any neural network trained to predict the shortest path on a 12×12 grid can correctly predict shortest paths on a 30×30 grid (Property **(ii)**). Moreover, deep VI does not provide guarantees of convergence to $p_{\boldsymbol{\lambda}}^{\mu}(\omega \mid \varphi)$ (Property **(i)**).

3.2.1 Consistent posterior approximations

Statistical machine learning has shifted to learning proposals for estimation [89]. Statistical estimation via importance sampling [100, 79], Markov Chain Monte Carlo (MCMC) [70, 40] or sequential Monte Carlo (SMC) [42] provide statistical guarantees with respect to a target distribution, but they often suffer when the proposal distribution is far from the target [4, 85]. Hence, modern statistical machine learning opts to learn a reasonable parametrised proposal distribution from available data through maximum likelihood estimation (MLE) or by using VI [48].

Consistency (Property **(i)**) can be achieved by following statistical machine learning. Consistency is not guaranteed for the variational approximation $q_{\boldsymbol{\theta}}(\omega)$. Even more, a relaxed ELBO for $q_{\boldsymbol{\theta}}(\omega)$ (Equation 4) might not lower bound the observational log-likelihood. However, $q_{\boldsymbol{\theta}}(\omega)$ should still be a reasonable proposal for statistical estimation, such as self-normalised importance sampling (SNIS) [39, 80]. SNIS is a one-step estimation technique that delivers consistent estimates of $\mathbb{E}_{p(\boldsymbol{X})}\left[f(\boldsymbol{X})\right]$ when only the unnormalised target $\tilde{p}(\boldsymbol{X})$ and a proposal distribution $q(\boldsymbol{X})$ are available. It estimates expectations by

$$\mathbb{E}_{p(\boldsymbol{X})}\left[f(\boldsymbol{X})\right] \approx \sum_{n=1}^{N} \frac{w_n \cdot f\left(\boldsymbol{x}^{(n)}\right)}{\sum_{j=1}^{N} w_j}, \quad \text{where} \quad w_n = \frac{\tilde{p}\left(\boldsymbol{x}^{(n)}\right)}{q\left(\boldsymbol{x}^{(n)}\right)} \quad \text{and} \quad \boldsymbol{x}^{(n)} \sim q(\boldsymbol{X}). \tag{5}$$

Put differently, SNIS estimates are based on the approximation

$$p(\boldsymbol{X}) \approx \sum_{n=1}^{N} \frac{w_n}{\sum_{j=1}^{N} w_j} \delta\left(\boldsymbol{X} - \boldsymbol{x}^{(n)}\right), \tag{6}$$

with δ the Dirac-delta distribution [90]. The neurosymbolic posterior $p_{\boldsymbol{\lambda}}^{\mu}(\omega \mid \varphi)$ only provides unnormalised information via $p_{\boldsymbol{\lambda}}^{\mu}(\omega \mid \varphi) \propto \mu(\varphi, \omega) p_{\boldsymbol{\lambda}}(\omega)$ and hence admits the SNIS approximation for samples $\omega^{(n)} \sim q_{\boldsymbol{\theta}}(\omega)$

$$p_{\boldsymbol{\lambda}}^{\mu}(\omega \mid \varphi) \approx \sum_{n=1}^{N} \frac{w_n}{\sum_{j=1}^{N} w_j} \delta\left(\omega - \omega^{(n)}\right), \quad \text{where} \quad w_n = \frac{\mu\left(\varphi, \omega^{(n)}\right) p_{\boldsymbol{\lambda}}\left(\omega^{(n)}\right)}{q_{\boldsymbol{\theta}}\left(\omega^{(n)}\right)}. \tag{7}$$

This approximation weighs samples according to their consistency with φ. In particular, **it assigns zero weight to any interpretation that is inconsistent with** φ, because then $w_n \propto \mu\left(\varphi, \omega^{(n)}\right) = 0$. It also has some degree of intervenability, because changing the knowledge, i.e. changing the sentence φ to a new sentence ψ, changes the weights w_n. If $q_{\boldsymbol{\theta}}$ assigns sufficient probability mass to the support of the posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \psi)$, the SNIS estimate will adapt to the intervened setting. The success of SNIS can be measured by the effective sample size (ESS) [52] that gives an actionable measurement of the consistency of the variational approximation (Appendix C).

3.2.2 Exchangeable random variables characterise posteriors that generalise

Relational generalisation (Property **(ii)**) is used to roughly describe the ability of a model to remain a capable predictor when the number of variables changes. It is often implicitly assumed to hold for statistical models. However, generalisation is not inherent to all statistical models. The mismatch between the expected behaviour and practical behaviour of statistical models has been attributed to not explicitly incorporating ones' expectations in the model [69]. It begs the following question.

"How to construct statistical models that generalise relationally?"

We answer this question by connecting generalisation to the notion of exchangeability that, in turn, provides functional characterisations of distributions with generalisation capabilities through foundational results from probability theory.

Relational generalisation of statistical models is formalised by the notion of extension. Intuitively, extension means that there is a potentially infinite sequence of random variables $(X_n)_{n\in\mathbb{N}}$ that we want our model to generalise to by only observing finite subsequences $(X_{i_1}, \dots, X_{i_k})$ for $k \in \mathbb{N}$ and $i_1, \dots, i_k \in \mathbb{N}$. In other words, we want to extend smaller distributions $p(X_{i_1}, \dots, X_{i_k})$ to an arbitrarily large distribution $p(X_1, X_2, \dots)$ that is consistent with each smaller $p(X_{i_1}, \dots, X_{i_k})$. For example, we might only have data on paths in a 12×12 grid and would like to learn a model that can also predict paths in 30×30 grids. Formally, an extension can be obtained by the Kolmogorov extension theorem [51] and a sufficient condition for extension is that the sequence $(X_1, X_2, \dots)$ is exchangeable. That is, the sequence is invariant to finite permutation.

Definition 3.1 (Exchangeable random variables)**.** A sequence $(X_n)_{n\in\mathbb{N}}$ of random variables is called exchangeable if, for any permutation π of $\mathbb{N}$ that only moves a finite number of elements, it holds that $p(X_1, X_2, \dots) = p(X_{\pi(1)}, X_{\pi(2)}, \dots)$.

The distributions of infinite exchangeable sequences satisfy a range of representation theorems [45] initialised by de Finetti [20], i.e. they are mixtures of conditionally i.i.d. distributions.

Theorem 3.1 (Exchangeability representation theorem [20, 45])**.** If $(X_n)_{n\in\mathbb{N}}$ is an infinite exchangeable sequence of random variables, then there is a distribution $p(\boldsymbol{\theta})$ such that for all $k \in \mathbb{N}$ and all distinct $i_1, \dots, i_k \in \mathbb{N}$ it holds that

$$p(X_{i_1}, \dots, X_{i_k}) = \int \prod_{j=1}^{k} p(X_{i_j} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}. \tag{8}$$

The theorem makes the requirements for a statistical model to exhibit infinite relational generalisation quite practical. Any distribution aiming to generalise from a finite to an infinite number of variables must be an integral mixture of conditionally i.i.d. distributions. That is, there is an underlying "mixing distribution" $p(\boldsymbol{\theta})$ that captures the dependencies between the variables. Such mixtures are straightforward to incorporate in variational distributions as even a mean-field approximation is a (one-component) mixture. Our experiments utilise additive mixtures where the mixing distribution $p(\boldsymbol{\theta})$ is a neural network [32]. Exploring other mixtures [46, 101, 60, 62] is left for future work.

Perhaps demanding infinite generalisation is too strong and it suffices to generalise to a finite upper limit n of variables from observing only at most k. This finite generalisation can be operationalised by considering finite exchangeable sequences of random variables and their representation theorems. More details on how to operationalise finite generalisation in neural variational approximations are provided in Appendix D as we do not directly use it in our experiments.

Connected efforts in equivariant neural networks and statistical relational AI do not cover our questions. Equivariant neural networks have been thoroughly connected to probabilistic symmetries [10]

by considering theoretical functional forms that are invariant to a group action. However, they do not link symmetries to generalisation and it is not clear how to operationalise their functional forms into neurally parametrised distributions that generalise. Generalisation has been studied in the context of statistical relational AI [22] by considering projective families of distributions [47, 103], which are closely linked to exchangeable distributions. This study focuses on statistical *relational* models and their generalisation properties [15], while we use established results from statistics in terms of representation to find general statistical models that generalise.

Remark 3.1 (A note on lifted inference)**.** Exploiting symmetries such as exchangeability is at the core of lifted inference [95]. Symmetries lead to factorised functional forms that in turn facilitate more efficient inference [76, 77, 84]. Our discussion on the connection between extendability and exchangeability (Theorem 3.1) shows that these functional forms also lead to generalisation.

3.3 Consistent and generalising variational approximations with FiNS

The solutions to the three problems **(P1)**, **(P2)** and **(P3)** result in a three-step algorithm dubbed FiNS (de Finetti NeuroSymbolic inference) given in Algorithm 1. FiNS applies BBVI with relaxed semantics and recovers constraint consistency with SNIS. The approximations can be exchangeable to facilitate generalisation to arbitrarily many variables.

Algorithm 1 FiNS for hard constraints with exchangeability

Require: **(exchangeable)** $q_{\boldsymbol{\theta}}(\omega)$ (Theorem 3.1), prior $p_{\boldsymbol{\lambda}}(\omega)$ and **sparse** likelihood $\mu(\varphi, \omega)$
Ensure: approximation $q^*_{\boldsymbol{\theta}}$ of posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$ consistent with φ

1: $\mu_r \leftarrow \textsc{InductiveRelaxation}(\mu)$ ▷ *Step 1: relax the exact likelihood (Appendix B)*
2: $q^+_{\boldsymbol{\theta}} \leftarrow \text{BBVI}(q_{\boldsymbol{\theta}}, p_{\boldsymbol{\lambda}}, \mu_r)$ ▷ *Step 2: train q on the relaxation with BBVI*
3: $q^*_{\boldsymbol{\theta}} \leftarrow \text{SNIS}(q^+_{\boldsymbol{\theta}}, p_{\boldsymbol{\lambda}}, \mu)$ ▷ *Step 3: enforce consistency with SNIS (Section 3.2.1)*
4: **return** $q^*_{\boldsymbol{\theta}}$

4 Related work

FiNS can be related to past efforts in applying variational inference to statistical relational models [22]. For instance, the idea of using exchangeable distributions through Theorem 3.1 appears in lifted relational variational inference (LRVI) [17] or lifted hybrid variational inference [16]. However, both approaches only work on (relational) Markov random fields [38] that have dense potential functions, while FiNS explicitly focuses on consistency with sparse constraints (Section 3.1). Moreover, they are not integrated with deep learning; only statistical models on structured data are considered, not deep probabilistic models where constraints are defined on unstructured data. Deep and universal probabilistic programming [8] relies heavily on programmable variational inference methods [55, 18], yet these fail to provide consistent approximations in highly-structured, discrete domains.

Scaling neurosymbolic inference has received attention in recent years from two main perspectives; exact inference in approximate models, or approximate inference in exact models. Exact inference scales by restricting to a less expressive language, e.g. probabilistic inference over a structured probabilistic circuit (PC) [101] or over linear arithmetic expressions (PLIA$_t$). PLIA$_t$ applies the fast Fourier transform for efficient computation of the probability of one linear expression. Generally, these methods rely on an available approximation of the intended model, which might not always exist, while FiNS applies to any problem with a known constraint φ. Moreover, there are theoretical questions surrounding the expressive efficiency of tractable approximate models [56]. Parametrised tractable models can none the less be an interesting choice of variational family $q_{\boldsymbol{\theta}}$ for FiNS.

Approximate inference for the exact model can be approached from a more logical, statistical or neural point of view. A-NeSI learns a neural surrogate of the satisfaction probability $p^{\mu}_{\boldsymbol{\lambda}}(\varphi)$ by generating a large dataset of synthetic interpretation distributions $p(\omega)$. Once a differentiable surrogate is learned, which becomes harder as the complexity of the constraint increases, it is used to optimise the neurally parametrised distribution $p_{\boldsymbol{\lambda}}(\omega)$. FiNS avoids expensive surrogate learning through approximate Bayesian inference with a relaxed target. More recent work on neural approximations goes beyond the common independence assumption in neurosymbolic AI by adapting expressive masked diffusion models to the neurosymbolic setting (NeSyDM). NeSyDMs optimise a variational lower bound with

respect to the masking process while the diffusion model targets maximum likelihood. Instead, FiNS' variational approximation directly targets the posterior. Additionally, A-NeSI and NeSyDM do not provide guarantees of constraint satisfaction and can not generalise from smaller to larger domains. From a statistical point of view, I-MLE [78] proposes a biased gradient estimator to approximately maximise the likelihood $p^{\mu}_{\boldsymbol{\lambda}}(\varphi)$. The estimator assumes a MAP solver is available and that the input to the symbolic problem is continuous . These assumptions make the application of I-MLE to many symbolic problems challenging, as many rely on discrete inputs where continuous relaxations are unavailable (Visual sudoku). Even if a continuous relaxation exists (Warcraft path planning), the output solution is then also a continuous approximation of the true underlying discrete solution. FiNS is generally applicable to symbolic problems with both continuous and discrete inputs and does not require a MAP solver. Finally, logical approximations can focus on searching for the k-best solutions and learn to maximise their probability (Scallop and DPLA*) [58, 65] instead of maximising the likelihood of all possible solutions. This approximation can reduce theoretical complexity from $\#P$-hard to NP-hard at the cost of top-k bias while still scaling poorly. Instead of only using solutions, which are hard to find, FiNS samples both valid and invalid interpretations that are scored with a relaxed likelihood to guide the approximation to satisfying interpretations.

5 Experiments

Our experiments aim to answer three empirical questions to show Bayesian inference with FiNS can yield meaningful approximations to neurosymbolic posteriors.[1]

(RQ1) Do our approximations scale to large neurosymbolic problems?
(RQ2) Are our approximations consistent (Property **(i)**)?
(RQ3) Do our approximations zero-shot generalise to larger problems (Property **(ii)**)?

5.1 Datasets and setup

The questions will be answered by considering two challenging neurosymbolic datasets; visual sudoku puzzles [5] and Warcraft shortest path planning [83]. More details can be found in Appendix E.

Visual Sudoku. The goal of the visual sudoku task is to predict whether a given image of a sudoku puzzle with handwritten digits (Figure 2, left) is valid. The input is an image of a $G \times G$ sudoku puzzle and an interpretation ω consists of the random variables representing the digits in the cells of the sudoku puzzle. That is, ω_{ij} represents the categorical random variable for the digit in the i-th row and j-th column of the sudoku puzzle with G possible categories. Note that the space Ω of interpretations has the combinatorial cardinality G^{G^2} showing the difficulty of the problem. The sentence φ encodes the rules of sudoku, i.e. that each row, column and subgrid must contain each digit exactly once. A solution to the problem should provide a probability of a given image satisfying the rules of sudoku. As satisfaction is a binary variable, all models are evaluated by thresholding the provided probability at 0.5 and comparing it to the ground truth labels. FiNS uses a mean-field approximation parametrised by a graph neural network (GNN) [11].

Warcraft shortest path planning. The aim of the Warcraft shortest path planning task is to take an image of a map from the game Warcraft (Figure 2, right) and predict the shortest path from the top left to the bottom right corners of the map. Each map has an underlying grid of $G \times G$ tiles that can each have one of five types, e.g. grassland or water. All 5^{G^2} possible assignments to the tiles of the map form the space of interpretations to navigate. Each assignment is turned into a path by applying Dijkstra's algorithm following the setup of other works [83, 96], which induces a distribution over paths. Each path should satisfy the constraint φ that it equals the true shortest path. Here, a solution is considered correct if the path with highest probability has the same cost as the true shortest path. A simple neural mixture [32] is used by FiNS as variational distribution.

5.2 (RQ1) Scaling learning and reasoning

We positively answer the question of scalability by evaluating the performance of FiNS in terms of accuracy and training time. Results for visual sudoku are shown in Table 1 and for Warcraft

[1] Code is available at https://github.com/ML-KULeuven/fins

Table 1: Median accuracy with 25th and 75th quantiles and average training time (minutes) for differently sized visual sudoku puzzles taken across 5 independent training runs.

	4×4		9×9	
Method	**Accuracy (↑)**	**Time (↓)**	**Accuracy (↑)**	**Time (↓)**
Scallop	$75.00^{+0.50}_{-0.50}$	1.52 min	T/O	T/O
PLIA$_t$	$90.50^{+0.50}_{-0.50}$	0.41 min	$61.00^{+1.00}_{-1.00}$	2.59 min
A-NeSI	$88.50^{+0.00}_{-2.50}$	46.08 min	$59.50^{+0.50}_{-2.50}$	73.33 min
FiNS	$91.50^{+0.50}_{-1.00}$	2.39 min	$61.00^{+1.50}_{-4.50}$	26.45 min

Table 2: Median accuracy with 25th and 75th quantiles and average training time (minutes) for Warcraft shortest path planning taken across 5 independent training runs.

	12×12		30×30	
Method	**Accuracy (↑)**	**Time (↓)**	**Accuracy (↑)**	**Time (↓)**
I-MLE	$95.30^{+0.40}_{-0.00}$	10.86 min	$95.20^{+\ 0.10}_{-\ 0.00}$	42.94 min
A-NeSI	$99.33^{+0.08}_{-0.17}$	436.74 min	$93.33^{+\ 2.58}_{-68.25}$	1595.65 min
NeSyDM	$99.40^{+0.10}_{-0.20}$	129.96 min	$98.40^{+\ 0.30}_{-\ 0.10}$	183.18 min
FiNS	$99.70^{+0.20}_{-0.30}$	33.02 min	$98.60^{+\ 0.10}_{-\ 0.40}$	37.52 min
FiNS (Gen.)	–	–	$99.80^{+\ 0.20}_{-\ 0.80}$	33.02 min

shortest path planning in Table 2 comparing to the state-of-the-art methods from the neurosymbolic literature. In general, **FiNS scales well to even the largest instances of both tasks and provides a good trade-off between accuracy and training time**. A-NeSI performs well in classifying visual sudokus and smaller shortest paths, but fails to provide consistent solutions for larger shortest paths. It can take more than $15\times$ the time to converge compared to FiNS. Neurosymbolic diffusion models do provide consistently high accuracy, but converge significantly slower than FiNS. In particular, NeSyDMs do not perform well at comparable training times (Table 3). Additionally, NeSyDMs have no immediate way of extrapolating what they have learned on smaller grids to larger ones due to their use of expressive diffusion models (Section **(RQ3)**). Other approximations either do not scale (Scallop [58]) or assume that a MAP solver is available (I-MLE [78]). PLIA$_t$ is both fast and accurate on visual sudoku, but requires a linear arithmetic encoding that is not available for Warcraft shortest path. In contrast, FiNS is generally applicable as it does not assume specific encodings or solvers.

5.3 (RQ2) Consistency

Consistency is evaluated by the effective sample size (ESS) of 1000-sample SNIS approximations of the constraint-corrected variational distribution $q^{\mu}_{\boldsymbol{\theta}}(\omega \mid \varphi) \propto \mu(\varphi, \omega) q_{\boldsymbol{\theta}}(\omega)$ and the true posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi) \propto \mu(\varphi, \omega) p_{\boldsymbol{\lambda}}(\omega)$ with $p_{\boldsymbol{\lambda}}(\omega)$ the approximately uniform prior of an untrained neural network (Table 4). Across all settings, the ESS against $q^{\mu}_{\boldsymbol{\theta}}(\omega \mid \varphi)$ exceeds 685, which confirms that $q_{\boldsymbol{\theta}}$ places most of its mass on constraint-satisfying interpretations . The ESS against the true posterior remains high for visual sudoku at $865.95^{+3.50}_{-6.78}$ and $590.37^{+38.71}_{-110.08}$ for 4×4 and 9×9, but collapses

Table 3: Median accuracy with 25th and 75th quantiles at comparable training times (minutes) for Warcraft shortest path planning taken across 5 independent training runs. A-NeSI is left out as it does not provide meaningful results within the considered time.

	12×12		30×30	
Method	**Accuracy (↑)**	**Time (↓)**	**Accuracy (↑)**	**Time (↓)**
I-MLE	$95.30^{+0.40}_{-0.00}$	10.86 min	$95.20^{+0.10}_{-0.00}$	42.94 min
NeSyDM	$9.60^{+2.80}_{-1.60}$	11.20 min	$4.50^{+7.60}_{-3.90}$	52.34 min
FiNS	$98.80^{+0.20}_{-0.30}$	13.69 min	$98.60^{+0.10}_{-0.40}$	37.52 min

Table 4: Median ESS over 1000 samples across 5 independent training runs for the two SNIS targets.

	Visual sudoku		Warcraft	
Target	4×4	9×9	12×12	30×30
$q^{\mu}_{\boldsymbol{\theta}}(\omega \mid \varphi)$	$931.05^{+13.33}_{-19.33}$	$851.52^{+\ 32.50}_{-\ 56.96}$	$952.22^{+9.12}_{-6.31}$	$685.04^{+5.36}_{-5.69}$
$p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$	$865.95^{+\ 3.50}_{-\ 6.78}$	$590.37^{+\ 38.71}_{-110.08}$	$2.25^{+0.12}_{-0.09}$	$1.54^{+0.11}_{-0.10}$

for Warcraft at $2.25^{+0.12}_{-0.09}$ and $1.54^{+0.11}_{-0.10}$ for 12×12 and 30×30, where the deterministic Dijkstra step concentrates mass on a few satisfying paths.

5.4 (RQ3) Generalisation

Relational generalisation is tested by learning a solution on a 12×12 Warcraft map and testing it on a larger map. Our exchangeable variational approximations (Theorem 3.1) directly translate from smaller grids to larger grids by adding as many copies of the same conditionally i.i.d. distribution as there are additional cells. This translation is done for 5 models learned on 12×12 maps and testing them on 30×30 yields a median accuracy of $99.80^{+0.20}_{-0.80}$ (FiNS (Gen.) in Table 2) . This result confirms that successful generalisation is possible with our approach. Interestingly, it also indicates that learning exchangeable distributions on smaller examples and generalising to larger examples results in better performance than learning on larger examples directly. Note that no other method can generalise to larger grids from smaller ones, because they are tied to the size of the grid.

6 Conclusion, limitations and future work

The present work argued that a Bayesian perspective on learning and reasoning under uncertainty is beneficial for the development of neurosymbolic AI. It demonstrated that the adaptation of general-purpose approximate Bayesian inference methods to neurosymbolic problems is feasible and yields results competitive to more specialised, state-of-the-art neurosymbolic AI methods. The proposed adaptation of variational inference integrates two key properties of neurosymbolic models, consistency and relational generalisation, through the use of statistical estimation methods and by exploiting representation theorems for exchangeable random variables. Doing so yields a principled approach to approximate distributions conditioned on symbolic knowledge. A principled approach that, we hope, initialises interest in leveraging deep statistical models for neurosymbolic AI and, conversely, integrating more symbolic, programmable properties into statistical models.

Limitations and future work. Some of our choices induce inherent limitations. First, we focused on two properties (Section 3.2) that we consider essential for neurosymbolic approximations, but we do not include any form of interpretability or explainability that is often associated with symbolic systems. Second, the choice to use SNIS (Section 3.2.1) limits the type of guarantees we can provide for consistency and does not resolve mode collapse (Section 5.3). Analysing the trade-off between the computational cost and guarantees of more advanced techniques like MCMC or SMC for neurosymbolic problems is an interesting direction for future work. Third, we have only hinted at a broader operationalisation of generalisation beyond infinite exchangeability (Appendix D). A study of relational generalisation could lead to a recipe for programming generalising distributions that can be used in approximate Bayesian inference. Of particular interest to programmable approximations that generalise are their equally programmable forms of inference and learning aligning with the ideas of universal probabilistic programming [41].

Acknowledgements

This research received funding from the Flemish Government under the "Flanders Artificial Intelligence Research program". It was also supported by the Internal Funds KU Leuven (project PDMT2/25/057) and is currently supported by a junior postdoctoral fellowship of the Research Foundation - Flanders (FWO) under grant agreement No. 12A6027N. The authors would also like to sincerely thank Isidoro Tamassia, Jaron Maene, Luc De Raedt and Robin Manhaeve for their valuable feedback and discussions that helped improve the exposition of the paper.

References

[1] Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Semantic probabilistic layers for neuro-symbolic learning. *Advances in Neural Information Processing Systems*, 35:29944–29959, 2022.

[2] Kareem Ahmed, Zhe Zeng, Mathias Niepert, and Guy Van den Broeck. Simple: A gradient estimator for k-subset sampling. In *International Conference on Learning Representations*, 2023.

[3] Antoine Amarilli, Florent Capelli, Mikaël Monet, and Pierre Senellart. Connecting knowledge compilation classes width parameters. *Theory of Computing Systems*, 64(5):861–914, 2020.

[4] Víctor Elvira Arregui, Luca Martino, David Luengo García, and Mónica F Bugallo. Generalized multiple importance sampling. *Statistical science*, 34(1):129–155, 2019.

[5] Eriq Augustine, Connor Pryor, Charles Dickens, Jay Pujara, William Wang, and Lise Getoor. Visual sudoku puzzle classification: A suite of collective neuro-symbolic tasks. In *International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*, 2022.

[6] Jon Barwise. An introduction to first-order logic. In *Studies in Logic and the Foundations of Mathematics*, volume 90, pages 5–46. Elsevier, 1977.

[7] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

[8] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 2019.

[9] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[10] Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(90):1–61, 2020.

[11] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

[12] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 2017.

[13] George Casella and Christian P Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 1996.

[14] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

[15] Florian Chen, Felix Weitkämper, and Sagar Malhotra. Understanding domain-size generalization in markov logic networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 297–314. Springer, 2024.

[16] Yuqiao Chen, Yibo Yang, Sriraam Natarajan, and Nicholas Ruozzi. Lifted hybrid variational inference. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4237–4244, 2021.

[17] Jaesik Choi and Eyal Amir. Lifted relational variational inference. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 196–206, 2012.

[18] Marco F Cusumano-Towner, Feras A Saad, Alexander K Lew, and Vikash K Mansinghka. Gen: a general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th acm sigplan conference on programming language design and implementation*, pages 221–236, 2019.

[19] Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

[20] Bruno de Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, volume 7, pages 1–68, 1937.

[21] Bruno de Finetti. Sulla proseguibilità di processi aleatori scambiabili, 1969.

[22] Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical relational artificial intelligence: Logic, probability, and computation*. Morgan & Claypool Publishers, 2016.

[23] Lennert De Smet and Luc De Raedt. Defining neurosymbolic AI. *arXiv preprint arXiv:2507.11127*, 2025.

[24] Lennert De Smet and Pedro Zuidberg Dos Martires. A fast convoluted story: Scaling probabilistic inference for integer arithmetics. *Advances in Neural Information Processing Systems*, 37:102456–102478, 2024.

[25] Lennert De Smet, Pedro Zuidberg Dos Martires, Robin Manhaeve, Giuseppe Marra, Angelika Kimmig, and Luc De Raedt. Neural probabilistic logic programming in discrete-continuous domains. In *Uncertainty in Artificial Intelligence*, pages 529–538. PMLR, 2023.

[26] Lennert De Smet, Emanuele Sansone, and Pedro Zuidberg Dos Martires. Differentiable sampling of categorical distributions using the catlog-derivative trick. *Advances in Neural Information Processing Systems*, 36:30416–30428, 2023.

[27] Rina Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Learning in graphical models*, pages 75–104. Springer, 1998.

[28] Persi Diaconis. Finite forms of de finetti's theorem on exchangeability. *Synthese*, 36(2): 271–281, 1977.

[29] Persi Diaconis and David Freedman. Finite exchangeable sequences. *The Annals of Probability*, pages 745–764, 1980.

[30] Persi Diaconis and David Freedman. A dozen de finetti-style results in search of a theory. In *Annales de l'IHP Probabilités et statistiques*, volume 23, pages 397–423, 1987.

[31] EW Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 (1):269–271, 1959.

[32] Pedro Zuidberg Dos Martires. Probabilistic neural circuits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17280–17289, 2024.

[33] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[34] Karl-Filip Faxén. A static semantics for haskell. *Journal of functional programming*, 12(4-5): 295–357, 2002.

[35] Tor Erlend Fjelde, Kai Xu, David Widmann, Mohamed Tarek, Cameron Pfiffer, Martin Trapp, Seth D Axen, Xianda Sun, Markus Hauru, Penelope Yong, et al. Turing.jl: a general-purpose probabilistic programming language. *ACM Transactions on Probabilistic Machine Learning*, 1(3):1–48, 2025.

[36] Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306–312, 2013.

[37] Artur d'Avila Garcez and Luis C Lamb. Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.

[38] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, pages 721–741, 1984.

[39] John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

[40] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992.

[41] Noah D Goodman, Vikash K Mansinghka, Daniel Roy, Keith Bonawitz, and Joshua B Tenenbaum. Church: a language for generative models. In *Uncertainty in Artificial Intelligence*, pages 220–229. PMLR, 2008.

[42] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

[43] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

[44] Petr Hájek, Lluís Godo, and Francesc Esteva. A complete many-valued logic with product-conjunction. *Archive for mathematical logic*, 35(3):191–208, 1996.

[45] Edwin Hewitt and Leonard J Savage. Symmetric measures on cartesian products. *Transactions of the American Mathematical Society*, 80(2):470–501, 1955.

[46] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[47] Manfred Jaeger and Oliver Schulte. A complete characterization of projectivity for statistical relational models. In *29th International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4283–4290. International Joint Conferences on Artificial Intelligence, 2020.

[48] Ghassen Jerfel, Serena Wang, Clara Wong-Fannjiang, Katherine A Heller, Yian Ma, and Michael I Jordan. Variational refinement for importance sampling using the forward kullback-leibler divergence. In *Uncertainty in Artificial Intelligence*, pages 1819–1829. PMLR, 2021.

[49] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[50] David G Kendall. On finite and infinite sequences of exchangeable events. *Studia Scientiarum Mathematicarum Hungarica*, 2:319–327, 1967.

[51] Andrey Nikolaevich Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer, Berlin, 1933.

[52] Augustine Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep*, 348:14, 1992.

[53] Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! *ICLR Deep RL Meets Structured Prediction Workshop*, 2019.

[54] Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International conference on machine learning*, pages 3499–3508. PMLR, 2019.

[55] Tuan Anh Le, Atilim Gunes Baydin, and Frank Wood. Inference compilation and universal probabilistic programming. In *Artificial Intelligence and Statistics*, pages 1338–1348. PMLR, 2017.

[56] John Leland and YooJung Choi. On the hardness of approximating distributions with tractable probabilistic models. *Advances in Neural Information Processing Systems*, 38:42977–43000, 2026.

[57] Yichong Leng, Zehua Chen, Junliang Guo, Haohe Liu, Jiawei Chen, Xu Tan, Danilo Mandic, Lei He, Xiangyang Li, Tao Qin, et al. Binauralgrad: A two-stage conditional diffusion probabilistic model for binaural audio synthesis. *Advances in Neural Information Processing Systems*, 35:23689–23700, 2022.

[58] Ziyang Li, Jiani Huang, and Mayur Naik. Scallop: A language for neurosymbolic programming. *Proceedings of the ACM on Programming Languages*, 7(PLDI):1463–1487, 2023.

[59] Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

[60] Lorenzo Loconte, Aleksanteri M Sladek, Stefan Mengel, Martin Trapp, Arno Solin, Nicolas Gillis, and Antonio Vergari. Subtractive mixture models via squaring: Representation and learning. In *International Conference on Learning Representations*, 2024.

[61] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[62] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[63] Jaron Maene, Vincent Derkinderen, and Luc De Raedt. On the hardness of probabilistic neurosymbolic learning. In *International Conference on Machine Learning*, pages 34203–34218, 2024.

[64] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems*, 31, 2018.

[65] Robin Manhaeve, Giuseppe Marra, and Luc De Raedt. Approximate inference for neural probabilistic logic programming. In *KR*, pages 475–486, 2021.

[66] Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. *Advances in Neural Information Processing Systems*, 36:72507–72539, 2023.

[67] Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Antonio Vergari, Andrea Passerini, and Stefano Teso. Bears make neuro-symbolic models aware of their reasoning shortcuts. In *Uncertainty in Artificial Intelligence*, pages 2399–2433. PMLR, 2024.

[68] Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, 328:104062, 2024.

[69] Peter McCullagh. What is a statistical model? *The Annals of Statistics*, 30(5):1225–1310, 2002.

[70] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[71] Eleonora Misino, Giuseppe Marra, and Emanuele Sansone. Vael: Bridging variational autoencoders and probabilistic logic programming. *Advances in Neural Information Processing Systems*, 35:4667–4679, 2022.

[72] Richard Montague. The proper treatment of quantification in ordinary english. In *Approaches to natural language: Proceedings of the 1970 Stanford workshop on grammar and semantics*, pages 221–242. Springer, 1973.

[73] Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *International Conference on Automated Deduction*, pages 625–635. Springer, 2021.

[74] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

[75] Radford M Neal. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, pages 47–95, 2011.

[76] Mathias Niepert. Markov chains on orbits of permutation groups. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 624–633, 2012.

[77] Mathias Niepert and Guy Van den Broeck. Tractability through exchangeability: A new perspective on efficient probabilistic inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[78] Mathias Niepert, Pasquale Minervini, and Luca Franceschi. Implicit mle: backpropagating through discrete exponential family distributions. *Advances in Neural Information Processing Systems*, 34:14567–14579, 2021.

[79] Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

[80] Art B Owen. Monte carlo theory, methods and examples, 2013.

[81] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Learning with algorithmic supervision via continuous relaxations. *NeurIPS*, 2021.

[82] Carsten Peterson and James R Anderson. A mean field theory learning algorithm for neural networks. *Complex systems*, 1(5), 1987.

[83] Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. Differentiation of blackbox combinatorial solvers. In *International conference on learning representations*, 2019.

[84] David Poole. First-order probabilistic inference. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 985–991, 2003.

[85] Maxim Rabinovich, Aaditya Ramdas, Michael I Jordan, and Martin J Wainwright. Function-specific mixing times and concentration away from equilibrium. *Bayesian Analysis*, 15(2): 505–532, 2020.

[86] C Radhakrishna Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 1945.

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

[88] Dan Roth. On the hardness of approximate reasoning. *Artificial intelligence*, 82(1-2):273–302, 1996.

[89] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International conference on machine learning*, pages 1218–1226. PMLR, 2015.

[90] Laurent Schwartz. Théorie des distributions à valeurs vectorielles. i. In *Annales de l'institut Fourier*, volume 7, pages 1–141, 1957.

[91] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[92] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[93] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep probabilistic programming. In *ICLR*, 2017.

[94] Grigori S Tseitin. On the complexity of derivation in propositional calculus. In *Automation of reasoning: 2: Classical papers on computational logic 1967–1970*, pages 466–483. Springer, 1983.

[95] Guy Van den Broeck, Kristian Kersting, Sriraam Natarajan, and David Poole. *An introduction to lifted probabilistic inference*. MIT Press, 2021.

[96] Emile van Krieken, Thiviyan Thanapalasingam, Jakub Tomczak, Frank Van Harmelen, and Annette Ten Teije. A-nesi: A scalable approximate method for probabilistic neurosymbolic inference. *Advances in Neural Information Processing Systems*, 36:24586–24609, 2023.

[97] Emile Van Krieken, Pasquale Minervini, Edoardo Ponti, and Antonio Vergari. On the independence assumption in neurosymbolic learning. In *International Conference on Machine Learning*, pages 49078–49097. PMLR, 2024.

[98] Emile van Krieken, Pasquale Minervini, Edoardo Ponti, and Antonio Vergari. Neurosymbolic diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[99] Guido Van Rossum, Fred L Drake, et al. *Python reference manual*, volume 111. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[100] Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

[101] Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. *Advances in Neural Information Processing Systems*, 34:13189–13201, 2021.

[102] Benjie Wang, Denis Mauá, Guy Van den Broeck, and YooJung Choi. A compositional atlas for algebraic circuits. *Advances in Neural Information Processing Systems*, 37:141318–141355, 2024.

[103] Felix Weitkämper. Projective families of distributions revisited. *International Journal of Approximate Reasoning*, 162:109031, 2023.

[104] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, pages 5502–5511. PMLR, 2018.

[105] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 2008–2026, 2018.

[106] Nevin Lianwen Zhang and David Poole. A simple approach to bayesian network computations. In *Proceedings of the Biennial Conference-Canadian Society for Computational Studies of Intelligence*, pages 171–178. Canadian Information Processing Society, 1994.

A An ELBO for learnable priors and generative neurosymbolic models

Neurosymbolic problems observe both symbolic sentences and subsymbolic data. These subsymbolic data have so far been treated as fixed inputs such that the desired neurosymbolic posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$ is only conditional on a sentence φ. However, the subsymbolic data can be modelled as random variables $\boldsymbol{X}$ to yield a neurosymbolic posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi, \boldsymbol{X})$ explicitly conditioned on both a symbolic sentence φ and the subsymbolic data $\boldsymbol{X}$. The addition of subsymbolic observations requires a corresponding likelihood, which will often be a learnable generative model. Consequently, the task of posterior computation now involves a generative component $p(X \mid \omega, \boldsymbol{\lambda})$

$$p^{\mu}(\omega, \boldsymbol{\lambda} \mid \varphi, \boldsymbol{X}) = \frac{\mu(\varphi, \omega) \cdot p(X \mid \omega, \boldsymbol{\lambda}) \cdot p(\omega, \boldsymbol{\lambda})}{p(\varphi, X)}. \tag{9}$$

The parameters $\boldsymbol{\lambda}$ are now explicitly included in the prior and posterior to obtain a valid ELBO when optimising the additional generative component. If they would only appear as optimisable parameters in $p_{\boldsymbol{\lambda}}(\boldsymbol{X} \mid \omega)$, then the evidence $p_{\boldsymbol{\lambda}}(\varphi, \boldsymbol{X})$ would depend on the parameters. This dependency would then prevent the derivation of an ELBO where the observational evidence is no longer part of the equation. Adding the parameters $\boldsymbol{\lambda}$ instead yields an extended negated ELBO

$$\mathsf{KL}\left(q_{\boldsymbol{\theta}}(\omega, \boldsymbol{\lambda}) \,||\, p(\omega, \boldsymbol{\lambda})\right) - \mathbb{E}_{q_{\boldsymbol{\theta}}(\omega, \boldsymbol{\lambda})}\left[\log \mu(\varphi, \omega)\right] - \mathbb{E}_{q_{\boldsymbol{\theta}}(\omega, \boldsymbol{\lambda})}\left[\log p(\boldsymbol{X} \mid \omega, \boldsymbol{\lambda})\right], \tag{10}$$

where the additional term now measures the generative likelihood. This generative likelihood together with the choice of variational family $q(\omega, \boldsymbol{\lambda})$ allows for different generative models to be part of the overall neurosymbolic model. For instance, a neural encoder as amortised variational distribution $q_{\boldsymbol{\theta}}(\omega, \boldsymbol{\lambda} \mid \boldsymbol{X})$ together with a neural decoder $p(X \mid \omega, \boldsymbol{\lambda})$ results in a variational autoencoder neurosymbolic model [71, 25]. Alternatively, implementing $q_{\boldsymbol{\theta}}(\omega, \boldsymbol{\lambda} \mid \boldsymbol{X})$ and $p(\boldsymbol{X} \mid \omega, \boldsymbol{\lambda})$ as forward and reverse diffusion processes would yield a diffusion-based neurosymbolic model [98].

Variational inference with an explicit generative component targets a more informative posterior. The posterior being conditioned on both the symbolic sentence and the subsymbolic data results in an ELBO that lower bounds the joint log probability $\log p(\varphi, \boldsymbol{X})$ instead of $\log p(\varphi)$. This difference manifests in a variational approximation that aims to simultaneously satisfy the symbolic sentence φ and be able to reproduce the subsymbolic data $\boldsymbol{X}$. Additionally, posterior inference immediately comes with a generative model that can be used to decode the internal symbolic states.

B Asymptotic relaxations of logical languages with Boolean semantics

The inductive semantics of logical languages with Boolean semantics facilitates an inductive relaxation. The general idea is to simply replace the usual 0 value of atomic statements that evaluate to false with an increasingly small ε between 0 and 0.5. Moreover, this value ε can be treated as a random variable to introduce stochastic behaviour into the optimisation process. This stochasticity can accelerate convergence by avoiding local optima, but is not used further in our results. Asymptotic convergence to the exact semantics is guaranteed by annealing ε towards 0 over the course of training.

Assume without loss of generality that the sentence φ is in conjunctive normal form (CNF), i.e.

$$\varphi \leftrightarrow \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} l_{ij}, \tag{11}$$

where each l_{ij} is a literal. Further, let the Boolean semantic values be 0 or 1 representing $\bot$ and $\top$, respectively. Then, the semantic value of φ in an interpretation ω is given by

$$\mu(\varphi, \omega) = \bigotimes_{i=1}^{n} \bigoplus_{j=1}^{m_i} \mu(l_{ij}, \omega), \tag{12}$$

where $\oplus$ and $\otimes$ are algebraic operations that implement Boolean semantics on 0 and 1 values. For instance, the probabilistic sum $a \oplus b = a + b - ab$ and product $a \otimes b = ab$. More generally, any fuzzy algebraic operations $\oplus$ and $\otimes$ on the unit interval can be used, since they coincide with Boolean semantics for 0 and 1 values. This inductive algebraic semantics can be relaxed by simply replacing 0 with $0 < \varepsilon < 0.5$ for atomic statements that evaluate to false and use the same $\oplus$ and $\otimes$ for the logical connectives. That is, the relaxation

$$\mu_r(\varphi, \omega) = \bigotimes_{i=1}^{n} \bigoplus_{j=1}^{m_i} \mu_r(l_{ij}, \omega) \tag{13}$$

with

$$\mu_r(l_{ij}, \omega) = \begin{cases} \varepsilon & \text{if } \mu(l_{ij}, \omega) = 0 \\ 1 - \varepsilon & \text{if } \mu(l_{ij}, \omega) = 1 \end{cases}. \tag{14}$$

It is immediate to see that the limit of μ_r for $\varepsilon \to 0$ is equal to μ.

C Effective sample size for approximate neurosymbolic posteriors

The effective sample size (ESS) is a metric used in statistics to measure how efficient the estimation of an expectation $\mathbb{E}_{p(\boldsymbol{X})}[f(\boldsymbol{X})]$ is based on a proposal distribution $q(\boldsymbol{X})$. Intuitively, it measures how many samples of p are obtained when taking a certain number of samples from q. More formally, importance sampling estimation of $\mathbb{E}_{p(\boldsymbol{X})}[f(\boldsymbol{X})]$ utilises the importance weights $\tilde{w}_n = \frac{p(\boldsymbol{x}^{(n)})}{q(\boldsymbol{x}^{(n)})}$ of samples $\boldsymbol{x}^{(n)} \sim q(\boldsymbol{X})$ that are then used to define the ESS as

$$\text{ESS} = \frac{1}{\sum_{n=1}^{N} w_n^2}, \tag{15}$$

where now w_n are the normalised weights

$$w_n = \frac{\tilde{w}_n}{\sum_{n=1}^{N} \tilde{w}_n}. \tag{16}$$

In the context of FiNS' SNIS approximations of the neurosymbolic posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$, we fill in the unnormalised weights

$$\tilde{w}_n = \frac{\mu\left(\varphi, \omega^{(n)}\right) p_{\boldsymbol{\lambda}}\left(\omega^{(n)}\right)}{q_{\boldsymbol{\theta}}\left(\omega^{(n)}\right)}, \tag{17}$$

in case one wants to be consistent with both the prior $p_{\boldsymbol{\lambda}}$ and the sentence φ. In practical applications, consistency with background knowledge φ usually has priority. Consequently, one can instead target the distribution $q^{\mu}_{\boldsymbol{\theta}}(\omega \mid \varphi) \propto \mu(\varphi, \omega) q_{\boldsymbol{\theta}}(\omega)$ with unnormalised weights being equal to only the semantics $\tilde{w}_n = \mu\left(\varphi, \omega^{(n)}\right)$.

D Finite generalisation through exchangeability and orthogonal invariance

Demanding generalisation to an arbitrary number of variables is a strong requirement and perhaps it is sufficient to generalise up to an upper limit n of variables from observing only at most k variables. In that case, representation theorems for finite exchangeable sequences of random variables are of interest. A long line of research dating back to Kendall [50] and continued over the subsequent decades [21, 28, 29] clarified the landscape of finite generalisation through exchangeability and its generalisation of orthogonal invariance [30]. If the variables of the sequence have a finite domain, then Diaconis and Freedman [29] give a precise characterisation based on previous works [50, 21, 28]; any finite exchangeable sequence of n variables with domain size d is governed by a "mixture of urns", i.e. a mixture of draws without replacement from a collection of n objects with d types.

Theorem D.1 (Finite exchangeability representation theorem [29])**.** If $(X_1, \ldots, X_n)$ is a finite sequence of exchangeable random variables taking values in a finite alphabet $\mathcal{A} = \{1, 2, \ldots, c\}$, then there exists a probability distribution $p(\boldsymbol{u})$ over the set of all possible urn compositions $\mathcal{U}_n = \{\boldsymbol{u} \in \mathbb{N}^c \mid \sum_{j=1}^{c} u_j = n\}$ such that the joint probability of a subsequence $\boldsymbol{x} = (x_1, \ldots, x_k)$ for any integer $k \leq n$ is

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathcal{U}_n} H_{n,k,\boldsymbol{u}}(\boldsymbol{x}) p(\boldsymbol{u}),$$

where $H_{n,k,\boldsymbol{u}}(\boldsymbol{x})$ is the probability of drawing the ordered sequence $\boldsymbol{x}$ without replacement from an urn of size n with composition $\boldsymbol{u}$. That is, it is the multivariate hypergeometric distribution

$$H_{n,k,\boldsymbol{u}}(\boldsymbol{x}) = \binom{n}{k}^{-1} \binom{k}{\boldsymbol{c}(\boldsymbol{x})}^{-1} \prod_{j=1}^{c} \binom{u_j}{c_j(\boldsymbol{x})},$$

where $c_j(\boldsymbol{x})$ is the number of times category j appears in the subsequence $\boldsymbol{x}$ and $\boldsymbol{c}(\boldsymbol{x}) = (c_1(\boldsymbol{x}), \ldots, c_c(\boldsymbol{x}))$ is the vector of these counts. $H_{n,k,\boldsymbol{u}}(\boldsymbol{x}) = 0$ if any required count exceeds the urn inventory ($c_j(\boldsymbol{x}) > u_j$).

Note how the representation theorem explicitly separates the sequence distribution into a distribution $p(\boldsymbol{u})$ over urn compositions and a conditional sampling mechanism $H_{n,k,\boldsymbol{u}}$ that draws k elements from an urn of size n *without replacement*. In a deep learning context, this sampling mechanism

H can be directly parametrised using recent advances in differentiable top-k sampling [54, 2]. By coupling these sampling algorithms with a neural network that parametrises the distribution $p(\boldsymbol{u})$, a fully differentiable architecture for finite exchangeable distributions is obtained. For instance, a neural network can output the parameters of a Dirichlet distribution to parametrise $p(\boldsymbol{u})$. Interestingly, this mathematical link between discrete urn models and differentiable top-k sampling provides a theoretical foundation for applying top-k parametrisations to achieve generalisation up to n variables.

If the variable domains are infinite, then exchangeability alone does not lead to a reasonable characterisation. Instead, replacing exchangeability (Definition 3.1) with orthogonal invariance, i.e. invariance to not just permutations, but also rotations and reflections, does give a useful characterisation [30].

Definition D.1 (Orthogonally invariant random variables)**.** A sequence $(X_n)_{n\in\mathbb{N}}$ of continuous random variables is called orthogonally invariant if, for any integer $k \geq 1$ and any orthogonal matrix $Q \in \mathcal{O}(k)$, it holds that

$$p(x_1, \ldots, x_k) = p(y_1, \ldots, y_k),$$

where $(y_1, \ldots, y_k)^\top = Q(x_1, \ldots, x_k)^\top$.

If we take a finite sequence of n orthogonal invariant *continuous* variables, then its distribution is a mixture of distributions over spheres in $\mathbb{R}^n$ where the mixture parameter is the radius of the sphere.

Theorem D.2 (Finite orthogonal representation theorem [30])**.** If $(X_1, \ldots, X_n)$ is a finite sequence of $n \in \mathbb{N}$ orthogonally invariant continuous random variables, then there exists a probability measure λ on $[0, \infty)$ governing the total radius $R_n = \sqrt{\sum_{i=1}^n X_i^2}$ such that for any $k < n$ it holds that

$$p(x_1, \ldots, x_k) = \int_0^\infty Q_{n,k,r}(\boldsymbol{x}) \, \mathrm{d}\lambda(r),$$

where $Q_{n,k,r}$ is the k-dimensional marginal density of a uniform distribution on the n-dimensional sphere of radius r. This density is

$$Q_{n,k,r}(\boldsymbol{x}) \propto \left(1 - \frac{\|\boldsymbol{x}\|^2}{r^2}\right)^{\frac{n-k-2}{2}},$$

for $\|\boldsymbol{x}\|^2 < r^2$ and $Q_{n,k,r}(\boldsymbol{x}) = 0$ for $\|\boldsymbol{x}\|^2 \geq r^2$.

A practical neural parametrisation of an orthogonal invariant continuous distribution over n variables X_i, i.e. a mixture of distributions on n-dimensional spheres, is straightforward to implement. First, use a neural network to parametrise a distribution over radii $r \in [0, \infty)$. For instance, by parametrising the rate parameter λ of an exponential distribution. Second, drawing samples from the joint orthogonal invariant $p(X_1, \ldots, X_n)$, requires drawing samples from uniform distributions over n-dimensional spheres. Such spherical samples can be obtained by simply sampling $\boldsymbol{Z}^{(i)} \sim \mathcal{N}(0, \boldsymbol{I}_n)$ and normalizing to the unit sphere, i.e. setting

$$\boldsymbol{X}^{(i)} = \frac{\boldsymbol{Z}^{(i)}}{\sqrt{\sum_{j=1}^n (Z_j^{(i)})^2}}. \tag{18}$$

These samples are uniformly distributed over the n-dimensional unit sphere because standard normal Gaussians are themselves spherically symmetric. To now get samples uniformly distributed over an n-dimensional sphere of radius r it suffices to multiply $\boldsymbol{X}^{(i)}$ by r. Third, samples from only $k < n$ observed variables can be obtained by discarding the last $n - k$ coordinates of $\boldsymbol{X}^{(i)}$. In other words, if we observe k variables, then the samples $\boldsymbol{X}^{(i)}_{1:k}$ can be used for learning. Importantly, the above procedure is fully reparametrisable as we only multiply with the sphere radius r sampled from a reparametrisable distribution.

E Details on experimental setup and additional results

All methods used the AdamW [61] optimiser with a learning rate of 0.001 unless specifically stated otherwise. All reported accuracies and timings are based on an optimal model selected with the highest accuracy on a hold-out validation set. Examples from both datasets can be seen in Figure 2. Finally, FiNS and all other baselines had access to the same computational resources; two AMD



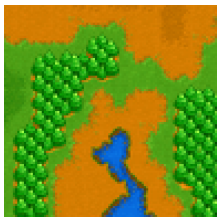


Figure 2: Left: an example visual sudoku that should be predicted as valid. Right: an example warcraft map on which the shortest path from top left to bottom right moves around the trees.

EPYC 9334 32-Core Processors with 256 GiB of RAM and two Nvidia L40S GPUs with 48 GiB of VRAM each. Overall compute time for single training runs are provided in Tables 1 and 2. While we did not conduct large-scale hyperparameter optimisation, we estimate that our exploratory experiments and running of multiple seeds took approximately 250 GPU hours, not taking into account running of baselines. Reproducing only FiNS' results on similar hardware should take around 25 GPU hours.

E.1 Visual sudoku

Baseline setup and hyperparameters. The three baselines were run with optimised parameters using their original architectures. Scallop and PLIA$_t$ used their respective optimal parameters as identified in previous works [24] using a grid search. Specifically, Scallop ran for 100 epochs using $k = 6$ for the 4×4 sudoku puzzles. It did not manage to compile successfully on 9×9 puzzles even with $k = 1$. PLIA$_t$ ran for 150 and 1000 epochs on 4×4 and 9×9 puzzles. A-NeSI was run with its optimal parameters as reported in the original publication [96].

FiNS setup and hyperparameters. FiNS aims to exploit expressive variational distributions $q_{\boldsymbol{\theta}}$ to approximate the neurosymbolic posterior $p^{\mu}_{\boldsymbol{\lambda}}(\omega \mid \varphi)$ where $p_{\boldsymbol{\lambda}}(\omega)$ is close to uniform as it is an untrained neural network. Here, the variational distribution $q_{\boldsymbol{\theta}}$ is a Graph Attention Network (GAT) [11]. The GAT-GNN is applied on a grid-shaped graph where the nodes represent cells of the sudoku puzzle with edges between adjacent cells along the cardinal directions. The embedding of each node is the LeNet convolutional embedding and the attention part of the GAT-GNN consists of 3 (4×4) or 9 (9×9) layers of width 128. Each processed node embedding is then classified by a LeNet head into one of the G possible digits to parametrise a mean-field approximation that factorises over grid cells. The row, column and box constraints are implemented using the same binary encoding of PLIA$_t$, which are then conjoined together to form the complete sudoku constraint. In terms of hyperparameters, FiNS ran consistently well with the same learning rate of 0.001 for AdamW with a weight decay of 0.0001. While a range of different weights for the KL divergence term in Equation 3 were tested, the default value of 1.0 proved reliable for both sizes of visual sudoku puzzles. A maximum training time of 100 epochs and 500 epochs was given for 4×4 and 9×9 puzzles within which the model with highest validation accuracy was selected. Training utilised 1000 (4×4) and 10000 (9×9) samples for Monte Carlo and RLOO [53] estimates, and a batch size of 5. Finally, norm-based gradient clipping was applied with a threshold of 0.30 and a fixed value of $\varepsilon = 10^{-12}$ was used in our relaxations.

E.2 Warcraft path planning

Baseline setup and hyperparameters. Each of the baselines are provided with their own implementation as well as optimised hyperparameters. We refer to the respective works for more details as we directly used these for our comparisons.

FiNS setup. The variational approximation consisted of a shallow probabilistic neural circuit [32] constructed as follows. A small MLP with sequential hidden sizes of 32 and 16 was used to model 5 mixture components $p_i(c)$ predicting the probabilities of a cell c (8 width by 8 height by 3 colour channels) being one of the 5 possible classes. Another such MLP then also predicted the 5 mixture weights $\alpha_i(c)$ from the cell c. The mixture distribution $p(c) = \sum_{i=1}^{5} \alpha_i(c) \cdot p_i(c)$ was then applied on each cell of an input map independently to predict the total distribution over cell types of the

entire map. Next, this distribution is sampled 20 times and a GPU-based implementation of Dijkstra's algorithm [31] is applied to each of the 20 samples to result in 20 sampled paths in the form of a $G \times G$ matrix filled with zeroes for tiles where the paths do not go and ones otherwise. These sampled paths are then compared to the true path by comparing each tile value of the samples with the tile value of the true shortest path. The final constraint consists of a conjunction across all tiles.

FiNS hyperparameters. Our setup uses the AdamW optimiser with a learning rate of 0.01 for both map sizes together with a weight decay of 0.005 (12×12) and 0.0005 (30×30). Training was given 5 epochs with a batch size of 50 on both map sizes. The best model within this training time was chosen based on a held-out validation set, as per usual. The norm-based gradient clipping was again applied with a threshold of 0.30 and a fixed value of $\varepsilon = 10^{-12}$ was used for the relaxed semantics on 12×12 maps. For 30×30 maps, we use $\varepsilon = 10^{-36}$, where ε was selected on the validation set from $\{10^{-12}, 10^{-24}, 10^{-36}\}$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our first contribution **(C1)** is on providing a variational inference approximation for neurosymbolic inference. This contribution starts by highlighting the challenges of optimising the variational objective corresponding to a neurosymbolic problem (Section 3.1). These challenges are resolved, including an in-depth discussion on how to mimick typically symbolic properties with test-time estimation (Section 3.2.1) and an exchangeable variational distribution (Section 3.2.2). The latter exchangeable distributions form the claimed theoretical framework of contribution **(C2)**. The claimed empirical success of contribution **(C3)** is demonstrated by providing a good trade-off between performance and time in challenges neurosymbolic problems (Section 5) compared to the state of the art (Section 4).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Three different axes of limitations and failure modes are discussed in Section 6, as well as with possible directions for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We do not make any theoretical claims that require a formal proof. Our discussion aims to open the door to using approximate Bayesian inference for neurosymbolic problems and shows initial success empirically.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experimental details are provided in Section E to allow direct reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code will be made public by publication time with open access at `https://github.com/ML-KULeuven/fins`. It will be provided with clear instructions and a direct reproducible environment. All data is already publicly available [83, 5].

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All details are provided in the dedicated Section E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: As we can not assume normally distributed results, we use medians together with the 25^{th} and 75^{th} quantiles taken from at least five independent runs with different seeds to indicate variability on all our results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The specific compute resources used for all our experiments and comparisons are provided in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We see no reason for this paper to break the NeurIPS Code of Ethics; datasets are publicly available and our method is a general approach to probabilistic machine learning with constraints.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no immediate negative societal impact of the proposed work. Positive impact is plausible through deployment of these methods in the context of enforcing constraint adherence, e.g. safe autonomous driving. Such positive impacts are far beyond the scope of this initial, theoretical paper.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: There is no immediate risk of misuse. Our models are generic machine learning architectures trained on data that has been available for a longer duration of time.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The datasets and machine learning models we use are properly cited in the text.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: We introduce no new assets yet. Our open-access code release will however be provided with ample documentation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development did not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.