Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
KR meets Machine Learning and Explanation

Gradient-Based Optimization on Gödel Logic as Discrete Local Search

Alessandro Daniele[1,2] , **Emile van Krieken**[3]
[1]University of Bozen-Bolzano, Bozen, Italy
[2]Fondazione Bruno Kessler, Trento, Italy
[3]Vrije Universiteit Amsterdam, Amsterdam, Netherlands
alessandro.daniele@unibz.it, e.van.krieken@vu.nl

Abstract

A fundamental challenge in neurosymbolic systems is applying continuous gradient-based optimization to discrete logical domains. While fuzzy relaxations provide differentiability, they often lack a formal structural alignment with classical logic. In this work, we show that Gödel semantics addresses this limitation through a homomorphism that maps its continuous interpretations to Boolean ones, allowing discrete variables to be encoded while maintaining full differentiability. Building on this foundation, we show that gradient-based optimization on Gödel logic instantiates a discrete local search for Boolean satisfiability. Our formal analysis proves that each optimization step identifies and modifies a single variable within an unsatisfied clause, precisely mimicking the steps of a discrete solver. We identify local optima as the primary limitation of such dynamics and introduce the Gödel Trick, a stochastic reparameterization technique designed to improve the exploration of the solution space. We further show a formal connection between this approach, probabilistic inference, and the Gumbel-Max trick. Experimental results on SAT benchmarks and the Visual Sudoku task validate our theoretical findings, demonstrating that our approach effectively navigates complex combinatorial landscapes and provides a solid foundation for differentiable discrete search.

1 Introduction

Deep learning has revolutionized artificial intelligence, driven by the power of Gradient-Based Optimization (GBO) and its efficient implementation, that is the backpropagation (Rumelhart, Hinton, and Williams 1986). While GBO excels in continuous domains, its integration into neurosymbolic (NeSy) systems remains a significant challenge due to the inherently discrete and combinatorial nature of symbolic reasoning (Feldstein et al. 2024; Marra et al. 2024). The difficulty does not lie merely in rendering logical constraints differentiable, but in doing so without sacrificing the structural rigor of classical logic.

A common strategy in the NeSy field is to employ fuzzy logics as continuous relaxations of Boolean operators (van Krieken, Acar, and van Harmelen 2022; Badreddine et al. 2022; Daniele and Serafini 2019). By allowing truth values to range across the unit interval $[0, 1]$, these "soft" logics provide a differentiable landscape suitable for GBO. However, this transition frequently leads to a semantic mismatch: as no fuzzy logic can satisfy all properties of classical logic (Gupta and Qi 1991), the resulting continuous approximations often diverge from the intended logical behavior, failing to capture the discrete essence of symbolic tasks (van Krieken, Acar, and van Harmelen 2022).

In this work, we demonstrate that Gödel logic is a remarkable exception to this trend. Rather than acting as a mere "soft" surrogate, we prove that Gödel semantics possesses unique algebraic properties that establish a structural bridge to Boolean logic. Specifically, we identify a homomorphism between Gödel and Boolean semantics that allows for a consistent mapping between the continuous and discrete worlds. Our central argument is that Gödel logic is "discrete in disguise": we prove that its truth function produces sparse gradients, such that each optimization step identifies and modifies only a single variable and, when applied to a Conjunctive Normal Form (CNF) formula, it identifies and modifies only a single unsatisfied clause. This reveals that Gradient-based Optimization on Gödel logic (GBOG) does not simply approximate logic; it formally instantiates a discrete local search for Boolean satisfiability.

By characterizing GBOG as a discrete Local Search Algorithm (LSA) for SAT, we can identify its primary theoretical limitation: similar to deterministic solvers like GSAT, it is prone to converging to local optima. To overcome this, we introduce the *Gödel Trick* (GT), a stochastic reparameterization technique that introduces noise into the optimization process to improve exploration. We show that the Gödel Trick is not merely a heuristic addition of noise, but provides a formal probabilistic grounding, acting as a Monte Carlo estimator for Weighted Model Counting (WMC).

Our main theoretical contributions are threefold: (i) we prove the existence of a homomorphism between Gödel and classical logic semantics; (ii) we provide a formal proof of the sparsity of the Gödel gradient, demonstrating that gradient-based optimization on Gödel logic circuits behaves as a LSA; (iii) we identify local optima as the fundamental barrier to convergence in this framework and introduce the Gödel Trick (GT), a stochastic variant of Gödel optimization. Moreover, we establish its formal connection to probabilistic inference, showing it serves as an efficient Monte Carlo estimator for probabilistic interpretations and connecting it to the Gumbel-Max trick (Gumbel 1954; Jang, Gu, and Poole 2022).

We validate these theoretical findings on SAT benchmarks (Hoos and Stützle 2000) and the Visual Sudoku

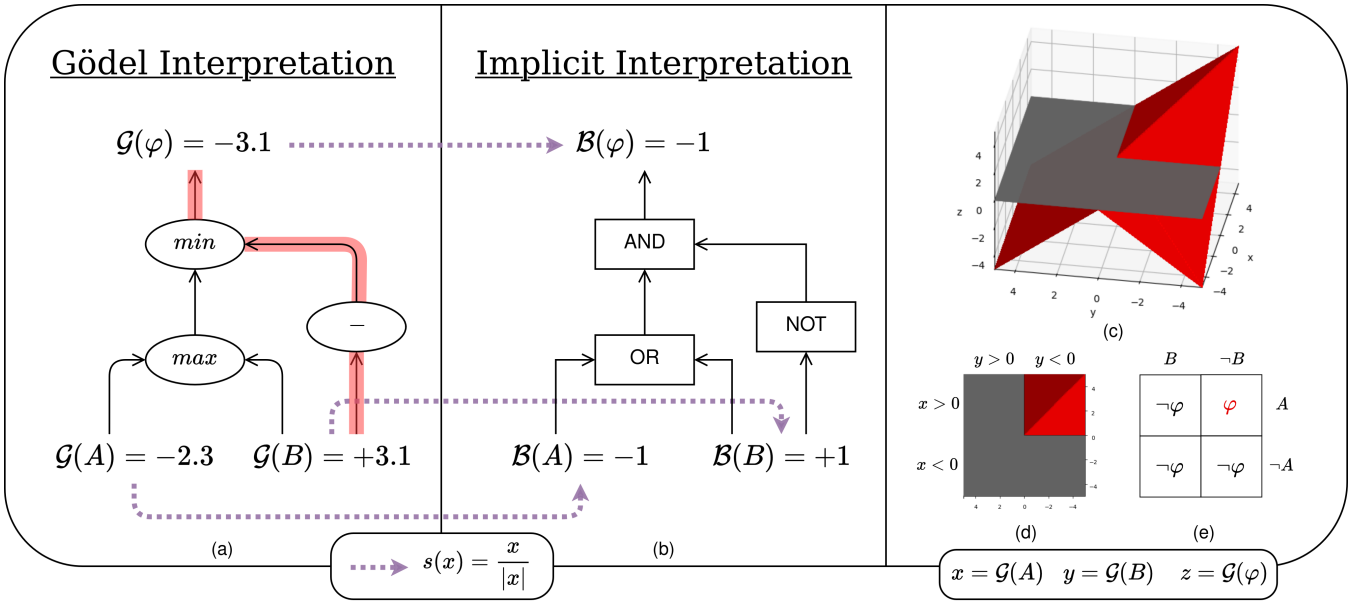

Figure 1: **The starting point of our work: there is a consistent mapping from Gödel to boolean interpretations.** Example for formula $\varphi = (A \vee B) \wedge \neg B$. (a) Gödel interpretation (active path highlighted in red); (b) the corresponding boolean interpretation; (c-d) plot of Gödel interpretation for φ (red) and the plane $z = 0$ (gray): (c) side view, (d) top view; (e) truth table of φ in boolean logic. There is a one-to-one correspondence between the quadrants in (d) and the discrete values in (e).

task (Augustine et al. 2022). To support reproducibility, we provide the source code at https://github.com/DanieleAlessandro/Godel-Trick; additional experimental details are included in the supplementary materials, available as ancillary files in the arXiv version of the paper: https://arxiv.org/abs/2503.01817.

2 Background and Notation

Propositional formulas are defined recursively as: $\varphi ::= p \mid \neg\varphi_1 \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2)$, where $p \in P$ is an atomic proposition, and $\neg$, $\wedge$, $\vee$ denote negation, conjunction, and disjunction.

We consider two semantics: classical (Boolean) logic and Gödel logic. In classical logic, formulas are assigned binary truth values in $\{-1, 1\}$ [1]. Gödel logic generalises classical logic by allowing truth values to be real numbers in the range $[0, 1]$, where 0 represents false, 1 represents true. In the remainder, we adopt truth values of formulas over the real number space $\mathbb{R}$ instead of $[0, 1]$ [2].

An interpretation is a function that maps formulas in truth values and is defined recursively as:

Formula	**Boolean** $\mathcal{B}(\varphi)$	**Gödel** $\mathcal{G}(\varphi)$
p_i	$\in \{-1, 1\}$	$\in \mathbb{R}$
$\neg\varphi$	$-\mathcal{B}(\varphi)$	$-\mathcal{G}(\varphi)$
$\varphi_1 \wedge \varphi_2$	$\min(\mathcal{B}(\varphi_1), \mathcal{B}(\varphi_2))$	$\min(\mathcal{G}(\varphi_1), \mathcal{G}(\varphi_2))$
$\varphi_1 \vee \varphi_2$	$\max(\mathcal{B}(\varphi_1), \mathcal{B}(\varphi_2))$	$\max(\mathcal{G}(\varphi_1), \mathcal{G}(\varphi_2))$

Note that the interpretation of any formula is recursively determined by the interpretation of atomic propositions. In the following, we sometimes represent an interpretation compactly using a vector of truth values over the atomic propositions: $\boldsymbol{\mu} = \langle \mathcal{G}(p_i) \rangle_{i=1}^{n}$, and write $\mathcal{G}(\varphi; \boldsymbol{\mu})$ to denote the value of formula φ, assuming $\mathcal{G}(p_i) = \mu_i$. This notation is particularly useful for Theorem 3, as it allows us to treat the interpretation as a fixed function on vector $\boldsymbol{\mu}$. Similarly, we use $\boldsymbol{\gamma} = \langle \mathcal{B}(p_i) \rangle_{i=1}^{n}$ and $\mathcal{B}(\varphi; \boldsymbol{\gamma})$ for the boolean interpretation.

3 Related Work

The neurosymbolic (NeSy) field focuses on incorporating logical knowledge into learning systems (Selsam et al. 2018; Li and Si 2022; Wang et al. 2019).

A prominent research direction investigates the continuous relaxation of logical constraints by adopting t-norm based fuzzy semantics. This framework enables the definition of differentiable logic layers, where logical formulas are translated into continuous functions suitable for gradient-based optimization. Logic Tensor Networks (LTN) (Badreddine et al. 2022) and Semantic-Based Regularization (SBR) (Diligenti, Gori, and Sacca 2017) inject prior knowledge into neural networks by embedding fuzzy logic in loss functions. These methods allow for the choice of different

[1] Without loss of generality, we use 1 and -1 as *true* and *false*, respectively

[2] We consider logits in $\mathbb{R}$, that is, the outputs of a neural network before applying the sigmoid function. More details in the Supplementary Materials.

fuzzy semantics that have been extensively analyzed and compared in terms of expressiveness and optimisation (van Krieken, Acar, and van Harmelen 2022; Grespan, Gupta, and Srikumar 2021; Flinkow, Pearlmutter, and Monahan 2024; Slusarz et al. 2023). Furthermore, fuzzy logic operators can be used as part of the neural network architecture itself to incorporate background knowledge (Giunchiglia et al. 2024; Daniele and Serafini 2019; Daniele et al. 2023b).

Gödel logic is frequently included as a standard semantic choice in various neurosymbolic frameworks (Badreddine et al. 2022; Daniele et al. 2023b), and as the only choice in other works (Daniele and Serafini 2019; Andreoni et al. 2025). However, in these contexts, it is primarily utilized as a continuous surrogate for Boolean constraints. Our contribution offers a novel perspective: by leveraging its specific algebraic properties, namely the existence of a homomorphism to Boolean logic, we show that gradient-based optimization on Gödel semantics can be formally interpreted as a discrete local search. Our analysis of this structural alignment shifts the focus from treating Gödel logic merely as a continuous relaxation to its specific optimization challenges, such as convergence to local optima. These issues are subsequently addressed by our proposed Gödel Trick.

A separate but influential line of research deals with the probabilistic paradigm. NeSy methods that use probabilistic logics, such as Semantic Loss (Xu et al. 2018), DeepProbLog (Manhaeve et al. 2018) and Semantic Probabilistic Layers (Ahmed et al. 2022), require computing the Weighted Model Counting (WMC) (Chavira and Darwiche 2008), a #P-hard problem (Valiant 1979), at each iteration. To overcome this, methods either use compiled probabilistic circuits (Choi, Vergari, and Van den Broeck 2020; Kisa et al. 2014) or approximations, for example via neural networks (van Krieken et al. 2023) or by sampling (De Smet, Sansone, and Zuidberg Dos Martires 2023). While we do not perform a direct empirical comparison with probabilistic methods, as our primary focus is on providing a formal foundation and theoretical characterization of optimization within Gödel logic, in Section 6.2 we bridge the fuzzy and probabilistic paradigms by showing that the Gödel Trick can be interpreted as a Monte Carlo estimator for WMC.

4 Homomorphism from Gödel to Boolean Algebraic Structures

In this section, we demonstrate the existence of an homomorphism that maps continuous interpretations of Gödel logic to discrete boolean interpretations of classical logic. In the case of the Gödel logic, we consider $\mathbb{R} \setminus \{0\}$, since this relation can only be proved when we exclude zero [3].

Two algebraic structures can be defined: for classical logic we have $\mathcal{L}_{\mathcal{B}} = \langle \{-1, 1\}, \min, \max, - \rangle$; while for Gödel semantics we define $\mathcal{L}_{\mathcal{G}} = \langle \mathbb{R} \setminus \{0\}, \min, \max, - \rangle$. Both structures are De Morgan Lattice (Moisil 1935) (also known as distributive i-lattices (Kalman 1958)).

[3] In practice, this is not an issue since we introduce noise during training, making the probability of $\mathcal{G}(\varphi) = 0$ negligible (see Section 6).

The sign function $s : \mathbb{R} \setminus \{0\} \to \{-1, 1\}$ is defined as:

$$s(x) = \frac{x}{|x|} = \begin{cases} -1, & \text{if } x < 0, \\ 1, & \text{if } x > 0. \end{cases}$$

Proposition 1. *The sign function $s(x)$ is a homomorphism from the Gödel lattice $\mathcal{L}_{\mathcal{G}}$ to the boolean lattice $\mathcal{L}_{\mathcal{B}}$.*

Proof. We verify that s preserves the lattice structure:

Negation: Let $x \in \mathbb{R} \setminus \{0\}$. Then, we have

$$s(-x) = -x/|-x| = -(x/|x|) = -s(x)$$

preserving the negation. Note that this would not hold if we assumed zero to be a valid truth value.

Conjunction: Let $x, y \in \mathbb{R} \setminus \{0\}$. Then,

$$s(\min(x, y)) = \min(s(x), s(y))$$

preserving the conjunction. This holds because the minimum of two inputs will be negative iff at least one input is negative.

Disjunction: Let $x, y \in \mathbb{R} \setminus \{0\}$. Then,

$$s(\max(x, y)) = \max(s(x), s(y))$$

preserving the disjunction. This holds because the maximum is positive iff at least one value is positive.

□

As a consequence, the Gödel interpretation of any formula can be mapped to a Boolean interpretation via s.

The relation defined by Proposition 1 plays a crucial role in enabling Gradient-Based Optimization (GBO) to optimise logical formulas in a differentiable setting. At each step of GBO on $\mathcal{G}(\varphi)$, the current Gödel interpretation can be mapped to a corresponding discrete interpretation via the sign function. We call this discrete interpretation the *implicit interpretation*, defined as $\mathcal{B}(\varphi) = s(\mathcal{G}(\varphi))$. Figure 1 shows an example: the implicit interpretation of formula φ can be computed either by discretizing (via the sign function) each proposition and interpreting φ with boolean semantics, or by interpreting it in Gödel semantics and discretizing the final output.

5 Gradient-based Optimization on Gödel Logic

In this section, we analyze the behaviour of Gradient-Based Optimization applied to Gödel logic (GBOG), showing its correspondence with Local Search Algorithms (LSAs). Since the aim is to increase the value of the Gödel interpretation, gradient ascent is applied on $\mathcal{G}(\varphi)$. As illustrated on the left side of Figure 2, the algorithm iteratively adjusts the continuous truth values according to the *update rule* $\boldsymbol{\mu} = \boldsymbol{\mu} + \lambda \nabla_{\boldsymbol{\mu}} \mathcal{G}(\varphi; \boldsymbol{\mu})$, where λ denotes the learning rate. Note that gradient ascent often updates the continuous interpretation without changing the sign of any proposition, leaving the implicit interpretation unchanged. Hence, we restrict our analysis to steps where a sign change occurs.

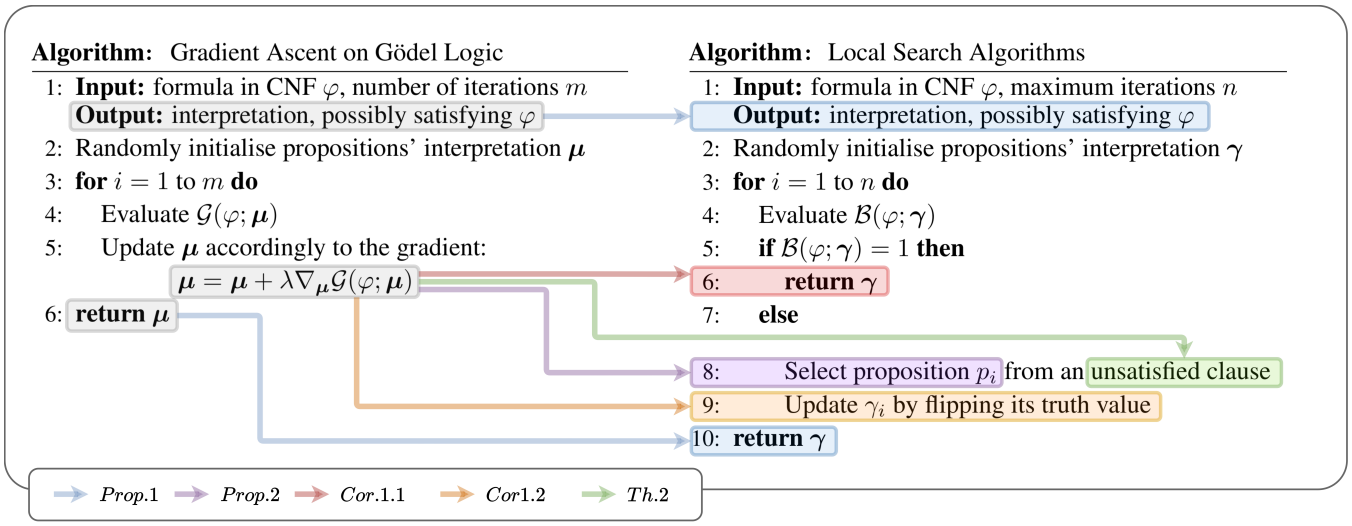

Figure 2: Gradient-based Optimization on Gödel logic (GBOG) acts as a local search algorithm for SAT.

The alignment between GBOG and LSAs is illustrated in Figure 2, which maps each step of the two algorithms based on the following theoretical results. Recall that every continuous interpretation in the Gödel logic maps to a discrete one (Proposition 1). We first prove that the gradients are sparse (Proposition 2), which implies that each gradient ascent step modifies a single proposition. We then prove that if the formula is satisfied, the current interpretation is a fixed point of the update dynamics (Corollary 1.1). Otherwise, the modified variable moves toward the decision threshold via a gradient ascent step (Corollary 1.2), eventually changing its sign. Moreover, the modified variable always appears in an unsatisfied clause (Theorem 2). These results collectively show that GBOG behaves as a discrete and deterministic LSA applied to the corresponding Boolean formula.

To prove the aforementioned alignment, we need to analyze the gradient of a Gödel interpretation w.r.t. the propositions. With abuse of notation, we will refer to the interpretation of a formula by ignoring the function $\mathcal{G}$ when computing the partial derivatives, e.g., $\frac{\partial \mathcal{G}(\varphi)}{\partial \mathcal{G}(\psi)}$ is reported as $\frac{\partial \varphi}{\partial \psi}$.

The first step is to show the sparsity of the gradients. To this end, we define the concept of path.

Definition 1 (Path)**.** *A path $\mathcal{P}$ from a formula φ to an atomic proposition p is a sequence of formulas: $\mathcal{P} = (\varphi, \psi_1, \psi_2, \ldots, p)$, where each ψ_i is a direct subformula of ψ_{i-1}, and p is an atomic proposition.*

An example of a path $\mathcal{P}$ for formula $\varphi = (A \vee B) \wedge \neg B$ is highlighted in red in Figure 1(a):

$$\mathcal{P} = \big((A \vee B) \wedge \neg B, \neg B, B\big)$$

Lemma 1. *The partial derivative $\frac{\partial \varphi}{\partial \psi_i}$ along a path $\mathcal{P} = (\varphi, \psi_1, \ldots, p)$, takes values in $\{-1, 0, 1\}$.*

Proof. The partial derivative of φ with respect to p along the path is given by $\frac{\partial \varphi}{\partial p} = \frac{\partial \varphi}{\partial \psi_1} \cdot \cdots \cdot \frac{\partial \psi_{n-1}}{\partial p}$, obtained by applying the chain rule. Each node $\psi_i \in \mathcal{P}$ represents either a $\min$, $\max$, or negation, and the corresponding partial derivative $\frac{\partial \psi_i}{\partial \psi_{i+1}} \in \{-1, 0, 1\}$. Hence, their product is also in $\{-1, 0, 1\}$. □

Remark 1 (Tie-breaking Rule)**.** *When multiple branches of a $\min$ or $\max$ operation result in the same truth value, we assume that a single branch is selected by the gradient.*

Definition 2 (Active path)**.** *A path $\mathcal{P} = (\varphi, \psi_1, \psi_2, \ldots, p)$ is* active *if the partial derivative $\frac{\partial \varphi}{\partial p}$ is different from zero.*

Proposition 2. *Given a formula φ, there exists a unique active path $\mathcal{P} = (\varphi, \psi_1, \ldots, p)$ in the computational graph of φ* [4]*.*

Proof. We proceed by induction on the structure of the formula.

Base Case (Atomic Proposition): If $\varphi = p$, where p is an atomic proposition, we have $\frac{\partial \varphi}{\partial p} = 1$. There are no intermediate subformulas, so the path is $\mathcal{P} = (p)$, and the gradient is trivially 1.

Inductive Step (Negation): Suppose the inductive hypothesis holds for a subformula ψ; i.e., there exists a unique active path in the computational graph of ψ. Assume $\varphi = \neg\psi$. The gradient of φ with respect to p is:

$$\frac{\partial \varphi}{\partial p} = \frac{\partial \varphi}{\partial \psi} \cdot \frac{\partial \psi}{\partial p} = -\frac{\partial \psi}{\partial p}.$$

which is different from zero, meaning the path is still active.

[4] This result previously appeared in a different form (van Krieken, Acar, and van Harmelen 2022).

Inductive Step (Conjunction/Disjunction): Consider the case where $\varphi = \psi_1 \wedge \psi_2$. By the inductive hypothesis, we know that there exists a unique active path from each subformula ψ_1 and ψ_2 to their respective atomic propositions p_1 and p_2, and that the properties hold along these paths.

Since conjunction is interpreted as the minimum function, only the path corresponding to the subformula with the smallest truth value will have a non-zero gradient. Let's assume $\mathcal{G}(\psi_1) < \mathcal{G}(\psi_2)$, then the gradient with respect to p_1 is:

$$\frac{\partial \varphi}{\partial p_1} = \frac{\partial \varphi}{\partial \psi_1} \cdot \frac{\partial \psi_1}{\partial p_1} = \frac{\partial \psi_1}{\partial p_1}.$$

and the gradient with respect to p_2 is:

$$\frac{\partial \varphi}{\partial p_2} = \frac{\partial \varphi}{\partial \psi_2} \cdot \frac{\partial \psi_2}{\partial p_2} = 0.$$

If $\mathcal{G}(\psi_1) > \mathcal{G}(\psi_2)$, then the gradient with respect to p_1 is zero, while the gradient with respect to p_2 is non-zero. The case for disjunction follows a similar reasoning, with the difference that the maximum function is used. □

As an example, consider Figure 1(a): each partial derivative is zero, except for the path from $\mathcal{G}(\varphi)$ to $\mathcal{G}(B)$ (red path in the figure).

Theorem 1. *Let φ be a formula, and let π be a subformula in the active path of φ. The partial derivative $\frac{\partial \varphi}{\partial \pi}$ is equal to the product of the implicit interpretation of the two formulas:*

$$\frac{\partial \varphi}{\partial \pi} = \mathcal{B}(\varphi) \cdot \mathcal{B}(\pi)$$

Proof. We proceed by induction on the structure of the formula φ.

Base Case: If $\varphi = \pi$, then:

$$\begin{aligned}\frac{\partial \varphi}{\partial \pi} = \frac{\partial \pi}{\partial \pi} = 1 = (\pm 1)^2 \\ = \mathcal{B}(\varphi)^2 = \mathcal{B}(\varphi) \cdot \mathcal{B}(\pi)\end{aligned}$$

Inductive Step (Negation): Consider $\varphi = \neg\psi$. By definition of the negation, $\mathcal{G}(\varphi) = -\mathcal{G}(\psi)$ and $\frac{\partial \varphi}{\partial \psi} = -1$. Applying the chain rule:

$$\begin{aligned}\frac{\partial \varphi}{\partial \pi} = \frac{\partial \varphi}{\partial \psi} \cdot \frac{\partial \psi}{\partial \pi} = -\frac{\partial \psi}{\partial \pi} && (1)\\ = -\mathcal{B}(\psi) \cdot \mathcal{B}(\pi) && (2)\\ = \mathcal{B}(\varphi) \cdot \mathcal{B}(\pi) && (3)\end{aligned}$$

where (2) corresponds to the inductive hypothesis, and (3) is a consequence of $\varphi = \neg\psi$. Hence, the proposition holds.

Inductive Step (Conjunction/Disjunction): Consider $\varphi = \psi_1 \wedge \psi_2$. The derivative is non-zero only for the subformula with the smallest truth value, for which it is 1. Assume ψ_1 to have the smallest truth value, i.e.

$$\mathcal{G}(\varphi) = \min(\mathcal{G}(\psi_1), \mathcal{G}(\psi_2)) = \mathcal{G}(\psi_1)$$

then, we have:

$$\begin{aligned}\frac{\partial \varphi}{\partial \pi} = \frac{\partial \varphi}{\partial \psi_1} \cdot \frac{\partial \psi_1}{\partial \pi} = \frac{\partial \psi_1}{\partial \pi} && (4)\\ = \mathcal{B}(\psi_1) \cdot \mathcal{B}(\pi) && (5)\\ = \mathcal{B}(\varphi) \cdot \mathcal{B}(\pi) && (6)\end{aligned}$$

where (5) holds by the inductive hypothesis, and (6) is a consequence of $\mathcal{G}(\varphi) = \mathcal{G}(\psi_1)$. Similarly, for disjunction, the derivative is non-zero for the subformula with the largest truth value, yielding the same result. □

Theorem 1 shows that the direction of the gradient on a specific formula depends exclusively on the implicit interpretation, as the following two related corollaries highlight.

Corollary 1.1. *Let φ be a formula, and let p be a proposition in the active path. If φ is satisfied ($\mathcal{B}(\varphi) = 1$), then the implicit interpretation is not modified.*

Proof. By Theorem 1, it holds: $\frac{\partial \varphi}{\partial p} = \mathcal{B}(p)$. Hence, the sign is not changed by the update rule:

$$\begin{aligned}\mathcal{B}'(p) = s(\mathcal{G}(p) + \lambda\mathcal{B}(p)) && (7)\\ = s\big(\mathcal{B}(p) \cdot (|\mathcal{G}(p)| + \lambda)\big) = \mathcal{B}(p) && (8)\end{aligned}$$

where $\lambda > 0$ is the learning rate, and $\mathcal{B}'(p)$ is the implicit interpretation after the update. In (8) we exploit the equality $\mathcal{G}(p) = s(\mathcal{G}(p)) \cdot |\mathcal{G}(p)| = \mathcal{B}(p) \cdot |\mathcal{G}(p)|$ to separate $|\mathcal{G}(p)| + \lambda > 0$ from the sign $\mathcal{B}(p)$. □

Corollary 1.2. *Let φ be a formula, and let p be a proposition in the active path. If φ is* not *satisfied ($\mathcal{B}(\varphi) = -1$), then the gradient ascent step moves the variable toward the decision threshold, eventually flipping its sign.*

Proof. By Theorem 1, it holds: $\frac{\partial \varphi}{\partial p} = -\mathcal{B}(p)$. Hence:

$$\mathcal{B}'(p) = s(\mathcal{G}(p) - \lambda\mathcal{B}(p)) = s\big(\mathcal{B}(p) \cdot (|\mathcal{G}(p)| - \lambda)\big)$$

If $|\mathcal{G}(p)| < \lambda$, then $\mathcal{B}'(p) = -\mathcal{B}(p)$. Otherwise, the sign is the same, but the magnitude is reduced:

$$|\mathcal{G}'(p)| = |\mathcal{G}(p)| - \lambda < |\mathcal{G}(p)|$$

□

Theorem 2. *Let φ be a formula expressed in CNF: $\varphi = \bigwedge_{i=1}^{m} c_i$, where c_i represent the i^{th} clause, and $\mathcal{B}$ a boolean interpretation. If φ is not satisfied ($\mathcal{B}(\varphi) = -1$), the active path passes through an unsatisfied clause.*

Proof. First, note that φ is a conjunction of clauses. As a consequence, $\mathcal{G}(\varphi) = \min_i \mathcal{G}(c_i)$, and $\frac{\partial \varphi}{\partial c_k} \neq 0 \iff k = \arg\min_i \langle \mathcal{G}(c_i) \rangle$. Let c_k be a satisfied clause: $\mathcal{G}(c_k) > 0$. Since formula φ is not satisfied, there must be at least an unsatisfied clause c_j ($\mathcal{G}(c_j) < 0$). Then: $\mathcal{G}(c_j) < 0 < \mathcal{G}(c_k)$, and it must hold $\frac{\partial \varphi}{\partial c_k} = 0$.

□

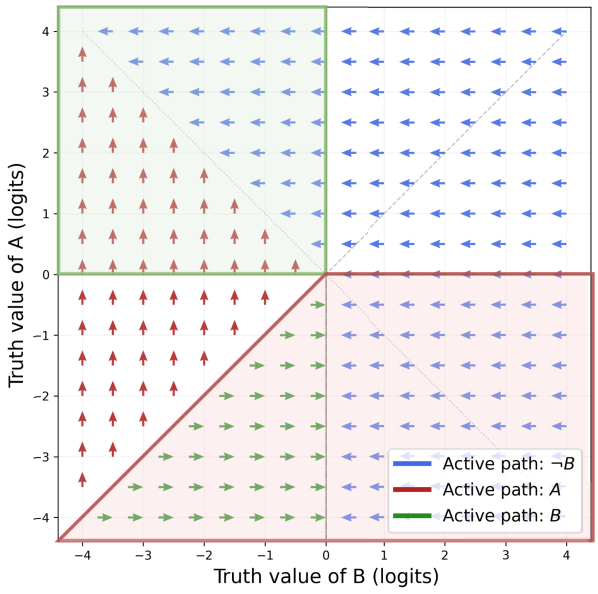

Figure 3: **Vector field of the gradient of a Gödel interpretation** Formula $\varphi = (A \vee B) \wedge \neg B$. Green zone: it corresponds to φ being satisfied ($A \wedge \neg B$); red zone: gradients points on opposite directions, forming a cycle between two unsatisfied interpretations: $\neg A \wedge \neg B$ and $\neg A \wedge B$.

6 The Gödel Trick

In the previous section, we showed that GBOG mimics the behaviour of deterministic local search algorithms. When applied to a CNF, both methods iteratively select an unsatisfied clause, flipping the truth value of one of its literals. The difference is in the selection heuristic: instead of relying on the boolean values, GBOG selects the variable based on continuous truth values given by the Gödel interpretation.

However, much like LSAs converge to local minima, GBOG encounters analogous challenges. In the continuous Gödel landscape, local maxima may emerge at the decision boundaries. This occurs when the gradients from different clauses point towards the threshold from opposite sides; since the threshold itself is excluded and never exactly reached, the variable is forced to continuously cross it. As the gradient flips its direction at each crossing, the system enters an oscillation. Consequently, what acts as a local optimum in the continuous space manifests as a *cycle* between distinct discrete interpretations on opposite sides of the boundary.

This behavior is illustrated in Figure 3 for the formula $\varphi = (A \vee B) \wedge \neg B$. When $\mathcal{G}(A) < 0 < \mathcal{G}(B)$, the gradient points in the direction of $\neg B$. Conversely, when $\mathcal{G}(A) < \mathcal{G}(B) < 0$, the gradient points towards B. The result is a cycle between two unsatisfied discrete interpretations ($\neg A \wedge \neg B$ and $\neg A \wedge B$).

6.1 The Gödel Trick

The **Gödel Trick** (GT) introduces a controlled perturbation to the Gödel interpretation of the propositions, enabling stochastic exploration of the solution space to escape local optima. Formally, we define the perturbed interpretation of a proposition p as: $\mathcal{G}_\epsilon(p) = \mathcal{G}(p) + \epsilon$, where ϵ is a noise term preventing cycles. For a general formula, we use the recursive definition of Gödel interpretation as defined in Section 2.

The Gödel Trick is a reparameterization trick (Kingma and Welling 2014), where the expected value of the gradient of a formula φ corresponds to the gradient of the expected value of φ: $\mathbb{E}_\epsilon\left[\nabla_{\mathcal{G}(p)}\mathcal{G}_\epsilon(\varphi)\right] = \nabla_{\mathcal{G}(p)}\mathbb{E}_\epsilon\left[\mathcal{G}_\epsilon(\varphi)\right]$. This can be derived from standard results on pathwise gradient estimators (Mohamed et al. 2020).

6.2 Gödel Trick as Approximate Probabilistic Inference

In Section 4, we proved that the sign function s is an homomorphism that maps a Gödel interpretation to a discrete implicit one. Similarly, a continuous probability distribution over Gödel interpretation can be mapped to an *implicit discrete distribution* over Boolean interpretations: $\mathcal{B}_\epsilon(\varphi) = s(\mathcal{G}_\epsilon(\varphi))$.

For a proposition p, the probability $P\big(\mathcal{B}_\epsilon(p)\big)$ of p being true under the implicit distribution $\mathcal{B}_\epsilon$ is:

$$P\big(\mathcal{B}_\epsilon(p)\big) = P\big(\mathcal{G}_\epsilon(p) > 0\big) = P\big(\mathcal{G}(p) + \epsilon > 0\big) \quad (9)$$

$$= 1 - F_\epsilon(-\mathcal{G}(p)) = \theta_\epsilon(\mathcal{G}(p)). \quad (10)$$

where F_ϵ is the cumulative distribution function (CDF) of ϵ, and $\theta_\epsilon(x) = 1 - F_\epsilon(-x)$ is the function that maps the unperturbed truth value $\mathcal{G}(p)$ of a proposition p, to the probability of p being true according to $\mathcal{B}_\epsilon$.

We next establish a connection between the implicit distribution $\mathcal{B}_\epsilon$ and probabilistic inference. In this context, the probability of a formula being true is equivalent to its Weighted Model Count (WMC).

Definition 3 (Probabilistic logic)**.** *Let φ be a formula defined over some propositions p_i. Let π_i be the probability associated with proposition p_i. Then, the probability associated to φ under probabilistic logic is defined as:*

$$P(\varphi) = \sum_{\boldsymbol{\omega} \in \{-1,1\}^n} H\big(\mathcal{B}(\varphi; \boldsymbol{\omega})\big) P(\boldsymbol{\omega}) \quad (11)$$

$$P(\boldsymbol{\omega}) = \prod_{i=1}^{n} \pi_i^{H(\omega_i)} (1 - \pi_i)^{1 - H(\omega_i)} \quad (12)$$

where H is the Heaviside function that maps positive values to one, and negative values to zero: $H(x) = \mathbf{1}(x > 0)$.

The connection between GT and probabilistic logic is given by the following theorem.

Theorem 3. *Let φ be a formula defined over some propositions p_i. Let φ be interpreted under probabilistic logic, with π_i the probability associated with proposition p_i, and let ϵ be a continuous distribution over the real numbers.*

If we define the proposition's truth values as $\mathcal{G}(p_i) = \theta_\epsilon^{-1}(\pi_i)$, then: $P\big(\mathcal{B}_\epsilon(\varphi)\big) = P(\varphi)$, where $P(\varphi)$ is the probability of φ being true under the probabilistic interpretation.

Proof. We first consider the probability of $\mathcal{B}_\epsilon(\varphi)$ being positive. Such a probability can be defined as the expected value of the formula's truth value, assuming it to be in $\{0, 1\}$ rather

than $\{-1, 1\}$. Let $\boldsymbol{\epsilon}$ be the vector of sampled noise, and $\boldsymbol{\mu}$ the corresponding vector of noisy interpretations:

$$\mu_i = \mathcal{G}_\epsilon(p_i) = \mathcal{G}(p_i) + \epsilon_i$$

Then:

$$P(\mathcal{B}_\epsilon(\varphi)) = \mathbb{E}_{\boldsymbol{\epsilon}}\big[H\big(\mathcal{B}_\epsilon(\varphi)\big)\big] \tag{13}$$
$$= \mathbb{E}_{\boldsymbol{\epsilon}}\big[H\big(s(\mathcal{G}(\varphi;\boldsymbol{\mu}))\big)\big] \tag{14}$$

where H is the Heaviside function, $\mathcal{G}(\varphi;\boldsymbol{\mu})$ is the Gödel interpretation of φ (as defined in Section 2) assuming $\mathcal{G}(p_i) = \mu_i$. By applying Proposition 1:

$$P(\mathcal{B}_\epsilon(\varphi)) = \mathbb{E}_{\boldsymbol{\epsilon}}\big[H\big(s(\mathcal{G}(\varphi;\boldsymbol{\mu}))\big)\big] \tag{15}$$
$$= \mathbb{E}_{\boldsymbol{\epsilon}}\big[H\big(\mathcal{B}(\varphi;s(\boldsymbol{\mu}))\big)\big] \tag{16}$$
$$= \mathbb{E}_{\boldsymbol{\gamma}}\big[H\big(\mathcal{B}(\varphi;\boldsymbol{\gamma})\big)\big] \tag{17}$$

where in (17) we applied a change of variable on the expectation by defining $\gamma_i = s(\mu_i)$. Note that $\boldsymbol{\gamma} \in \{-1, 1\}^n$, and the expected value can be represented as a summation:

$$P(\mathcal{B}_\epsilon(\varphi)) = \sum_{\boldsymbol{\gamma}\in\{-1,1\}^n} H\big(\mathcal{B}(\varphi;\boldsymbol{\gamma})\big)P(\boldsymbol{\gamma}) \tag{18}$$

with

$$P(\boldsymbol{\gamma}) = \prod_{i=1}^{n} P(\gamma_i)^{H(\gamma_i)}(1 - P(\gamma_i))^{1-H(\gamma_i)} \tag{19}$$

Equations (19) and (12) are identical, except for the usage of $P(\gamma_i)$ instead of π_i. We need to prove the equivalence between these two probabilities:

$$P(\gamma_i) = P(s(\mu_i)) \tag{20}$$
$$= P(\mathcal{G}(p_i) + \epsilon_i > 0) \tag{21}$$
$$= \theta_\epsilon(\mathcal{G}(p_i)) \tag{22}$$
$$= \theta_\epsilon(\theta_\epsilon^{-1}(\pi_i)) = \pi_i \tag{23}$$

where (21) is obtained by applying the definition of μ_i, (22) derives from (10), and (23) comes from the definition of $\mathcal{G}(p_i)$ in theorem statement.

As a consequence:

$$P(\mathcal{B}_\epsilon(\varphi)) = \sum_{\boldsymbol{\gamma}\in\{-1,1\}^n} H\big(\mathcal{B}(\varphi;\boldsymbol{\gamma})\big)P(\boldsymbol{\gamma}) \tag{24}$$
$$= \sum_{\boldsymbol{\omega}\in\{-1,1\}^n} H\big(\mathcal{B}(\varphi;\boldsymbol{\omega})\big)P(\boldsymbol{\omega}) = P(\varphi) \tag{25}$$

□

Theorem 3 formally connects GT with probabilistic inference by showing that it acts as a Monte Carlo estimator for WMC. This result provides a theoretical justification for our approach and establishes a basis for future research directions (see Section 11). However, Theorem 3 does not imply that GT is also an estimator of the WMC gradient; the specific nature of its optimization dynamics, which remains rooted in local search, is further clarified in the following remark.

Remark 2. *Unlike* GBOG*,* GT *optimizes a probability distribution over assignments rather than a single state. Therefore, it can not be properly framed as a LSA. Nonetheless, the local search intuition persists: when a sampled solution violates a clause, the gradient penalizes the responsible literal's truth value within the distribution parameters. Effectively, this performs literal flips in expectation rather than deterministically.*

6.3 The Choice of the Noise Distribution

The noise ϵ plays a crucial role in the GT. We consider two options: logistic and uniform distributions.

Logistic distribution. For a standard logistic distribution, the corresponding CDF is the sigmoid function $F_\epsilon(x) = \sigma(x) = \frac{1}{1+e^{-x}}$, hence:

$$\theta_\epsilon(\mathcal{G}(p)) = 1 - F_\epsilon(-\mathcal{G}(p)) \tag{26}$$
$$= 1 - \sigma(-\mathcal{G}(p)) = \sigma(\mathcal{G}(p)) \tag{27}$$

Thus, in the logistic noise case, $\mathcal{G}(p)$ can be interpreted as the logit of the probability of p being true, establishing a direct link between the GT and probabilistic inference. Moreover, since the logistic distribution can be expressed as the difference of two Gumbel distributions, one can establish a connection between the Gödel Trick and the Gumbel-Max Trick (Gumbel 1954), a common reparameterization method for estimating expected values in discrete settings (more details in Section 8).

Uniform distribution. The probability mass of the logistic distribution is concentrated close to the center, potentially reducing the exploration effect of the noise. Therefore, we also consider the uniform distribution, which is more spread out, potentially increasing the exploration of the solution space.

Note that the probabilistic interpretation of GT holds for other distributions than the logistic, as highlighted by Equation 10. However, we can no longer directly interpret $\mathcal{G}(p)$ as the logits of $P\big(p\big)$ as it induces a different discrete distribution. Specifically, if $\epsilon \sim \mathcal{U}(a, b)$:

$$\theta_\epsilon(x) = \begin{cases} 0 & \text{if } x < -b \\ \frac{x+b}{b-a} & \text{if } x \in [-b, -a] \\ 1 & \text{if } x > -a. \end{cases} \tag{28}$$

Given a probability π of a proposition p, the unperturbed truth value is $\theta_\epsilon^{-1}(\pi) = \pi \cdot (b - a) - b$.

7 Gödel Trick with Categorical Variables

The Gödel Trick proposed in Section 6 is defined exclusively on boolean variables. Despite its generality, in the context of NeSy, categorical variables are frequently used (see as an example Section 10), for which the Gödel Trick is not directly applicable. Let π_i, with $i \in [1, \dots K]$, be the probabilities associated with K categories, such that $\sum_{i=1}^{K} \pi_i = 1$. We can define a propositional variable p_i for each category, and use the Gödel Trick: $\mathcal{G}(p_i) = \theta_\epsilon^{-1}(\pi_i)$. However, by doing so we are implicitly assuming independence between the propositions. On the contrary, we need to assure that exactly one category is true. To include such a constraint, a possibility is to enforce the satisfaction of the XOR between the K propositions through logical constraints. However, such a solution could require to include a large amount of rules, reducing the efficiency of the method.

We propose a different strategy (see Section 10 for an empirical evaluation): we define a shift function that shifts the perturbed truth values $\mathcal{G}_\epsilon(p_i)$ of propositions p_i in order to enforce the aforementioned constraint:

$$\mathrm{shift}(\mathbf{x}) = \mathbf{x} - \frac{x_i + x_j}{2},$$

where i and j are the indexes of the highest and second highest elements of the vector $\mathbf{x}$, respectively. Note that for any vector $\mathbf{x}$, the shifted vector $\bar{\mathbf{x}} = \mathrm{shift}(\mathbf{x})$ contains exactly one value larger than zero:

$$\bar{x}_i = x_i - \frac{x_i + x_j}{2} = \frac{x_i - x_j}{2} > 0 \tag{29}$$

$$\bar{x}_j = x_j - \frac{x_i + x_j}{2} = \frac{x_j - x_i}{2} < 0 \tag{30}$$

$$\bar{x}_k \leq \bar{x_j} < 0 \quad \forall k \neq i \tag{31}$$

It is worth noting two properties of the shifted vector: first, the highest element of $\bar{\mathbf{x}}$ is the only one higher than zero; second, the second-highest element is the negation of the first: $\bar{x}_j = -\bar{x}_i$.

We define the vector $\mathcal{G}_\epsilon(\mathbf{p})$ of perturbed truth values as:

$$\mathcal{G}_\epsilon(\mathbf{p}) = \langle \mathcal{G}_\epsilon(p_1), \mathcal{G}_\epsilon(p_2) \dots \mathcal{G}_\epsilon(p_K) \rangle$$

We also define its shifted version as: $\bar{\mathcal{G}}_\epsilon(\mathbf{p}) = \mathrm{shift}(\mathcal{G}_\epsilon(\mathbf{p}))$. Because of the properties of the shift function, we have: $\bar{\mathcal{G}}_\epsilon(p_i) > 0$ for the highest value i in $\mathcal{G}_\epsilon(\mathbf{p})$, and $\forall k \neq i \;\; \bar{\mathcal{G}}_\epsilon(p_k) < 0$. As a consequence, the constraint of the categorical variable is satisfied (i.e., exactly one value is true). Since the shifted truth values are still used in combination with Gödel Logic:

$$\bar{\mathcal{G}}_\epsilon(p_i) = -\bar{\mathcal{G}}_\epsilon(p_j) = \bar{\mathcal{G}}_\epsilon(\neg p_i) \tag{32}$$

representing the categorical variable as a Bernoulli variable that select one among the two highest values.

Finally, when using the Gumbel distribution to generate noise, the resulting behavior is equivalent to the one of the Gumbel-Max Trick (see Section 8).

8 Connections with Gumbel-Max Trick

The Gumbel-Max Trick is a well-known reparameterization approach for sampling from a categorical distribution (Gumbel 1954; Jang, Gu, and Poole 2022). Given a categorical distribution defined by probabilities π_i, the trick allows for an efficient way to sample from this distribution by adding continuous noise to its logits.

Starting from the softmax logits $\mathbf{z} = \langle z_1, z_2 \dots z_K \rangle$ of the probabilities π_i, one can add noise sampled from a standard Gumbel distribution to the logits, and the distribution of the argmax of the resulting vector corresponds to the original categorical distribution. Consider the following equation:

$$\mathbf{x} = \mathrm{onehot}(\arg\max_i(z_i + \epsilon_i)) \tag{33}$$

with $\epsilon_i \sim \mathrm{Gumbel}(0, 1)$. The key aspect of the Gumbel-Max Trick is that the probability that $\arg\max_i(z_i + \epsilon_i) = j$ is equivalent to π_j.

Now, consider replacing the $\arg\max$ function with the shift function, and applying the sign function s to the resulting vector:

$$\mathbf{v} = s\big(\mathrm{shift}\big(\langle z_i + \epsilon_i \rangle\big)\big) \tag{34}$$

with the noise still being sampled from a standard Gumbel distribution. As proven in section 7, the shift function translates the values of the vector in such a way that only the highest is greater than zero. As a consequence, the sign function s maps all the values to -1, except for the $\arg\max$, which is mapped to $+1$.

Note that, in our framework, we defined $\top$ to be $+1$, and $\bot$ to be -1, different from the context of the Gumbel-Max Trick, where $\bot$ is defined as 0. Therefore, vector $\mathbf{v}$ can be seen as a rescaled version of $\mathbf{x}$ suitable for the application of the Gödel Trick:

$$\mathbf{v} = 2\mathbf{x} - 1 \tag{35}$$

9 Experiments on SAT

In this section, we evaluate the performance of the GT method on SAT problems to validate our theoretical findings. SAT problems contain strong dependencies among clauses, which often cause fuzzy logics to get stuck in local optima. This makes them a natural stress test for GT, which is specifically designed to escape such local optima.

We focus on satisfiable instances from the SATLIB library (Hoos and Stützle 2000), since GT is an incomplete solver. Note that most problems in SATLIB are satisfiable, and many SAT solvers are themselves incomplete.

Setup. SATLIB is a well-established repository of SAT benchmarks, organized into collections representing different domains. For example, the UF family contains random 3-SAT problems, while the Planning set includes SAT-encoded planning instances.

We compare GT against three fuzzy semantics: Product, Łukasiewicz, and Gödel logics. These are natural baselines, as GT can be applied in the same contexts and it is proposed as an alternative in the NeSy context.

Methodology. Initially, we conducted a grid search to optimise learning rate and momentum. This search was performed separately for each method on the simplest benchmark of the UF collection: `uf20-91`. For the noise, we used the standard logistic and uniform distributions, since our preliminary experiments showed that GT is not particularly sensitive to the choice of variance. We ran these starting experiments on an NVIDIA RTX 2080Ti with 11GB of RAM.

Among the baselines, Product logic performed best in the preliminary `uf20-91` experiments, followed by Gödel, whereas Łukasiewicz did not solve any instance within the optimization budget. We therefore excluded Łukasiewicz from the remaining SATLIB benchmarks and report the extended evaluation for the other methods in Table 1. The table reports the average performance across benchmark collections.

Given that each sample can be run in parallel, we explore multiple initial interpretations efficiently. For each problem,

DOMAIN	PRODUCT LOGIC		GÖDEL LOGIC		GT LOGISTIC		GT UNIFORM	
	S(%)	B(%)	S(%)	B(%)	S(%)	B(%)	S(%)	B(%)
UF	2.9 ± 7.4	6.6 ± 15.0	0.9 ± 2.4	11.5 ± 29.6	25.0 ± 25.8	57.5 ± 24.3	$\mathbf{74.5 \pm 12.8}$	$\mathbf{99.4 \pm 0.7}$
RTI/BMS	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	8.3 ± 4.2	28.3 ± 12.3	$\mathbf{46.2 \pm 23.4}$	$\mathbf{95.6 \pm 4.4}$
CBS	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	6.7 ± 10.6	24.8 ± 25.1	$\mathbf{70.3 \pm 19.3}$	$\mathbf{100.0 \pm 0.1}$
FLAT	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.2 ± 2.4	5.4 ± 10.7	20.3 ± 39.9	$\mathbf{41.8 \pm 34.8}$	$\mathbf{80.2 \pm 39.6}$
SW	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{30.2 \pm 44.7}$	$\mathbf{38.0 \pm 46.9}$
PLANNING	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	6.8 ± 6.8	21.4 ± 21.4	$\mathbf{9.9 \pm 9.9}$	$\mathbf{28.6 \pm 28.6}$
AIS	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.8 ± 0.0	25.0 ± 0.0	$\mathbf{29.8 \pm 0.0}$	$\mathbf{75.0 \pm 0.0}$
QG	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{0.4 \pm 0.0}$	$\mathbf{10.0 \pm 0.0}$	0.0 ± 0.0	0.0 ± 0.0
BMC	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
DIMACS	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{1.4 \pm 2.3}$	$\mathbf{27.4 \pm 35.9}$
BEIJING	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	5.9 ± 0.0	$\mathbf{15.4 \pm 0.0}$	$\mathbf{14.8 \pm 0.0}$	$\mathbf{15.4 \pm 0.0}$

Table 1: Comparative table showing results on SATLIB benchmarks for four methods: Product Logic, Gödel Logic, Gödel Trick Logistic, and Gödel Trick Uniform. Columns: S stands for Sample Solved (mean ± std of number of solved samples), B stands for Best Solution (mean ± std of number of problem solved by keeping the best results among the samples). Highest S and B in bold.

we evaluated 100 samples (equivalent to 100 independent runs) and computed two metrics: *S*, the percentage of successfully solved instances across all samples, and *B*, the number of solved problems considering the best result among the 100 samples for each problem. Note that each sample is computed in parallel on the GPU, and memory usage remains minimal. Consequently, the *B* measure can be further improved by increasing the number of samples.

Results. The results clearly demonstrate that GT yields a substantial improvement over all baselines, with the Uniform variant achieving the best performance. These results align with our theoretical analysis: while Gödel logic provides the correct gradient direction for a discrete flip, it lacks a mechanism to escape local optima in complex combinatorial landscapes. The noise introduced in GT-based models facilitates this exploration, allowing the solver to "jump" between different regions of the Boolean hypercube.

The results also show an advantage of GT over Product logic. The latter is often viewed as an approximation of probabilistic logic that assumes independence between clauses. In contrast, while GT also possesses a probabilistic interpretation (see Theorem 3), it does not make such an assumption. This property explains the difference in performances on SAT problems, which are characterized by tight variable dependencies.

Despite these gains, certain benchmarks like BMC and QG remain challenging for GT. These results suggest that, while GT provides a powerful differentiable alternative to traditional fuzzy logics, very large-scale combinatorial problems may still require the integration of more advanced search heuristics, which we leave for future investigation.

10 Visual Sudoku

In addition to SAT problems, we evaluated our framework on the Visual Sudoku task (Augustine et al. 2022), a NeSy benchmark where the goal is to classify the validity of a 9×9 grid represented by MNIST images, where the only available supervision is the global validity of the board.

Method	Accuracy (%)	Avg. Time (m)
CNN (Perception)	51.20 ± 2.20	-
NeuPSL	51.50 ± 1.37	-
A-NeSI	62.25 ± 2.20	20.3
Gödel Logic (Det.)	61.19 ± 1.61	20.5
Gödel Trick (GT)	$\mathbf{62.95 \pm 1.42}$	**8.5**

Table 2: Comparisons on the 9×9 Visual Sudoku task.

The architecture stacks a GT layer on top of a neural perception network. Specifically, a CNN maps each MNIST image in the grid to propositions $p_{i,k}$, where i and k denote the cell index and the digit, respectively. These outputs are then processed by the GT-based reasoner to evaluate the satisfaction of the Sudoku rules. The logical constraints enforce that no two cells i and j in the same row, column, or 3×3 block can contain the same digit k, which we encode via the formula $\phi_{i,j,k} = \neg p_{i,k} \vee \neg p_{j,k}$.

We compare our approach with a CNN, NeuPSL (Pryor et al. 2022), and A-NeSI (van Krieken et al. 2023). [5] The experiments were conducted on a machine equipped with an NVIDIA GTX 3070 with 12GB RAM. For the perception neural network, we use the same architecture as A-NeSI. For the Gödel logic, we implement the interpretation in the log-space to ensure numerical stability. For GT, we apply the categorical $shift$ function described in Section 7. In both cases, training is performed in two phases: we first optimize clauses independently to provide a denser gradient signal, then aggregate them into the full Sudoku formula. A comprehensive description of the architecture and evaluation settings is provided in the Supplementary Material.

Results in Table 2 show that both Gödel-based approaches reach accuracies statistically equivalent to A-NeSI. It is worth noting that, to correctly classify a valid Sudoku instance, the model must simultaneously predict all 81 images correctly.

[5] We use the baseline results reported by (van Krieken et al. 2023) for the CNN and NeuPSL. We run A-NeSI on our machine to compare execution time.

As a consequence, even a 0.994 per-digit accuracy limits the expected grid accuracy to $0.994^{81} \approx 0.61$.

Notably, the deterministic Gödel Logic is the slowest method. In contrast, GT is more than twice as fast, demonstrating a significant computational advantage. This efficiency stems from the mutual exclusivity constraints: while GT uses the extremely efficient $shift$ function to enforce this constraint, in Gödel logic we cannot utilize such a strategy, requiring more expensive operations to maintain numerical stability while enforcing the constraint.

Overall, these results validate our theoretical framework and its applicability in combination with neural networks, demonstrating that Gödel logic and its variants are viable tools for neurosymbolic domains.

11 Conclusions and Future Work

In this work, we challenged the view of Gödel logic as a mere continuous relaxation by proving its homomorphism to classical logic semantics. Moreover, thanks to its gradient sparsity, we proved that Gödel optimization formally instantiates a discrete local search for satisfiability.

To address the local optima inherent in this deterministic search, we introduced the Gödel Trick (GT), a stochastic reparameterization designed for better exploration. Beyond its empirical success, GT establishes a formal theoretical bridge between fuzzy optimization and probabilistic inference, acting as a Monte Carlo estimator for WMC.

Future work will focus on extending GT with optimizations inspired by the local search algorithms literature, for instance with Tabu Search, to further improve convergence and robustness. Additionally, we plan to evaluate GT on more complex NeSy tasks and experiment with alternative noise distributions beyond Logistic and Uniform to understand their effect on performance and generalization.

Another promising direction involves integrating GT with generative models, exploiting the connections with probabilistic inference (see Theorem 3). By interpreting unperturbed truth values as negative energies in a Gibbs distribution, GT could be used to combine multiple Energy-Based Models (EBMs), providing a differentiable mechanism to enforce logical constraint satisfaction on generated samples.

Finally, we aim to investigate the integration of the Gödel Trick into models capable of learning knowledge, such as DSL (Daniele et al. 2023a), to assess its potential in end-to-end neurosymbolic learning frameworks.

11.1 Limitations

While gradient sparsity is the very reason GBOG mimics LSA, it may also reduce convergence speed compared to dense-gradient logic. Furthermore, GT remains susceptible to cycles and local optima, lacking the global deductive capabilities of modern SAT solvers. Finally, GT inherits common NeSy challenges, such as the independence assumption between propositions (van Krieken et al. 2024), which can cause the optimization to collapse into a single deterministic solution in discriminative learning settings, thereby losing the ability to model uncertainty. Additionally, GT remains susceptible to reasoning shortcuts (Marconato et al. 2024b; 2024a).

Acknowledgments

We would like to express our gratitude to Samy Badreddine for the insightful discussions and valuable feedback, which had a significant impact on this work. We also thank Samuel Cognolato and Davide Bizzaro for their helpful comments and suggestions.

AI Declaration

LLMs have been used during the preparation of this manuscript exclusively as an editing tool. Additionally, they have been used for creating the python code that generated the plots of Figure 1(c), Figure 1(d), and Figure 3. Finally, they have been used for the python script that converts the results of SAT experiments into the latex code for Table 1, and for the longer version of the table in the Supplementary Materials. All generated content have been checked by the authors.

References

Ahmed, K.; Teso, S.; Chang, K.-W.; Van den Broeck, G.; and Vergari, A. 2022. Semantic probabilistic layers for neuro-symbolic learning. *Advances in Neural Information Processing Systems* 35:29944–29959.

Andreoni, R.; Buliga, A.; Daniele, A.; Ghidini, C.; Montali, M.; and Ronzani, M. 2025. T-ilr: a neurosymbolic integration for ltlf. *arXiv preprint arXiv:2508.15943*.

Augustine, E.; Pryor, C.; Dickens, C.; Pujara, J.; Wang, W.; and Getoor, L. 2022. Visual sudoku puzzle classification: A suite of collective neuro-symbolic tasks. In *International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*.

Badreddine, S.; Garcez, A. d.; Serafini, L.; and Spranger, M. 2022. Logic tensor networks. *Artificial Intelligence* 303:103649.

Chavira, M., and Darwiche, A. 2008. On probabilistic inference by weighted model counting. *Artificial Intelligence* 172(6-7):772–799.

Choi, Y.; Vergari, A.; and Van den Broeck, G. 2020. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf* 6.

Daniele, A., and Serafini, L. 2019. Knowledge enhanced neural networks. In *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part I 16*, 542–554. Springer.

Daniele, A.; Campari, T.; Malhotra, S.; and Serafini, L. 2023a. Deep symbolic learning: discovering symbols and rules from perceptions. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 3597–3605.

Daniele, A.; van Krieken, E.; Serafini, L.; and van Harmelen, F. 2023b. Refining neural network predictions using background knowledge. *Machine Learning* 112(9):3293–3331.

De Smet, L.; Sansone, E.; and Zuidberg Dos Martires, P. 2023. Differentiable sampling of categorical distributions

using the CatLog-derivative trick. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 30416–30428. Curran Associates, Inc.

Diligenti, M.; Gori, M.; and Sacca, C. 2017. Semantic-based regularization for learning and inference. *Artificial Intelligence* 244:143–165.

Feldstein, J.; Dilkas, P.; Belle, V.; and Tsamoura, E. 2024. Mapping the neuro-symbolic ai landscape by architectures: A handbook on augmenting deep learning through symbolic reasoning.

Flinkow, T.; Pearlmutter, B. A.; and Monahan, R. 2024. Comparing differentiable logics for learning with logical constraints. *arXiv preprint arXiv:2407.03847*.

Giunchiglia, E.; Tatomir, A.; Stoian, M. C.; and Lukasiewicz, T. 2024. Ccn+: A neuro-symbolic framework for deep learning with requirements. *International Journal of Approximate Reasoning* 109124.

Grespan, M. M.; Gupta, A.; and Srikumar, V. 2021. Evaluating relaxations of logic for neural networks: A comprehensive study. *arXiv preprint arXiv:2107.13646*.

Gumbel, E. J. 1954. Statistical theory of extreme valuse and some practical applications. *Nat. Bur. Standards Appl. Math. Ser. 33*.

Gupta, M. M., and Qi, J. 1991. Theory of t-norms and fuzzy inference methods. *Fuzzy sets and systems* 40(3):431–450.

Hoos, H. H., and Stützle, T. 2000. Satlib: An online resource for research on sat. *Sat* 2000:283–292.

Jang, E.; Gu, S.; and Poole, B. 2022. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Kalman, J. A. 1958. Lattices with involution. *Transactions of the American Mathematical Society* 87(2):485–491.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In Bengio, Y., and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kisa, D.; Van den Broeck, G.; Choi, A.; and Darwiche, A. 2014. Probabilistic sentential decision diagrams. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Li, Z., and Si, X. 2022. Nsnet: A general neural probabilistic framework for satisfiability problems. *Advances in Neural Information Processing Systems* 35:25573–25585.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems* 31.

Marconato, E.; Bortolotti, S.; van Krieken, E.; Vergari, A.; Passerini, A.; and Teso, S. 2024a. Bears make neuro-symbolic models aware of their reasoning shortcuts. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

Marconato, E.; Teso, S.; Vergari, A.; and Passerini, A. 2024b. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. *Advances in Neural Information Processing Systems* 36.

Marra, G.; Dumančić, S.; Manhaeve, R.; and De Raedt, L. 2024. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence* 328:104062.

Mohamed, S.; Rosca, M.; Figurnov, M.; and Mnih, A. 2020. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research* 21(132):1–62.

Moisil, G. C. 1935. Recherches sur l'algebre de la logique. *Ann. Sci. Univ. Jassy* 22(3):1–117.

Pryor, C.; Dickens, C.; Augustine, E.; Albalak, A.; Wang, W.; and Getoor, L. 2022. Neupsl: Neural probabilistic soft logic. *arXiv preprint arXiv:2205.14268*.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature* 323(6088):533–536.

Selsam, D.; Lamm, M.; Bünz, B.; Liang, P.; de Moura, L.; and Dill, D. L. 2018. Learning a sat solver from single-bit supervision. *arXiv preprint arXiv:1802.03685*.

Slusarz, N.; Komendantskaya, E.; Daggitt, M. L.; Stewart, R.; and Stark, K. 2023. Logic of differentiable logics: Towards a uniform semantics of dl. In *Proceedings of 24th International Conference on Logic*, volume 94, 473–493.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *siam Journal on Computing* 8(3):410–421.

van Krieken, E.; Acar, E.; and van Harmelen, F. 2022. Analyzing differentiable fuzzy logic operators. *Artificial Intelligence* 302:103602.

van Krieken, E.; Thanapalasingam, T.; Tomczak, J.; Van Harmelen, F.; and Ten Teije, A. 2023. A-nesi: A scalable approximate method for probabilistic neurosymbolic inference. *Advances in Neural Information Processing Systems* 36:24586–24609.

van Krieken, E.; Minervini, P.; Ponti, E.; and Vergari, A. 2024. On the independence assumption in neurosymbolic learning. In *Forty-first International Conference on Machine Learning*.

Wang, P.-W.; Donti, P. L.; Wilder, B.; and Kolter, J. Z. 2019. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International conference on machine learning*, 6545–6554. PMLR.

Xu, J.; Zhang, Z.; Friedman, T.; Liang, Y.; and Broeck, G. 2018. A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, 5502–5511. PMLR.