Rule-Selection Surrogates for Tractable Neurosymbolic Inference

David Debot[1] **Giuseppe Marra**[2]

[1,2]KU Leuven, Department of Computer Science

Abstract

Probabilistic neurosymbolic models combine neural perception with symbolic reasoning, but exact inference can become costly when logical rules overlap. We introduce Rule-Selection Surrogates, a post-hoc approximation method that adds a categorical neural selector over the original rules. This transforms expensive disjunctive inference into exact inference in a tractable surrogate model, requiring only a linear-time mixture computation over rules and concepts. By adding a single always-true rule, the surrogate can represent any target Bernoulli marginal pointwise. The selector is trained by distillation from an exact teacher model, while the concept predictor and symbolic rule set remain fixed. Experiments on MNIST-Addition show substantial reductions in inference time and memory usage with low approximation error, suggesting that rule-selection surrogates offer an efficient and interpretable route to scalable approximate neurosymbolic inference.

1 INTRODUCTION

Neurosymbolic AI combines learned neural perception with explicit symbolic reasoning, allowing to incorporate expert knowledge, explain predictions, and enforce high-level constraints [Marra et al., 2024, Manhaeve et al., 2018]. These benefits are especially appealing when one wants both strong predictive performance and some degree of interpretability or control.

Many probabilistic neurosymbolic models are hard to query, as scalability is one of the main problems in neurosymbolic AI. In neurosymbolic systems like DeepProbLog [Manhaeve et al., 2018], a target variable can be written as a disjunction of conjunctive logic rules, where the probability that each individual rule fires is easy to compute, while their disjunction is expensive. This is the classical disjoint-sum problem: naively adding rule probabilities double-counts worlds satisfying multiple clauses, and avoiding this double-counting makes probabilistic inference hard [De Raedt and Kimmig, 2015, Darwiche, 2001].

In practice, exact inference is often performed by compiling to tractable representations such as arithmetic circuits [Darwiche, 2003], where marginals can be computed by a linear-time pass over the circuit, once the model has been compiled. Such approaches are typically tractable in the size of the compiled representation, even though that representation may itself still be exponential in the number of random variables in the worst case.

In this paper, we want to explore a new way of providing tractable but approximate inference. Our method is inspired by prior work on rule selection, particularly Debot et al. [2024], which introduced a categorical neural rule selector as part of a broader neurosymbolic model. Debot et al. [2024] combine gradient-based rule learning and rule selection to achieve a model that can be used in settings where no expert knowledge is available. In contrast, we aim to study neural rule selection for a different setting, specifically to define a rule-selection-based surrogate for approximate inference for an existing intractable neurosymbolic probabilistic model.

Our contributions are threefold. First, we formulate rule-selection surrogates as a post-hoc approximation strategy for existing propositional neurosymbolic models. Second, we show that adding a rule that trivially evaluates to $True$ restores enough expressive range to match the original disjunctive marginal pointwise. Third, we frame learning as post-hoc distillation from an intractable exact model to a tractable surrogate. This way, we study a tractable logical representation, its approximation behavior, and its relation to probabilistic logic.

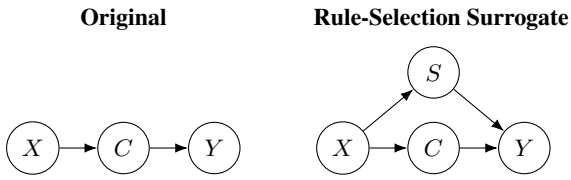

Figure 1: Probabilistic graphical model of the original neurosymbolic model and rule-selection surrogate. The target task Y is computed using symbolic concepts C that are in turn predicted from the input X, yielding a marginal $p(y \mid c)\, p(c \mid x)$. The surrogate adds a categorical rule selector S, yielding the marginal $p(y \mid c, s)\, p(s \mid x)\, p(c \mid x)$.

2 PROBLEM FORMULATION

We target the setting where performing exact inference on an existing neurosymbolic model is still feasible at training time, but is deemed too slow at test time. The input to our method is a neurosymbolic model with three components: an input X, Bernoulli random variables $C = (C_1, \ldots, C_n)$ representing human-understandable concepts, and a target variable Y defined in terms of C by a set of conjunctive rules (Figure 1). We make the standard conditional-independence assumption

$$p(C_1 = c_1, ..., C_n = c_n \mid x) = \prod_{i=1}^{n} p(C_i = c_i \mid x),$$

which is common in neurosymbolic and probabilistic frameworks. We start from an overlapping propositional rule theory in disjunctive normal form:

$$Y \Leftrightarrow \bigvee_{j=1}^{m} B_j(C), \tag{1}$$

where each concept C_i is a Bernoulli random variable and each B_j is a conjunction of literals over the concepts, which corresponds to the body of a conjunctive rule. Under the conditional-independence assumption above, the probability that a single rule body B_j fires is easy to compute. If rule s contains positive literals P_s and negative literals N_s, then

$$p(B_s = 1 \mid x) = \prod_{i \in P_s} p(C_i = 1 \mid x) \prod_{i \in N_s} p(C_i = 0 \mid x), \tag{2}$$

which has a computational cost linear in the number of concepts. Instead, the difficulty lies in computing the disjunction of Equation 1.

The output of our approach is a new surrogate probabilistic logic model for the same prediction task. This surrogate keeps the original concept predictor and logic rules, but replaces the intractable disjunctive aggregation between the different rules with a tractable approximation.

3 RULE-SELECTION SURROGATES

Our surrogate adds a categorical random variable $S \in \{1, \ldots, k\}$ to the model, which serves as a selection mechanism over rules, parameterized by a neural network (Figure 1). Here, k is the number of rules in the original neurosymbolic model. From a semantics viewpoint, each rule body B_j is conjoined with an additional atom $S = j$ that requires the categorical random variable S to take on a specific value. In this surrogate model, exact inference is scalable because the alternatives $S = j$ (and thus, the different rule bodies) are mutually exclusive, making the resulting query $p(y|x)$ computable in linear time in the number of rules and concepts. This surrogate model defines

$$p_{\text{sel}}(Y = 1 \mid x) = \sum_{j=1}^{k} p(S = j \mid x)\, p(B_j = 1 \mid x). \tag{3}$$

The selector turns an expensive disjunctive marginal into a tractable mixture computation in a surrogate model.

Proposition 1. *For k conjunctive rules over n independent concept marginals, Equation* (3) *computes $p(Y = 1 \mid x)$ in $O(kn)$ time.*

This proposition means that inference in the surrogate scales well with the size of the symbolic layer: even as the number of rules and concepts grows, the cost increases only linearly rather than combinatorially, which is precisely the kind of scalability we want from a tractable neurosymbolic approximation. The practical point of our approach is amortization: once a selector has been learned, inference in the surrogate is linear-time. Moreover, the inference computation is just a batched set of products as in Equation (2) over concepts followed by sums as in Equation (3), so it can be parallelized naturally on GPU hardware.

Note that Equation (3) is exact inference for the surrogate model. In other words, our approach performs exact inference over an approximate model [Choi et al., 2012]. This is in contrast to approaches that do approximate inference over the exact model, like e.g. sampling-based approaches. The selector can be viewed as a determinizing technique. Although two original rule bodies $B_i(C)$ and $B_j(C)$ may overlap (i.e. $p(B_i(C) = 1 \wedge B_j(C) = 1 \mid x) \neq 0$), the events $(S = i) \wedge B_i(C)$ and $(S = j) \wedge B_j(C)$ are mutually exclusive for $i \neq j$ (i.e. $p(S = i \wedge B_i(C) = 1 \wedge S = j \wedge B_j(C) = 1 \mid x) = 0$). From an arithmetic circuit perspective, the rule selection effectively introduces a deterministic sum node.

An additional advantage of the surrogate is that it still preserves much of the structure that makes the original neurosymbolic model attractive in the first place. Because we keep the original logic rules, the model retains part of the symbolic interpretability and verifiability: one can inspect the rule-body probabilities, the selector probabilities over

rules, and the resulting mixture prediction. Thus, although the selector itself is a black-box neural component, the surrogate can still be interpreted more easily than a fully black-box approximation.

A natural question is whether the surrogate model can in theory represent any probability distribution $p(Y \mid x)$. Let $p(Y = 1 \mid x) = p(\bigvee_{j=1}^{k} B_j(C) \mid x)$ denote the target marginal in the original model. Without further modification, Equation (3) is a convex combination of rule-body probabilities and therefore cannot exceed the largest one. As a solution to this, we propose adding a rule that trivially evaluates to True, written $B_0(C) = \mathit{True}$, whose probability is one.

Proposition 2. *Let $q_i(x) = p(B_i(C) = 1 \mid x)$ for $i \in \{1, \ldots, k\}$, and let $u(x) = p(Y = 1 \mid x)$. If the rule set is appended with a rule that trivially evaluates to True, i.e. $B_0(C) = \mathit{True}$, then for every x there exists a categorical distribution over $\{0, \ldots, k\}$ for the rule selector such that $p_{\mathrm{sel}}(Y = 1 \mid x) = u(x)$.*

Proof. Let $i^\star \in \arg\max_i q_i(x)$ and write $q^\star = q_{i^\star}(x)$. Since each body event is contained in the union event, $q^\star \leq u(x) \leq 1$. If $q^\star = 1$, selecting $i^\star$ with probability one is enough. Otherwise set

$$\alpha = \frac{u(x) - q^\star}{1 - q^\star},$$
$$p(S = 0 \mid x) = \alpha, \qquad p(S = i^\star \mid x) = 1 - \alpha,$$

and give zero probability to the other rules. The resulting mixture is exactly $u(x)$. □

We hypothesize that rule-selection surrogates are most suitable when the original model's predictive mass is concentrated on a small number of rule bodies for each input. The selector can then approximate the intractable disjunctive aggregation well. We believe the approach is less suitable when probability mass is broadly distributed across many heavily overlapping rules. In such cases, accurate approximation may require a strong reliance on the rule that trivially evaluates to True, reducing interpretability.

4 LEARNING THE SURROGATE

The learning problem is post-hoc. We assume an already trained intractable model with concept predictor $p(c \mid x)$ and fixed rules $B_1, \ldots, B_k$. Our method takes this model as input, adds a rule that trivially evaluates to True, and learns only the selector $p(S \mid x)$, which is parameterized by a neural network. The concept predictor and rule set remain frozen throughout.

Let $p(Y \mid x)$ denote the probability of the target under the original neurosymbolic model. The surrogate uses Equation 3 to approximate this probability, with an additional rule $B_0(C) = \mathit{True}$ which trivially evaluates to True. For a binary target variable Y, we train the selector's parameters by minimizing KL divergence to the teacher predictive distribution:

$$\sum_{x \in \mathcal{D}} \mathrm{KL}(p(Y \mid x) \,\|\, p_{\mathrm{sel}}(Y \mid x)) \,. \tag{4}$$

5 RELATED WORK

Our starting point is a neurosymbolic probabilistic logic model, where neural networks parameterize probabilistic facts (representing random variables) and a set of logic rules defines some target label [Manhaeve et al., 2018] in terms of these facts. Exact inference is typically performed through knowledge compilation [Darwiche, 2011, Choi et al., 2020], but is often still computationally expensive due to the disjoint-sum problem [De Raedt and Kimmig, 2015, Darwiche, 2001].

Some existing approaches perform approximate inference in the original model, for example by sampling the probabilistic facts or by searching for high-probability proofs with top-k or A*-like methods [Li et al., 2023, Manhaeve et al., 2021]. In contrast, we perform exact inference in an approximate surrogate model. Another related approach is A-NESI [van Krieken et al., 2023], which trains neural models to approximate probabilistic neurosymbolic inference. In contrast, our method does not learn a generic inference network, but keeps the original rules explicit and introduces a categorical selector that makes exact inference tractable.

Our surrogate perspective is also related to distillation [Hinton et al., 2015], but rather than imitating hard labels, we imitate the predictive distribution of a slow symbolic teacher. It is also close in spirit to amortized variational inference [Kingma and Welling, 2013], in the broad sense that a cheap parametric map is trained to approximate an expensive probabilistic computation. The difference from a generic black-box student is that we keep the original rules explicitly and amortize only the expensive probabilistic aggregation.

The closest conceptual inspiration is CMR [Debot et al., 2024] due to its rule selection. CMR is a neurosymbolic model that performs rule learning and works in settings without expert knowledge. In contrast, we exploit CMR's idea of rule selection in a different setting. The semantics of our surrogate model are effectively the sound probabilistic semantics of DeepProbLog [Manhaeve et al., 2018], combined with the rule-selection idea used for scalability in models such as CMR and frameworks like DeepStochLog [Winters et al., 2022].

Setting	Test set size	Method	Time (s)	Memory (MB)	KL divergence
1 digit	5000	Original model	0.22 ± 0.00	699	–
1 digit	5000	Rule-selection surrogate	0.09 ± 0.00	44	$(0.34 \pm 0.01) \times 10^{-3}$
1 digit	5000	Sampling (N=1600)	0.10 ± 0.00	878	$(1.94 \pm 0.01) \times 10^{-3}$
1 digit	5000	Sampling (N=4000)	0.21 ± 0.00	2172	$(0.83 \pm 0.00) \times 10^{-3}$
2 digits	2500	Original model	0.40 ± 0.00	1515	–
2 digits	2500	Rule-selection surrogate	0.07 ± 0.00	210	0.04 ± 0.01
2 digits	2500	Sampling (N=20)	0.09 ± 0.00	1834	0.13 ± 0.00
2 digits	2500	Sampling (N=100)	0.36 ± 0.00	9081	0.04 ± 0.00

Table 1: Results on MNIST-Addition for different numbers of digits per term. We report the test set size, the time for one inference pass through the dataset, the KL divergence to the exact teacher distribution, and the GPU peak memory usage.

6 EXPERIMENTS

Datasets. We adapt MNIST-Addition [Manhaeve et al., 2018] in two configurations: adding two one-digit numbers and adding two two-digit numbers, with each digit represented by an MNIST image. Our setup is harder than the usual formulation of MNIST-Addition: each digit is represented by 10 independent Bernoulli concepts, one per digit class, rather than by a single categorical variable with 10 values. This makes the rules not mutually exclusive, and thus the probabilistic inference harder. 1-digit MNIST-Addition has 100 rules and 20 concepts, while 2-digit MNIST-Addition has 10000 rules and 40 concepts.

Baselines and rule-selection surrogates. The exact teacher probabilities are obtained with DeepProbLog inference using knowledge compilation [Manhaeve et al., 2018]. Concretely, the DeepProbLog program is compiled to an arithmetic circuit, using KLay [Maene et al., 2025] to execute the circuit efficiently on GPU. The rule-selection surrogate uses the same rules and digit concept predictors, adds a rule that trivially evaluates to $True$, and learns only the selector from teacher probabilities. As a baseline, we consider sampling concept assignments from the predicted concept marginals, where the sampling and evaluation is parallelized on GPU.

Metrics. We measure inference time for one forward pass through the test set, peak GPU memory usage, and KL divergence to the exact model's probabilities. We report mean and standard-deviation over three seeds. We also compute the mean approximation error, ignoring overestimations as they are negligible in magnitude and frequency.

For additional details of the setup, we refer to Appendix A.

6.1 RESULTS AND DISCUSSION

Rule-selection surrogates substantially reduce inference time compared to the original model (Table 1). Replacing exact inference in the original model with exact inference in the surrogate yields a $2.4\times$ speedup for 1-digit and a $5.7\times$ speedup for 2-digit MNIST-Addition. Additionally, we require a substantially lower peak memory usage.

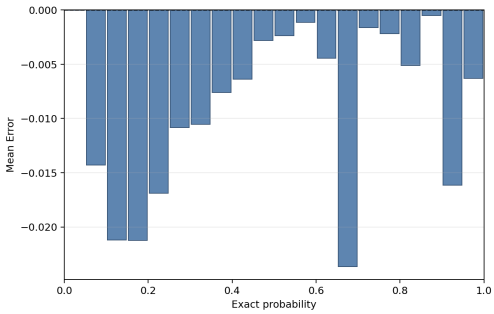

Figure 2: Mean approx. error of rule-selection-surrogates by exact teacher probability (2-digit MNIST-Addition).

The surrogate exhibits only small errors (Figure 2). The mean approximation error remains close to zero for all probability bins, indicating that the surrogate closely approximates the exact predictive distribution. We noticed that the errors are consistently negative, suggesting a tendency to underestimate probabilities. Moreover, the selector assigns essentially no probability mass to the rule that trivially evaluates to True, with an average mass below $10^{-6}\%$ for 2-digit MNIST-Addition. This is desirable because it means the surrogate rarely falls back on the uninterpretable residual rule, and instead relies almost entirely on the original rules.

7 CONCLUSION

We introduced rule-selection surrogates for tractable approximate inference in probabilistic neurosymbolic models. The surrogate replaces expensive disjunctive inference with a linear-time computation while preserving the original symbolic structure. We showed that the surrogate can represent any target Bernoulli marginal pointwise, and preliminary experiments suggest large gains in inference cost with only small approximation error. Future work could evaluate the approach on more benchmarks against state-of-the-art approximate inference approaches.

Acknowledgements

This research has received funding from the Research Foundation-Flanders FWO (GA No. G033625N) and from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme. DD is a fellow of the Research Foundation-Flanders (FWO-Vlaanderen, 1185125N).

References

Arthur Choi, Mark Chavira, and Adnan Darwiche. Node splitting: A scheme for generating upper bounds in bayesian networks. *arXiv preprint arXiv:1206.5251*, 2012.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA*, 2020.

Adnan Darwiche. On the tractable counting of theory models and its application to truth maintenance and belief revision. *Journal of Applied Non-Classical Logics*, 11 (1-2):11–34, 2001.

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

Adnan Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *International Joint Conference on Artificial Intelligence*, pages 819–826, 2011.

Luc De Raedt and Angelika Kimmig. Probabilistic logic programming. *Machine Learning*, 100(1):5–47, 2015.

David Debot, Pietro Barbiero, Francesco Giannini, Gabriele Ciravegna, Michelangelo Diligenti, and Giuseppe Marra. Interpretable concept-based memory reasoning. In *Advances in Neural Information Processing Systems*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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.

Jaron Maene, Vincent Derkinderen, and Pedro Zuidberg Dos Martires. Klay: Accelerating arithmetic circuits for neurosymbolic ai. In *International Conference on Learning Representations*, 2025.

Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, 2018.

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

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

Emile van Krieken, Thiviyan Thanapalasingam, Jakub M. Tomczak, Frank van Harmelen, and Annette ten Teije. A-nesi: A scalable approximate method for probabilistic neurosymbolic inference. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. Deepstochlog: Neural stochastic logic programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10090–10100, 2022.

Rule-Selection Surrogates for Tractable Neurosymbolic Inference (Supplementary Material)

David Debot[1] **Giuseppe Marra**[2]

[1,2]KU Leuven, Department of Computer Science

A EXPERIMENTAL SETUP

We first train a concept predictor on MNIST-Addition [Manhaeve et al., 2018] by exploiting direct supervision on the digits. Alternatively, the concept predictor may be trained using distant supervision from the target label; our approach is agnostic to where the concept predictor comes from. Then, we evaluate the exact symbolic teacher (a DeepProbLog program with the rules of addition) and export exact teacher probabilities for the train, validation, and test splits. As validation split, we use 10% of the training data. We train the rule-selection surrogate using different seeds on these teacher probabilities, and evaluate the surrogate. We also evaluate the GPU sampling baseline across a sweep of sample counts (between 100 and 5000 for 1-digit MNIST-Addition, and between 10 and 200 for 2-digit MNIST-Addition). The concept predictor and the symbolic rule set are kept fixed when learning the selector. Only the selector parameters are trained in the surrogate stage.

All our metrics are averaged over 3 repeated evaluation runs, and we report mean and standard-deviation.

For the rule selector neural network (that parameterizes $p(S \mid x)$), we let it operate on the predicted concept probabilities rather than directly on the raw image input. The selector is a small multilayer perceptron.

This sampling baseline approximates the predictive distribution of the original exact teacher by sampling possible worlds (over the concept random variables) and evaluating the symbolic rules on GPU.

We use a batch size of 256 and train the surrogate for 200 epochs for 1-digit MNIST-Addition and 500 epochs for 2-digit MNIST-Addition. The learning rate is 0.001.

We ran our experiments on a machine with 256GB memory, an NVIDIA GeForce RTX 3080 Ti GPU and an Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz.