Differentiable Answer Set Programming for Neurosymbolic AI

Sota Moriyama[1,2]

[1]The Graduate University for Advanced Studies, SOKENDAI, Shonan Village, Hayama, Kanagawa 240-0193 Japan
[2]National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan

Abstract

Neurosymbolic AI aims to combine the perceptual capabilities of neural networks with the reasoning capabilities of symbolic systems. Prior works have proposed the use of Answer Set Programming (ASP) as a reasoning engine in neurosymbolic pipelines, but these approaches often depend on classical ASP solvers, creating a scalability bottleneck. Our work focuses on developing differentiable ASP solvers that can be integrated seamlessly with neural networks, enabling scalable and end-to-end trainable neurosymbolic systems. We propose NDProp (Neural Decision-Propagation), a differentiable bottom-up ASP computation method that alternates falsity decisions and truth propagations. Through experiments on standard neurosymbolic benchmarks, NDProp demonstrates strong performance in terms of both accuracy and scalability compared to existing ASP-based neurosymbolic approaches. In future work, we plan to investigate the effectiveness of NDProp in applications that require both large-scale neural perception and complex reasoning. Autonomous driving is our primary target, where safe and efficient deployment requires both real-time perception and complex reasoning.

Keywords

Answer Set Programming, Neurosymbolic AI, Fuzzy Logic, Differentiable Solving

1. Introduction

Neural AI, which is based on neural networks, has achieved remarkable success in various domains such as computer vision, natural language processing, and speech recognition. Symbolic AI, on the other hand, excels at tasks that require structured reasoning and explicit knowledge representation, but can struggle with tasks that require robust perception and generalization or large amounts of data. Neurosymbolic AI aims to combine the strengths of both approaches, allowing for systems that can perform complex reasoning while also being able to learn from data and generalize to new situations [1, 2]. The systems are used for solving tasks in various domains such as visual question answering [3], concept learning [4], compliance checking [5], and many more [6].

One approach towards building neurosymbolic AI systems is based on Answer Set Programming (ASP) [7, 8]. Works such as NeurASP [9] have proposed to use ASP as a reasoning engine in neurosymbolic pipelines, allowing for the integration of neural perception with symbolic reasoning. SLASH [10] extends this line of work with neural-probabilistic predicates and removes low-probability predicate outcomes before ASP solving, thereby reducing the number of stable models that must be considered. However, these approaches largely follow a hybrid design in which neural components and symbolic ASP solvers remain separated by an interface: continuous neural predictions must be translated into symbolic atoms/literals, constraints, or probabilistic facts before reasoning can be performed. This introduces a combinatorial grounding problem, as the system must reason over multiple candidate associations between perceptual entities and symbolic objects before being processed by an ASP solver.

To overcome such limitations, we have explored the possibility of developing differentiable solvers that can be integrated seamlessly with neural networks, enabling scalable and end-to-end trainable neurosymbolic systems. Initially, we have worked on using Graph Neural Networks (GNNs) for Minimal Unsatisfiable Subset extraction in Boolean Satisfiability (SAT) [11] and Maximum Satisfiability (MaxSAT)

Joint Proceedings of the Workshops and Doctoral Consortium of the 42nd International Conference on Logic Programming (ICLP 2026) co-located with the Federated Logic Conference (FLoC 2026).

sotam@nii.ac.jp (S. Moriyama)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-03

CEUR-WS.org/Vol-4311/DC_short_11.pdf

solving [12], showing that these approaches can learn to predict solutions to their respective problems and can be effective when used as heuristics for classical solvers. Here, differentiable means that the computation is performed using continuous operations for which gradients can be computed, allowing its parameters and connecting neural components to be trained jointly through gradient-based optimization.

Building on this foundation, we recently proposed Neural Decision–Propagation (NDProp), a differentiable solver based on a bottom-up computation scheme that alternates falsity decisions and truth propagation steps. In the crisp (binary) case, this approach is proven to capture stable-model semantics. With the use of fuzzy operators and neural decision modules, NDProp can be implemented efficiently with modern deep learning frameworks and can be trained in an end-to-end manner, effectively mitigating problems that arise from the interface between neural perception and symbolic reasoning. Experiments on standard neurosymbolic benchmarks show that NDProp performs strongly in terms of both accuracy and scalability compared to existing ASP-based neurosymbolic approaches, and can be effectively used as a differentiable reasoning component in neurosymbolic systems.

One of the main open problems is how to extend the NDProp framework to support more complex ASP constructs such as constraints, choice rules, and aggregates while preserving a clear connection to stable model semantics. While the core mechanism of NDProp has been established for basic normal ASP programs, these extensions are crucial for practical applicability, as they are central in real-world ASP modeling. We also view this line of research as potentially useful beyond neurosymbolic learning. By studying how solver-like behavior can be learned and approximated in a differentiable framework, our findings may provide insights for the design of new heuristics for classical ASP solvers, or for hybrid solving strategies in settings where existing solvers struggle.

Another open problem is how NDProp can be better aligned with Fuzzy Answer Set Programming (FASP) [13, 14]. FASP is a combination of ASP and fuzzy logic, with the goal of expressing the uncertainty that real world problems often involve. This aligns with our motivation of NDProp, which is to develop a differentiable ASP solver that can handle the uncertainty that arises from neural perception, and thus we view this as an important direction for future work. This direction could potentially lead to a deeper understanding of the relationship between differentiable solving and fuzzy logic, and could also provide insights into how to design better architectures for differentiable ASP solvers.

Finally, applications are a major focus of our future work. Recently, there has been a growing interest in using neurosymbolic techniques for autonomous driving, where the human-made requirements are given as logical rules [15, 16]. Generally, these are tackled by using loss functions that penalize the violation of the rules, but this approach can be inefficient and may not fully capture the reasoning capabilities of symbolic systems [17, 18]. Our plan is to use NDProp as a differentiable reasoning component in such settings, allowing for end-to-end training of neurosymbolic systems that can perform complex reasoning while aligning with human-made requirements.

2. Related Works

ASP-based Neurosymbolic AI. ASP-based neurosymbolic AI has been explored in various works, with NeurASP [9] being one of the pioneering approaches that integrates neural perception with symbolic reasoning using ASP. NeurASP directly extends ASP programs with neural predicates, allowing neural network outputs to be translated into probabilistic facts that are then processed by a classical ASP solver. However, scalability has been a major issue for NeurASP [19]. Subsequent systems such as SLASH [10] address this issue by dynamically pruning unlikely symbolic solutions using probabilistic feedback from neural components, thereby reducing the cost of ASP reasoning. Nevertheless, these approaches still maintain an explicit interface between neural and symbolic components: continuous neural predictions must be mapped to discrete symbolic objects before reasoning is performed by an ASP solver. As a result, the combinatorial nature of symbolic inference is not eliminated, and discrete search remains a potential scalability bottleneck.

Differentiable Solvers. Differentiable solvers have been explored in various domains such as Boolean Satisfiability (SAT) [20, 21], Constraint Satisfaction Problems (CSPs) [22] and Minimal Unsatisfiable Subset extraction [11]. These approaches have shown to be able to learn to predict the solutions to their respective problems, and have been effective when used as a heuristic for classical solvers. Furthermore, our recent work on differentiable MaxSAT solving has shown that fuzzy logic can be utilized in the architecture design to achieve stronger performance compared to other continuous approaches [12].

End-to-end differentiable ASP solvers have also been explored to some extent in the literature. Sato et al. [23] use matrix representations of logic programs to derive the cost function for directly computing stable models, without performing any discrete search-like computation. The stability is enforced by adding loop formulas to the program. Ielo and Ricca [24] use Graph Neural Networks (GNNs) to solve negative two literal programs, with the GNN focused on computing supported models, which in this case coincide with stable models. While these approaches are interesting and have merits, the solving process does not closely resemble the search procedure of classical ASP solvers, making it difficult to encompass as a general differentiable extension of ASP solving.

3. Goal of the research

The overall goal is to develop a differentiable ASP solver that can be trained end-to-end with neural perception modules, enabling scalable and efficient ASP-based neurosymbolic systems. To achieve this, we will first develop the NDProp framework, which is based on a novel bottom-up computation scheme that alternates falsity decisions and truth propagation steps. The focus on false decisions is to align with reduct-based stable-model checking, where atoms not in the candidate set are treated as false. We will then expand the framework to support more complex ASP constructs such as constraints, choice rules, and aggregates. Finally, we will investigate the effectiveness of NDProp in applications that require both large-scale neural perception and complex reasoning, with a focus on autonomous driving.

4. Current status of the research

The research has progressed through several stages, from developing differentiable solvers, applications in neurosymbolic learning, and the current development of the NDProp framework for differentiable ASP solving.

Graph Neural Networks for Symbolic Solving. As a first step toward differentiable solving, we studied how Graph Neural Networks (GNNs) can support classical symbolic solvers. Our first work was in the domain of Boolean Satisfiability (SAT) [11], where we proposed a GNN architecture trained to predict whether clauses belong to a Minimal Unsatisfiable Subset (MUS) and a hybrid MUS extraction algorithm that uses these predictions as heuristics.

We then extended into the domain of Maximum Satisfiability (MaxSAT) [12], where we proposed a novel GNN architecture designed for differentiable weighted MaxSAT solving. In comparison to our previous work on MUS extraction, the MaxSAT architecture is designed to perform a more complex computation that approximates greedy distributed local search, and is trained without ground-truth assignments by minimizing the violation of clauses. Experiments on MaxSAT benchmark instances showed that the architecture outperforms existing neural architectures and achieves strong performance among continuous MaxSAT approaches. These works were foundational for our current project, as they demonstrate that neural models can learn to perform solver-like computation, and that fuzzy logic can be used as an effective architectural component for differentiable solving.

Differentiable ASP solving. Building on these results, the current core contribution of our project is Neural Decision–Propagation (NDProp), a differentiable ASP computation framework [25]. NDProp extends a novel bottom-up computation scheme where falsity decisions and truth propagation steps are alternated with the use of fuzzy operators and neural decision modules, both of which were shown

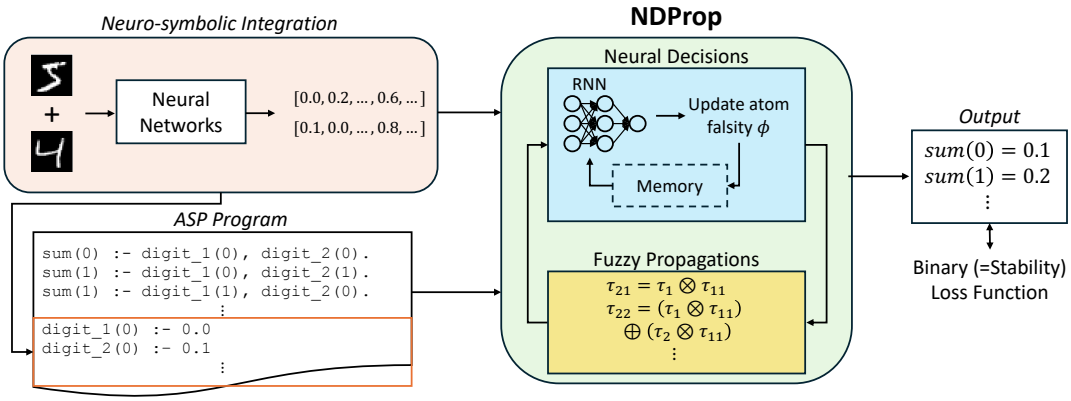


Figure 1: NDProp pipeline. The neuro-symbolic integration is added for specific instantiations, and the arrow leading from the neuro-symbolic block to the ASP program is added only when the backbone neural networks are being trained.

to be effective in our previous work on MaxSAT solving. NDProp has been shown to be effective in traditional neurosymbolic benchmarks, and can be trained end-to-end with neural perception modules, effectively mitigating problems that arise from the interface between neural perception and symbolic reasoning. However, the framework is currently limited to normal ASP programs, and a major ongoing task is to extend it to support more complex ASP constructs such as constraints, choice rules, and aggregates while preserving a clear connection to stable model semantics. What the neural architecture learns is also not yet fully understood, and we are currently investigating how the fuzzy propagation used in NDProp relates to existing Fuzzy Answer Set Programming semantics.

Constraint-aware learning for autonomous driving. On the application side, we have investigated neurosymbolic learning in autonomous driving, where human-written requirements are represented as logical constraints over object-detection outputs. Our work studied how losses that compute how much the neural outputs violate the constraints can be used to reduce violations of such requirements in large-scale perception models [17, 18]. While these studies showed that continuous logical computation is applicable in realistic perception settings, they also highlighted a limitation of constraint-penalty approaches: symbolic knowledge is still introduced externally as a loss term, rather than being integrated as a reasoning procedure. We are currently investigating how NDProp can be used as a differentiable reasoning component in such settings, allowing for end-to-end training of neurosymbolic systems that can perform complex reasoning while aligning with human-made requirements.

5. Preliminary results

The preliminary results of this project focus on NDProp, both in terms of theoretical properties and empirical performance on neurosymbolic benchmarks. The former is given by analyzing the properties of NDProp in the crisp case. The main property is summarized in Theorem 1: terminal binary NDProp states characterize stable models. This means that if we run NDProp until convergence and obtain a state where all outputs are binary (0 or 1), then the set of atoms corresponding to the outputs that are 1 will be a stable model of the program. This gives NDProp a semantic foundation, rather than treating it as a purely neural approximation of ASP solving, which is the case for existing differentiable ASP solvers. [1]

Theorem 1 (Binary terminal states characterize stable models)**.** *Let P be a finite normal ASP program and let $S \subseteq \mathcal{A}_P$, where $\mathcal{A}_P$ is the set of atoms occurring in P. Then S is a stable model of P if and only if*

[1]The full definitions and proofs are given in the extended version at https://arxiv.org/abs/2605.01797

	Addition			Addition2			Add2x2			Apply2x2			Member3			Member5		
	NDProp	NASP	SLASH	NDProp	NASP	SLASH	NDProp	NASP	SLASH	NDProp	NASP	SLASH	NDProp	NASP	SLASH	NDProp	NASP	SLASH
Train	**160**	1612	430	**235**	TO	2374	**692**	TO	TO	346	385	**166**	**110**	2829	TO	**969**	TO	TO
Test	**0.37**	24.25	9.94	**1.28**	33.90	3.73	**1.15**	41.59	3.75	**1.52**	5.31	4.32	**0.79**	5.13	3.08	**4.40**	496.4	6.38
R.Acc	**99.0**	98.5	98.9	98.9	98.5	**99.0**	**99.0**	98.6	98.9	**100.0**	**100.0**	**100.0**	**97.3**	96.8	97.2	**97.0**	11.6	8.9
I.Acc	**98.2**	97.1	97.8	96.1	94.6	**96.5**	**95.7**	93.70	95.2	**100.0**	**100.0**	**100.0**	**99.0**	98.6	98.7	**97.5**	59.7	60.3

Table 1
Neuro-symbolic MNIST results for NDProp, NeurASP (NASP) and SLASH, all trained for 10 epochs. Metrics are Train: Training time, Test: Testing time, R.Acc: Recognition Accuracy (%), and I.Acc: Inference accuracy (%).

there exists a terminal binary NDProp state, obtained after propagation has been applied until convergence, such that the outputs satisfy $p_i \in \{0, 1\}$ *for all atoms and* $S = \{a_i \mid p_i = 1\}$.

Table 1 summarizes our initial neurosymbolic benchmark results on MNIST-based tasks, comparing NDProp with NeurASP [9] and SLASH [10]. The neurosymbolic pipeline using NDProp is shown in Figure 1. Across the tested tasks, NDProp achieves competitive or better recognition and inference accuracy while substantially reducing test-time computation. For example, on Addition, NDProp achieves 99.0% recognition accuracy and 98.2% inference accuracy, with a test time of 0.37 seconds, compared with 24.25 seconds for NeurASP and 9.94 seconds for SLASH. The scalability advantage becomes more pronounced on harder tasks: on Member5, NDProp retains 97.0% recognition accuracy and 97.5% inference accuracy with a test time of 4.40 seconds, while NeurASP and SLASH show much lower accuracies and substantially larger test times. These results support the main motivation of the project: replacing classical solvers with a differentiable computation can overcome the problems that arise from the interface between the neural and symbolic components.

6. Open issues and expected achievements

While NDProp has shown promising results, several important issues remain open. The first is expressivity: the current theoretical development of NDProp mainly focuses on normal ASP programs, while practical ASP modeling often relies on constraints, choice rules, aggregates, weak constraints, and optimization statements. Extending NDProp to such constructs while preserving a clear connection to stable-model semantics is a central goal of the next stage of the project. The second issue is semantic: since NDProp uses fuzzy operators to relax propagation and reason from uncertain inputs, its relationship to Fuzzy Answer Set Programming should be clarified. A better understanding of this relationship would make the framework more principled and would help compare it with existing fuzzy approaches. The third issue is applicability: while the initial results on neurosymbolic benchmarks are promising, it is important to investigate how NDProp can be used in larger-scale applications that require both neural perception and symbolic reasoning, such as autonomous driving.

Overall, this research aims to contribute a new approach to ASP-based neurosymbolic AI in which symbolic reasoning is not merely attached to neural perception through an external solver, but through a differentiable computation. If successful, the project will provide a framework that retains a meaningful connection to stable-model semantics while enabling end-to-end training with neural networks. In the longer term, we expect that this line of research could pave the way for more scalable and efficient neurosymbolic systems. A highly integrated approach that does not require interfacing between the neural and symbolic components is crucial to the further development of ASP-based neurosymbolic AI, and we hope that our work on NDProp will be a significant step in this direction.

Acknowledgement

This work has been supported by JSPS KAKENHI Grant Number JP25K03190, JST CREST Grant Number JPMJCR22D3 and Grant-in-Aid for JSPS Fellows Grant Number JP26KJ1236, Japan.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT for Grammar and spelling checks. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] A. d'Avila Garcez, L. C. Lamb, Neurosymbolic AI: the 3rd wave, Artif. Intell. Rev. 56 (2023) 12387–12406.
[2] W. Wang, Y. Yang, F. Wu, Towards data-and knowledge-driven AI: A survey on neuro-symbolic computing, IEEE Trans. Pattern Anal. Mach. Intell. 47 (2025) 878–899.
[3] T. Eiter, N. Higuera, J. Oetsch, M. Pritz, A neuro-symbolic ASP pipeline for visual question answering, Theory Pract. Log. Program. 22 (2022) 739–754.
[4] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, J. Wu, The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision, in: Proceedings of ICLR 2019, OpenReview.net, 2019.
[5] V. Barbara, M. Guarascio, N. Leone, G. Manco, A. Quarta, F. Ricca, E. Ritacco, Neuro-symbolic AI for compliance checking of electrical control panels, Theory Pract. Log. Program. 23 (2023) 748–764.
[6] M. Borroto, A. Ielo, G. Mazzotta, F. Ricca, Answer set programming and neurosymbolic ai: Applications and future perspectives (invited talk), Springer Nature Switzerland, Cham, 2025, pp. 1–21.
[7] V. Lifschitz, What is answer set programming?, in: D. Fox, C. P. Gomes (Eds.), Proceedings of AAAI 2008, AAAI Press, 2008, pp. 1594–1597.
[8] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, Commun. ACM 54 (2011) 92–103.
[9] Z. Yang, A. Ishay, J. Lee, NeurASP: Embracing neural networks into answer set programming, in: Proceedings of IJCAI 2020, ijcai.org, 2020, pp. 1755–1762.
[10] A. Skryagin, D. Ochs, D. S. Dhami, K. Kersting, Scalable neural-probabilistic answer set programming, J. Artif. Intell. Res. 78 (2023) 579–617.
[11] S. Moriyama, K. Watanabe, K. Inoue, GNN based extraction of minimal unsatisfiable subsets, in: E. Bellodi, F. A. Lisi, R. Zese (Eds.), ILP, LNCS, Springer, 2023, pp. 77–92.
[12] S. Moriyama, K. Inoue, Graph-based attention for differentiable maxsat solving, in: Advances in Neural Information Processing Systems, volume 38, Curran Associates, Inc., 2025, pp. 137740–137768.
[13] M. Alviano, R. Peñaloza, Fuzzy answer sets approximations, Theory Pract. Log. Program. 13 (2013) 753–767.
[14] M. Blondeel, S. Schockaert, D. Vermeir, M. D. Cock, Fuzzy answer set programming: An introduction, in: Soft Computing: State of the Art Theory and Novel Applications, Studies in Fuzziness and Soft Computing, Springer, 2013, pp. 209–222.
[15] E. Giunchiglia, M. C. Stoian, S. Khan, F. Cuzzolin, T. Lukasiewicz, ROAD-R: the autonomous driving dataset with logical requirements, Mach. Learn. 112 (2023) 3261–3291.
[16] S. Khan, I. Teeti, R. J. Alitappeh, M. C. Stoian, E. Giunchiglia, G. Singh, A. Bradley, F. Cuzzolin, Road-waymo: Action awareness at scale for autonomous driving, CoRR abs/2411.01683 (2024).

[17] S. Moriyama, K. Watanabe, K. Inoue, A. Takemura, MOD-CL: multi-label object detection with constrained loss, CoRR abs/2403.07885 (2024).
[18] T. Eiter, K. Inoue, N. Higuera, S. Moriyama, T-norm selection for object detection in autonomous driving with logical constraints, in: Advances in Neural Information Processing Systems, volume 38, Curran Associates, Inc., 2025, pp. 86906–86933.
[19] Z. Yang, Extending answer set programs with neural networks, in: Proceedings 36th International Conference on Logic Programming (Technical Communications), EPTCS, 2020, pp. 313–322.
[20] D. Selsam, M. Lamm, B. Bünz, P. Liang, L. de Moura, D. L. Dill, Learning a SAT solver from single-bit supervision, in: Proceedings of ICLR 2019, OpenReview.net, 2019.
[21] D. Selsam, N. S. Bjørner, Guiding high-performance SAT solvers with unsat-core predictions, in: Proceedings of SAT 2019, LNCS, Springer, 2019, pp. 336–353.
[22] J. Tönshoff, B. Kisin, J. Lindner, M. Grohe, One model, any CSP: graph neural networks as fast global search heuristics for constraint satisfaction, in: IJCAI 2023, ijcai.org, 2023, pp. 4280–4288.
[23] T. Sato, A. Takemura, K. Inoue, Towards end-to-end asp computation, Neurosymbolic Artificial Intelligence 2 (2026) 29498732261419298.
[24] A. Ielo, F. Ricca, Answer set computation of negative two-literal programs based on graph neural networks: Preliminary results, in: Proceedings of the 36th Italian Conference on Computational Logic, CEUR Workshop Proceedings, CEUR-WS.org, 2021, pp. 188–195.
[25] T. Eiter, K. Inoue, S. Moriyama, Neural Decision-Propagation for Answer Set Programming, in: Proceedings of IJCAI-ECAI, 2026, to appear.