Proceedings of Machine Learning Research vol vvv:1–25, 2026

Modal Logical Neural Networks

Antonin Sulc ASULC@LBL.GOV
Lawrence Berkeley National Lab
Berkeley, CA, USA

Abstract

We introduce Modal Logical Neural Networks (MLNNs), the first framework enabling differentiable reasoning over necessity and possibility by integrating neural networks with Kripke semantics from modal logic. Specifically, we instantiate the operators $\Box$ (necessity) and $\Diamond$ (possibility) as specialized neurons that aggregate truth values across possible worlds, allowing the model to function as a differentiable "logical guardrail."

The proposed architecture is highly flexible: the accessibility relation between worlds can either be fixed by the user to enforce known rules or learned via a parameterized neural network. This allows the model to simultaneously perform deductive reasoning within a logical structure while inductively learning that structure from data. The entire framework is differentiable end-to-end, with learning driven by minimizing a logical contradiction loss.

We demonstrate MLNNs on five case studies: (1) enforcing grammatical constraints over statistical priors, (2) detecting out-of-distribution inputs via axiomatic reasoning, (3) learning multi-agent trust in negotiation, (4) identifying deception through temporal consistency checking, and (5) solving combinatorial constraint satisfaction problems. These experiments show how modal structure, whether fixed or learned, increases logical consistency and interpretability without changing underlying task architectures.

Keywords: Modal Logic, Neurosymbolic AI, Kripke Semantics, Differentiable Reasoning, Epistemic Logic

1. Introduction

Modern neural networks, particularly large language models, have achieved remarkable success in learning complex statistical patterns from vast datasets. However, their reliance on purely data-driven learning presents a critical challenge in high-stakes environments. These models can produce outputs that are statistically plausible yet logically incoherent, factually incorrect, or in violation of fundamental domain constraints. This unpredictability hinders deployment in safety-critical applications such as autonomous systems, medical diagnostics, or legal reasoning, where adherence to explicit rules and principles is not just desirable, but essential. The core of this problem is a methodological gap: a lack of a native mechanism within these architectures to enforce declarative, symbolic knowledge and guarantee that outputs conform to a set of verifiable logical rules.

This paper addresses[1] this gap by turning to modal logic, a framework that extends classical reasoning by evaluating truths across a structure of "possible worlds" rather than a single fixed reality. This relies on Kripke semantics, where an accessibility relation determines which worlds interact. This structure is critical for robust AI: in a safety context, a temporal rule like $\Box(\neg\text{moving} \wedge \text{red\_light})$ uses the accessibility relation to treat future states as "reachable worlds," allowing us to enforce that a safety constraint holds necessarily across an entire trajectory. Complementing this,

1. Code available https://github.com/sulcantonin/MLNN_public

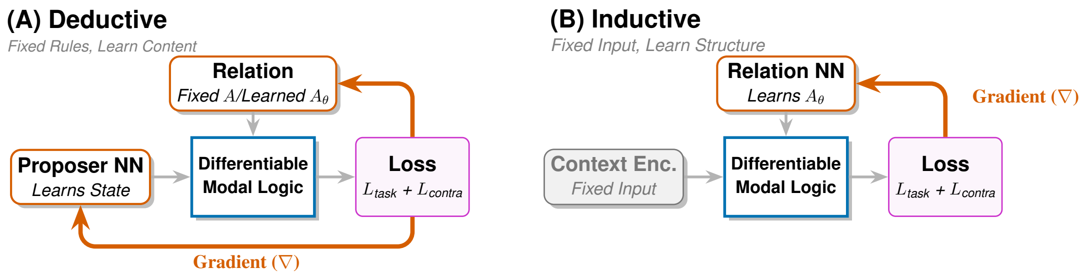

Figure 1: MLNN learning modes executing the Upward-Downward inference algorithm. (A) Deductive: Enforces fixed axioms by updating state representations (Proposer NN or Learnable Logits for CSPs) (B) Inductive: Discovers relational structure (A_θ) by updating the Relation NN. Gray arrows denote forward inference; orange arrows denote gradient flow minimizing logical contradiction.

the possibility operator allows us to enforce liveness goals, such as $\Diamond$(task_complete), ensuring that a system does not trivially satisfy safety by freezing forever, but necessarily reaches a success state in at least one accessible future.

Furthermore, for agentic systems, we utilize Epistemic (knowledge) or Doxastic (belief) logics to reason about subjective states. While $K_a\phi$ asserts verified knowledge, a doxastic operator $B_a\phi$ allows us to reason about beliefs that may differ from reality. For instance, modeling a scenario where $B_{\text{agent}}(\text{safe}) \land \neg\text{safe}$ allows the system to explicitly detect false confidence or hallucinations, a distinction impossible in standard logic where a proposition is simply true or false. Here, the accessibility relation maps the topology of trust and information: $B_a\phi$ holds only if ϕ is true in all worlds *accessible* to agent a's current perception. This explicitly models nuances like deception while ensuring explainability: we can inspect the learned accessibility relation to understand *why* an agent believes a statement or why a future is deemed safe, rather than relying on opaque feature correlations. While these logics offer the formal language to govern such real-world dynamics, integrating their relational structures into differentiable, learnable models remains an open challenge.

We introduce Modal Logical Neural Networks (MLNNs), a neurosymbolic framework extending Logical Neural Networks (LNNs) Riegel et al. (2020) with Kripke semantics to reason over possible worlds. By parameterizing the accessibility relation with a neural network, MLNNs uniquely enable simultaneous deductive reasoning from axioms and inductive learning of relational structures, see Figure 1. We demonstrate these dual capabilities empirically: the model inductively discovers trust networks in multi-agent systems, and deductively solves complex constraints like the “AI Escargot” Sudoku, all by minimizing logical conflicts rather than training on supervised labels.

2. Related Work

The integration of logic and neural networks has a rich history spanning neurosymbolic AI. Logical Neural Networks (LNNs) Riegel et al. (2020) introduced weighted real-valued logic with provable bounds, while Markov Logic Networks Richardson and Domingos (2006) and ProbLog De Raedt et al. (2007) combined first-order logic with probabilistic reasoning. Logic Tensor Networks Serafini and d'Avila Garcez (2016) embedded logical knowledge into tensor spaces for end-to-end learning.

Modal logic in neural systems has received less attention. Connectionist Modal Logic Garcez et al. (2007) represented modal operators in recurrent networks but lacked differentiable world structures. Recent work on temporal logic satisfiability Luo et al. (2022) and epistemic logic in reinforcement learning Engesser et al. (2025) has made progress, but these approaches do not provide a unified framework for learning accessibility relations.

Our contribution is a differentiable implementation of full Kripke semantics with learnable accessibility, enabling both deductive and inductive modal reasoning in a single architecture.

3. Method: Modal Logical Neural Networks

3.1. Preliminaries and Notation

The power of modal logic stems from Kripke semantics, which extends classical logic by evaluating propositions across multiple 'possible worlds'. Formally, a Kripke model is a tuple $M = \langle W, R, V \rangle$, where W is a set of possible worlds, R is a binary accessibility relation on W (i.e., $R \subseteq W \times W$), and V is a valuation function that assigns a truth value to each atomic proposition $p \in P$ in each world $w \in W$. An MLNN instantiates a differentiable, learnable version of this model, where a "world" is a flexible concept representing an agent's belief, a moment in time, or a specific context.

We formalize our framework using the following notation. Let W be a finite set of possible worlds and T be a finite set of discrete time steps, which together define a set of spacetime states $S = W \times T$. Atomic propositions and logical formulae are denoted by p and ϕ, respectively, with their truth values represented by continuous lower and upper bounds $[L, U] \subseteq [0, 1]$. The structural relationships between worlds are defined by a crisp binary accessibility relation $R \subseteq W \times W$ or a learnable, neurally parameterized accessibility matrix A_θ, which may be applied as a masked matrix $\tilde{A}$. We utilize the standard modal operators $\Box$ (necessity) and $\Diamond$ (possibility), which instantiate as $K_a\phi$ ("Agent a knows ϕ") in epistemic logic, $B_a\phi$ ("Agent a believes ϕ") in doxastic logic, or $G\phi$ ("Globally") and $F\phi$ ("Finally") in temporal logic. The soft logic aggregations are controlled by a temperature parameter τ. Finally, training minimizes a standard task loss L_{task} alongside a logical contradiction loss L_{contra}, balanced by β, which strictly optimizes for consistency with the modal axioms.

3.2. Differentiable Kripke Semantics

In an MLNN, the components of a Kripke model are realized as differentiable tensors. For each proposition p, the MLNN stores a tensor of truth bounds of shape $(|W|, 2)$, where each row $[L_{p,w}, U_{p,w}]$ represents the truth bounds of p in world w. The accessibility relation R, which defines which worlds can "see" each other, dictates the function of modal operators. A key feature of the MLNN is the option to make this relation a learnable component, parameterized by a neural network, A_θ. This allows the model to learn the underlying relational physics of the problem domain, such as which agents trust each other or how time flows, directly from data, guided by the goal of achieving logical consistency.

3.2.1. MODAL OPERATORS: THE NECESSITY AND POSSIBILITY NEURONS

The modal operators are specialized neurons that aggregate information across worlds. While classical logic uses hard minimums and maximums over a fixed set of neighbors, we employ differentiable relaxations over the weighted accessibility matrix $\tilde{A}$ to allow gradients to propagate through the structural decision boundaries. To ensure the soundness of our bounds, we define a set of differentiable, monotonic operators. Let $x = \{x_i\}$ be a set of truth values from all worlds. We define $\mathrm{softmin}_\tau(x) = -\tau \log \sum_i \exp(-x_i/\tau)$ as a sound lower bound on $\min(x)$, and $\mathrm{softmax}_\tau(x) = \tau \log \sum_i \exp(x_i/\tau)$ as a sound upper bound on $\max(x)$. We also define $\text{conv-pool}_\tau(x, z) = \sum_i w_i x_i$ where $w_i = \mathrm{softmax}(z_i/\tau)$ as a convex pooling operator. When $z = x$, this provides a lower bound on $\max(x)$, and when $z = -x$, it provides an upper bound on $\min(x)$. We use these to define the modal neuron bounds with default temperature $\tau = 0.1$.

The $\Box$ (Necessity) Neuron In differentiable Kripke semantics, $\Box\phi$ represents a weighted universal quantification: ϕ must be true in all worlds to the degree that they are accessible. We implement this using the differentiable implication $(1 - \tilde{A}_{w,w'} + \text{truth})$. Intuitively, this acts as a "weakest link" detector: if a world is highly accessible ($\tilde{A} \approx 1$) but ϕ is false, the score collapses.

$$\begin{aligned} L_{\Box\phi,w} &= \underset{w' \in W}{\mathrm{softmin}_\tau} \left((1 - \tilde{A}_{w,w'}) + L_{\phi,w'} \right) && \text{(Weighted Universal)} \\ U_{\Box\phi,w} &= \underset{w' \in W}{\text{conv-pool}_\tau} \left((1 - \tilde{A}_{w,w'}) + U_{\phi,w'}, \dots \right) && \text{(Sound Upper Bound)} \end{aligned} \tag{1}$$

The $\Diamond$ (Possibility) Neuron Dually, $\Diamond\phi$ represents a weighted existential quantification: ϕ must be true in at least one highly accessible world. We implement this using a differentiable conjunction. Functionally, this acts as an "evidence scout": the neuron activates if it finds any world that is both accessible and where ϕ is true.

$$\begin{aligned} L_{\Diamond\phi,w} &= \underset{w' \in W}{\text{conv-pool}_\tau} \left(\tilde{A}_{w,w'} + L_{\phi,w'} - 1, \dots \right) && \text{(Sound Lower Bound)} \\ U_{\Diamond\phi,w} &= \underset{w' \in W}{\mathrm{softmax}_\tau} \left(\tilde{A}_{w,w'} + U_{\phi,w'} - 1 \right) && \text{(Weighted Existential)} \end{aligned} \tag{2}$$

This formulation ensures that the bounds respect the fundamental modal duality $\Diamond\phi \equiv \neg\Box\neg\phi$ via the identity $\mathrm{softmax}(x) = 1 - \mathrm{softmin}(1 - x)$.

3.3. Flexible and Learnable Accessibility Relations

A key capability of MLNNs is the ability to treat the accessibility relation as a learnable parameter, rather than a fixed part of the model structure. In classical modal logic, R is a fixed, given structure. In an MLNN, R can be replaced by a learnable, weighted relation A_θ. The flexibility to use either a fixed or learnable structure is crucial; we use fixed logical rules in our grammatical guardrailing and axiomatic detection of the unknown, while the learnable relation A_θ is showcased in our multi-agent epistemic trust analysis.

We parameterize this relation with a neural network, $A_\theta : W \times W \to [0, 1]$. For smaller domains, this can be instantiated as a direct matrix of learnable logits passed through a sigmoid function. For scalable applications, we can employ a metric learning parameterization, where a neural encoder maps each world w to a latent embedding $\mathbf{h}_w \in \mathbb{R}^d$. The accessibility score is then defined by a kernel function, such as $A(w_i, w_j) = \sigma(\mathbf{h}_{w_i}^\top \mathbf{h}_{w_j})$, effectively determining logical access via geometric proximity. This factorized form reduces the parameter space from quadratic to linear with respect to $|W|$.

For differentiability, we use these weights directly in a soft aggregation. The necessity neuron, for example, becomes a weighted soft minimum, see Equation 1. The truth value L_{ϕ,w_j} from a target world w_j is incorporated into the minimum at w_i using a differentiable implication $1-(\tilde{A})_{ij}+L_{\phi,w_j}$. If $(\tilde{A})_{ij} \approx 1$ (full access), the term $(1-(\tilde{A})_{ij})$ is near 0, and L_{ϕ,w_j} fully participates in the minimum. If $(\tilde{A})_{ij} \approx 0$ (no access), the term is near 1, effectively removing L_{ϕ,w_j} from consideration.

The parameters θ of this accessibility network are learnable, updated via gradient descent on the system's overall contradiction loss. This is the core inductive capability of the MLNN. It allows the model to discover the logical structure that best resolves contradictions in the data.

3.4. Loss Function and Training

The MLNN is trained end-to-end by minimizing a combined loss function:

$$L_{\text{total}} = L_{\text{task}} + \beta L_{\text{contra}} \tag{3}$$

where L_{task} is a standard supervised loss (e.g., cross-entropy for classification), and L_{contra} is the logical contradiction loss. The contradiction loss penalizes states where the learned lower bound exceeds the upper bound:

$$L_{\text{contra}} = \sum_{w \in W} \sum_{\phi} \max\left(0, L_{\phi,w} - U_{\phi,w}\right) \tag{4}$$

The hyperparameter β controls the trade-off between task performance and logical consistency. Higher β values enforce stricter logical coherence, while lower values prioritize task accuracy.

Pure Satisfiability Mode For constraint satisfaction tasks like Sudoku (see Section 5.4, where the goal is to find *any* valid configuration that satisfies a set of axioms, we set $L_{\text{task}} = 0$ and drive learning entirely via L_{contra}. In this regime, the MLNN functions as a differentiable energy-based model, where the "Proposer" is a set of free learnable parameters (logits) rather than a conditional network.

4. Theoretical Properties

The theoretical validity of MLNNs is supported by three key guarantees: soundness, convergence, and expressiveness. They are formally proven in Appendix A. Regarding soundness, we demonstrate that the differentiable operators softmin_τ and conv-pool_τ maintain valid bounds, such that $\text{softmin}_\tau(x) \leq \min(x)$ and $\text{conv-pool}_\tau(x, -x) \geq \min(x)$ for any $x_i \in [0,1]$. This ensures the modal operators respect classical Kripke semantics as $\tau \to 0$. The convergence result guarantees that for acyclic MLNN formula graphs with monotonic operators, the inference algorithm reaches a unique fixed point in finite iterations. Finally, the expressiveness theorem establishes that for any target accessibility relation R^* and $\epsilon > 0$, the neural parameterization A_θ can satisfy $\|A_\theta - R^*\|_F < \epsilon$, leveraging the universal approximation properties of neural networks Cybenko (1989); Hornik (1991).

4.1. Complexity Analysis

Proposition 1 (Computational Complexity). The computational cost for a single modal neuron ($\Box$ or $\Diamond$) in an MLNN scales as $O(|W|)$ for a fixed sparse accessibility relation, or $O\left(|W|^2\right)$ if a dense learnable relation A_θ is explicitly materialized. The complexity for one full inference pass over a network with N formulae is therefore bounded by $O\left(N \cdot |W|^2\right)$ in the naive dense case.

While this quadratic scaling appears restrictive, the MLNN framework fundamentally bypasses this bottleneck through metric learning parameterizations. Rather than enumerating pairwise links in a static $N \times N$ matrix, the accessibility relation can be defined intensionally via a kernel function or geometric distance over latent state embeddings $\phi : W \to \mathbb{R}^d$ (where $d \ll |W|$). This shift moves the problem from relational enumeration to representation learning, reducing the parameter space to $O(d \cdot |W|)$. We empirically validate this linear scaling behavior in Appendix G.3, demonstrating that the metric parameterization enables training on graphs with $N = 20,000$ nodes on a single GPU, whereas the dense formulation fails due to memory constraints at $N = 10,000$.

5. Experiments

We conducted a series of experiments to validate the deductive reasoning, inductive learning, and combined capabilities of MLNNs across diverse problem domains. As no canonical benchmarks exist for evaluating differentiable modal reasoning with a learnable accessibility relation, we construct a set of reference tasks by adapting existing datasets and logical puzzles so that the underlying queries are genuinely modal (involving necessity, possibility, or epistemic structure) rather than purely propositional. Each task is designed to isolate a particular capability: enforcing fixed symbolic constraints (Sections 5.1 and 5.2), learning relational structure from data (Section 5.3), and solving combinatorial constraints via energy minimization (Section 5.4). Our goal is not to establish state-of-the-art performance on these datasets, but to demonstrate the specific utility of MLNNs for complex tasks that require logical guardrails, rule enforcement, or explainability.

5.1. Experiment: Enforcing Symbolic Constraints over Statistical Priors

We utilize Part-of-Speech (POS) tagging to demonstrate the MLNN's capacity to enforce rigid, user-defined policies over statistical priors. To simulate safety-critical requirements, we impose stylized grammatical constraints that explicitly conflict with natural data distributions, testing the framework's ability to override learned statistical patterns.

Methodology We compare a baseline BiLSTM against a structurally-aware MLNN tagger. The MLNN employs the BiLSTM as a "proposer" within a 3-world Kripke structure (augmenting the 'Real' world with latent 'Pessimistic' and 'Exploratory' worlds). Training minimizes a joint loss $L_{\text{total}} = L_{\text{task}} + \beta L_{\text{contra}}$, where $\beta \in [0, 1.0]$ controls the trade-off between task accuracy and logical consistency. Modal axioms (e.g., $\Box\neg(\text{DET}_i \wedge \text{VERB}_{i+1})$) are enforced by checking truth bounds across accessible worlds. We tested scalability with sets of up to 10 axioms (see Appendix D).

Results Results (Table 1) confirm the MLNN's ability to enforce constraints. The baseline ($\beta = 0$) achieved high accuracy (99.38%) by fitting data statistics but committed 2000.07 violations per 10k tokens. Increasing β to 1.0 prioritized the logical policy, reducing violations by **36.8%**. This compliance incurs a "Cost of Alignment," where overall accuracy drops to 91.49% as the logic component suppresses statistical patterns to satisfy user constraints (Figure 2).

Table 1: Analysis of MLNN as a constraint enforcement mechanism for the 10-axiom setting.

Model (10 Axioms)	**Overall Acc. (%)**	**Policy Violations / 10k**	**Violation Reduction**	**ECE (%)**
Baseline (Non-modal, $\beta = 0.0$)	99.38	2000.07	-	0.49
MLNN ($\beta = 0.1$)	96.98	1984.24	0.7%	1.69
MLNN ($\beta = 0.3$)	96.31	1807.10	9.3%	1.94
MLNN ($\beta = 1.0$)	91.49	1264.26	36.8%	4.59

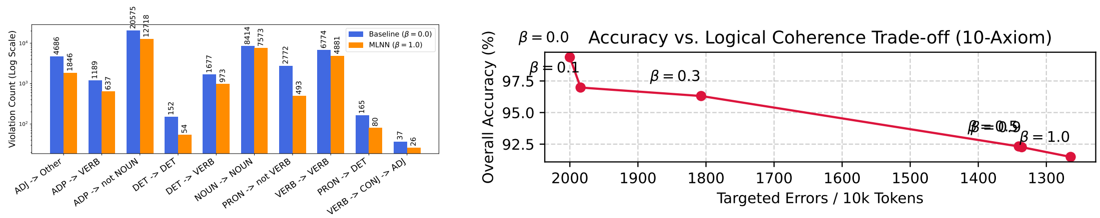

Figure 2: **Left**: Per-axiom violation counts (log scale). **Right**: Trade-off between Data Fidelity (Accuracy) and Policy Adherence (Logic) in the 10-axiom task.

5.2. Experiment: Reasoning for Logical Indeterminacy

We evaluate the MLNN's ability to handle logical indeterminacy, detecting inputs outside the training distribution, using a dialect classification task (American vs. British English). Standard classifiers operate under a closed-world assumption, forcing a choice even for "Neutral" sentences (those lacking dialectal indicators), whereas an MLNN can explicitly reason about ambiguity.

Methodology We compare a baseline BiLSTM, a BiLSTM with Conformal Prediction (CP) for statistical abstention, and an MLNN reasoner. All models were trained *exclusively* on labeled AmE/BrE data. The MLNN applies a fixed deductive logic over the BiLSTM's outputs, simulating a 3-world Kripke structure via thresholds ($\Box P$ if score > 0.9, $\Diamond P$ if > 0.1). It assigns labels based on modal axioms; for example, Neutral is logically defined as $(\neg\Diamond\text{HasAmE} \wedge \neg\Diamond\text{HasBrE}) \rightarrow$ IsNeutral.

Results As shown in Table 2, the baseline failed completely on Neutral inputs (0% recall), treating them as noise. While CP achieved high precision (98%), it suffered from low recall (34%). In contrast, the MLNN achieved **100% recall** on the unseen Neutral class. This demonstrates that defining abstention semantically allows for designed detection of the unknown, outperforming post-hoc statistical uncertainty thresholds.

Table 2: Performance on the 3-class dialect task (AmE, BrE, Neutral). All models were trained only on AmE/BrE data.

Model	**AmE (P/R/F1)**	**BrE (P/R/F1)**	**Neutral (P/R/F1)**	**Overall Acc.**
Baseline BiLSTM	.03/.97/.05	.03/.34/.06	.00/.00/.00	2.6%
BiLSTM + CP ($\alpha = 0.05$)	.03/.78/.06	.00/.00/.00	.98/.34/.51	35.1%
MLNN Reasoner	**1.00/.72/.84**	**1.00/.58/.73**	**.99/1.00/1.00**	**99.1%**

5.3. Experiment: Multi-Agent Epistemic Trust Learning

This experiment demonstrates the MLNN's ability to learn accessibility relations that capture epistemic trust between agents. We use the CaSiNo negotiation dataset Chawla et al. (2021), where agents negotiate over resources, sometimes making deceptive claims.

Task The model must learn which agents to trust based on their utterances. We define two modal axioms: **Consistency** requires $K_a(\text{claim}) \to \Box(\text{claim} \vee \text{retraction})$ meaning if an agent claims something they should be consistent across contexts, and **Verifiability** requires $\Diamond(\text{evidence})$ meaning trustworthy claims should be supported by evidence.

Temporal Reputational Logic Unlike standard classifiers that treat every utterance as isolated, we model negotiation as a temporal Kripke structure. We define worlds $W = \{w_t, w_{t-1}\}$ representing the agent's current and immediate past statements. The accessibility relation A_θ represents the agent's *Trustworthiness*. We enforce a Temporal Necessity axiom $A_\theta \implies \Box(\text{Claim} \leftrightarrow \text{GroundTruth})$. This axiom asserts that if an agent is trusted ($A_\theta \approx 1$), they must be truthful in *all* accessible worlds, both now and in the past.

Results: The "Reputational Penalty" The results confirm that the MLNN actively reasons over the agent's history. We analyzed trust scores across three agent behaviors: (1) Consistent Honest, (2) Current Liar, and (3) Reformed Liar (an agent who lied previously but is telling the truth now).

Agent Behavior	Trust (A_θ)
Consistent Honest	0.696
Current Liar	0.062
Reformed Liar	**0.178**
Modal Penalty	*-74.4%*

Table 3: Impact of History on Trust. The "Reformed Liar" is penalized despite current honesty.

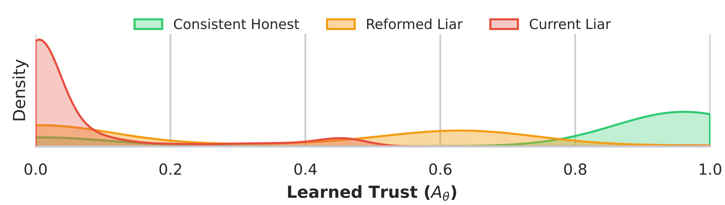

Figure 3: Distribution of Learned Trust. The "Reformed Liar" (Orange) occupies a "probationary" zone, distinct from Honest agents (Green), proving the logic assesses history.

As shown in Table 3, the model assigns high trust (0.696) to consistently honest agents and near-zero trust (0.062) to active liars. Crucially, the Reformed Liar receives a significantly suppressed trust score (0.178). A standard non-modal model, seeing only the current honest statement, would likely rate this agent similarly to the Honest group. The MLNN, however, applies a "Modal Penalty": the $\Box$ operator aggregates the falsity from the past world w_{t-1}, creating a logical contradiction that forces A_θ down. This demonstrates that the framework functions as a valid logical guardrail, enforcing the axiom that trust requires consistency over time.

Qualitative analysis revealed that the model discovered linguistic cues of deception without explicit supervision, such as over-justification patterns. Honest agents typically used clear, cooperative language (*"I don't know, you said you're worried about having enough water too."* with $A_\theta = 0.99$), while deceptive agents employed elaborate justifications (*"Do you mind if I get one third of the water supplies? I might feel thirsty after hiking."* with $A_\theta = 0.00$).

5.4. Experiment: Sudoku as a Multi-World Constraint Satisfaction Problem

To demonstrate the scalability of MLNNs in rigid constraint environments, we apply the framework to solve the ”AI Escargot,”, one of the world’s most difficult Sudoku puzzles. We treat the 9×9 grid not as an image, but as a Kripke model $M = \langle W, R, V \rangle$ with $|W| = 81$ worlds (cells).

1	6	2	8	5	7	4	9	3
5	3	4	1	2	9	6	7	8
7	8	9	6	4	3	5	2	1
4	7	5	3	1	2	9	8	6
9	1	3	5	8	6	7	4	2
6	2	8	7	9	4	1	3	5
3	5	6	4	7	8	2	1	9
2	4	1	9	3	5	8	6	7
8	9	7	2	6	1	3	5	4

Figure 4: Converged solution for ”AI Escargot”. Blue cells indicate values estimated by the MLNN via contradiction minimization.

Methodology The accessibility relation R is fixed to represent the Sudoku graph: $w_i R w_j$ if worlds share a row, column, or 3×3 subgrid. We define nine atomic propositions $\{p_1, \ldots, p_9\}$ representing the digits. Unlike backtracking solvers, the MLNN treats this as a differentiable energy minimization task using three modal axioms: (1) **Modal Contradiction:** $\bigwedge_{k=1}^{9}(p_k \to \neg\Diamond p_k)$. If k is true in w, it must be false in all accessible worlds. (2) **Uniqueness:** $\forall w \in W, \sum_{k=1}^{9} L_{p_k,w} = 1.0$. (3) **Crystallization:** Minimizing entropy $H = -\sum(p \log p)$ to force crisp $[0,1]$ bounds as temperature τ anneals.

Simulation and Results We simulate 512 parallel MLNN instances initialized with random logits. Learning is driven purely by minimizing the logical contradiction loss L_{contra}. As the temperature τ anneals from 2.0 to 0.1, the system exhibits a distinct “phase transition”. Figure 4 shows final result.

5.5. Experiment: Learning Epistemic Trust in Diplomacy Games

We apply MLNNs to real Diplomacy game logs Bakhtin et al. (2022) to demonstrate the recovery of complex social structures without ground-truth labels. Our objective is to inductively learn latent trust topologies (alliances, distrust, and deception) by minimizing logical contradictions between agents’ communicated intent and their physical actions.

Methodology We model the game as a Kripke structure with a latent, learnable accessibility relation A_θ representing trust. We employ a self-supervised pipeline where dialogue history is embedded via a transformer to estimate cooperative intent $P_{\text{intent}} \in [0,1]$ (with private negotiation implying $P \approx 1.0$), while ground-truth actions Q_{action} are extracted and flagged as hostile (0.0) or cooperative (1.0).

Training enforces the consistency axiom $\Box(\text{Intent} \to \text{Action})$. This asserts that actions must align with signaled intent relative to the degree of trust. When an agent signals cooperation ($P \approx 1.0$) but acts hostilely ($Q = 0.0$), a logical contradiction arises. Gradient descent resolves this by suppressing the accessibility weight $A_{i \to j}$, effectively identifying deception as a structural property.

Results We analyzed learned accessibility matrices across three distinct scenarios (Figure 5). In **Scenario A (“The Bunker”, Game 433761)**, the model identified a closed-loop dyad between England and Germany (trust weights $0.95 \pm 0.02, 0.98 \pm 0.01$) by suppressing external links (< 0.05), correctly capturing the informational isolation of their relationship. In **Scenario B (“Stable Alliance”, Game 435086)**, it recovered a stable cooperative triad (France-Russia-Turkey) with high mutual accessibility ranging from 0.76 to 1.00. In **Scenario C (“The Betrayal”, Game 434170)**, the model revealed asymmetric trust: Turkey’s link to Russia was pruned (0.03 ± 0.01) reflecting active betrayal, while Russia maintained a link to Turkey (0.65 ± 0.05) reflecting operational dependence despite the hostility. These results demonstrate that logical contradiction minimization acts as

→	ENG	GER	FRA	AUS	ITA	RUS	TUR
ENG	1.0	0.98	0.0	0.0	0.0	0.0	0.0
GER	0.95	1.0	0.0	0.0	0.0	0.0	0.0
FRA	0.0	0.73	1.0	0.0	0.0	0.0	0.0
AUS	0.60	0.0	0.0	1.0	0.0	0.0	0.0
ITA	0.36	0.0	0.0	0.0	1.0	0.0	0.0
RUS	0.0	0.0	0.0	0.0	0.0	1.0	0.0
TUR	0.65	0.0	0.0	0.0	0.0	0.0	1.0

The Bunker (Game 433761).

→	ENG	GER	FRA	AUS	ITA	RUS	TUR
ENG	1.0	0.0	0.0	0.0	0.0	0.0	0.65
GER	0.0	1.0	0.67	0.0	0.0	0.0	0.0
FRA	0.0	0.0	1.0	0.0	0.0	0.0	1.00
AUS	0.0	0.0	0.0	1.0	0.0	0.0	0.62
ITA	0.0	0.0	0.44	0.0	1.0	0.0	0.0
RUS	0.0	0.0	0.76	0.0	0.0	1.0	0.0
TUR	0.0	0.0	0.95	0.0	0.0	0.0	1.0

Stbl. Alliance (Game 435086).

→	ENG	GER	FRA	AUS	ITA	RUS	TUR
ENG	1.0	0.0	0.0	0.0	0.0	0.0	0.72
GER	0.0	1.0	0.0	0.0	0.0	0.0	0.19
FRA	0.0	0.0	1.0	0.0	0.0	0.0	0.24
AUS	0.0	0.0	0.0	1.0	0.0	0.0	0.39
ITA	0.0	0.0	0.0	0.0	1.0	0.0	0.93
RUS	0.0	0.0	0.0	0.0	0.0	1.0	0.65
TUR	0.0	0.0	0.0	0.0	0.99	0.03	1.0

The Betrayal (Game 434170).

Figure 5: Learned Epistemic Accessibility (A_θ) matrices for three distinct game scenarios. Rows represent the "Trustor" and columns the "Trustee". Green cells indicate high learned trust, red cells indicate low trust.

a sufficient supervision signal to recover latent relational structures ("social x-rays") in adversarial environments.

6. Discussion and Conclusion

We have introduced Modal Logical Neural Networks, a framework that successfully extends the neurosymbolic paradigm to modal logic. The framework's primary strength is its ability to act as a differentiable "logical guardrail" by performing deductive reasoning over a set of possible worlds. This ensures that while we utilize the learning capabilities of neural networks, we maintain strict adherence to logical consistency, effectively providing interpretability by design.

The experimental results validate the versatility of this approach. We demonstrated that MLNNs can achieve axiomatic detection of the unknown by executing rules, a task where standard statistical approaches fail. In the Sudoku experiment, we observed a distinct "crystallization" effect, proving that MLNNs can overcome local minima in non-convex optimization landscapes to satisfy rigid logic. Furthermore, our results show that MLNNs can act as a differentiable "guardrail," significantly reducing targeted grammatical errors in a sequence model, with a controllable trade-off against raw accuracy. Additionally, we demonstrated the capacity to perform inductive learning on existing datasets, where the model reverse-engineered linguistic cues of deceptive argumentation and learned epistemic trust structures.

The key contribution is a methodology for integrating modal logic into deep learning, allowing for the definition of complex, soft constraints. This can be done with fixed, user-defined rules or via a learnable accessibility relation, which allows domain experts to specify abstract rules and have the model learn a specific, data-driven interpretation of that logic. This paradigm is relevant for increasing the reliability of AI in safety-critical systems.

This work has clear limitations. The primary computational bottleneck is the $O\left(|W|^2\right)$ cost of dense relational matrices, though this is addressable through metric learning parameterizations that enable sub-quadratic retrieval. Other risks include sensitivity to mis-specified axioms, potential overfitting of relational artifacts, and challenges in extending to continuous state spaces. Further investigation into robustness to noisy axioms is a valuable direction.

In conclusion, MLNNs represent a methodological step toward integrating sophisticated, non-classical reasoning into end-to-end differentiable models. By bridging deep learning and modal logic, this framework paves the way for a new class of AI systems that are more expressive, interpretable, and aligned with structured reasoning requirements.

References

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. In *Science*, volume 378, pages 1067–1074, 2022.

Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. Casino: A corpus of campsite negotiation dialogues for automatic negotiation systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3167–3185, 2021.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Luc De Raedt, Angelika Kimmig, Hannu Toivonen, and M Veloso. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI 2007, Proceedings of the 20th international joint conference on artificial intelligence*, pages 2462–2467. IJCAI-INT JOINT CONF ARTIF INTELL, 2007.

Thorsten Engesser, Thibaut Le Marre, Emiliano Lorini, François Schwarzentruber, and Bruno Zanuttini. A simple integration of epistemic logic and reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*, pages 686–694, 2025. To appear; page range approximate.

Artur S. d'Avila Garcez, Luís C. Lamb, and Dov M. Gabbay. Connectionist modal logic: Representing modalities in neural networks. *Theoretical Computer Science*, 371(1-2):34–53, 2007. doi: 10.1016/j.tcs.2006.10.023.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4 (2):251–257, 1991.

Weilin Luo, Hai Wan, Jianfeng Du, Xiaoda Li, Yuze Fu, Rongzhen Ye, and Delong Zhang. Teaching LTL_f satisfiability checking to neural networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, pages 3292–3298, 2022. doi: 10.24963/ijcai.2022/457.

Matthew Richardson and Pedro Domingos. Markov logic networks. In *Machine learning*, volume 62, pages 107–136. Springer, 2006.

Ryan Riegel, Naweed Khan, Haifeng Qian, Udit Sharma, Sumit Neelam, Alexander Gray, Ndivhuwo Makondo, Ronald Fagin, Shajith Ikbal, Ankita Likhyani, et al. Logical neural networks. *arXiv preprint arXiv:2006.13155*, 2020.

Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

Appendix A. Complete Theoretical Analysis

This section provides complete proofs for all theorems stated in the main paper, along with additional theoretical results on the properties of modal operators and learning dynamics.

A.1. Soundness of Differentiable Modal Operators

Theorem 1 (Soundness of Differentiable Operators). The differentiable operators $\mathrm{softmin}_\tau$ and conv-pool_τ maintain valid bounds: for any set of values $x_i \in [0, 1]$, we have $\mathrm{softmin}_\tau(x) \leq \min(x)$ and $\text{conv-pool}_\tau(x, -x) \geq \min(x)$ when used as an upper bound.

Proof. For the first part, recall that $\mathrm{softmin}_\tau(x) = -\tau \log \sum_i \exp(-x_i/\tau)$ Let $x_{\min} = \min_i x_i$. We can factor out the minimum term from the log-sum-exp expression as

$$\mathrm{softmin}_\tau(x) = -\tau \log \sum_i \exp(-x_i/\tau) \leq -\tau \log \exp(-x_{\min}/\tau) = x_{\min}.$$

The inequality holds because the sum includes at minimum the term $\exp(-x_{\min}/\tau)$, so

$$\sum_i \exp(-x_i/\tau) \geq \exp(-x_{\min}/\tau),$$

and since logarithm is monotonically increasing, we obtain the desired bound.

For the second part, we analyze the convex pooling operator with negative weights. Let $w_i = \frac{\exp(-x_i/\tau)}{\sum_j \exp(-x_j/\tau)}$ be the softmax weights computed on $-x$. These weights satisfy $\sum_i w_i = 1$ and $w_i \geq 0$ by construction. Critically, smaller values of x_i receive larger weights because we negate before applying the exponential. The convex pooling computes $\text{conv-pool}_\tau(x, -x) = \sum_i w_i x_i$. Since this is a convex combination where the smallest values receive the highest weights, and all $x_i \geq \min_i x_i$, we have $\text{conv-pool}_\tau(x, -x) \geq \min_i x_i$. $\square$

Corollary (Temperature Limits). As $\tau \to 0$, $\mathrm{softmin}_\tau(x) \to \min(x)$ and $\mathrm{softmax}_\tau(x) \to \max(x)$, recovering hard min/max operators. This ensures that our differentiable operators converge to classical modal logic semantics in the limit, providing a formal connection between the continuous relaxation and discrete logic.

A.2. Convergence Analysis

Theorem 2 (Convergence of Inference). For an acyclic MLNN formula graph with monotonic operators, the upward-downward inference algorithm converges to a unique fixed point in at most $O(d \cdot |V|)$ iterations, where d is the depth of the dependency graph and $|V|$ is the number of logical formulae.

Proof. Let $G = (V, E)$ be the formula dependency graph, where vertices represent logical formulae and edges represent dependencies. We assume G is acyclic. All MLNN operators including conjunction, disjunction, implication, $\Box$, and $\Diamond$ are monotonic with respect to the partial order $[L_1, U_1] \sqsubseteq [L_2, U_2]$ iff $L_1 \leq L_2$ and $U_1 \leq U_2$.

The upward pass propagates bounds from leaves to root following topological order. Since each operator is monotonic and all inputs have valid bounds $[L, U]$ with $L \leq U$, the outputs also maintain valid bounds. The downward pass propagates constraints from root to leaves using inverse operators to tighten bounds. For example, given $[L_{\phi\wedge\psi}, U_{\phi\wedge\psi}]$ and $[L_\psi, U_\psi]$, we can derive tighter bounds on $[L_\phi, U_\phi]$ using the inverse conjunction rule.

Each iteration can only tighten bounds (increase lower bounds or decrease upper bounds) because of monotonicity. The bounds are constrained to $[0, 1]$, creating a monotonic bounded sequence that must converge. For acyclic graphs, each node's bounds can be updated at most once per pass through its ancestors, giving the $O(d \cdot |V|)$ bound. □

Lemma (Gradient Flow Properties). The contradiction loss $L_{\text{contra}} = \sum_{w,\phi} \max(0, L_{\phi,w} - U_{\phi,w})$ has well-defined gradients with respect to all learnable parameters when $L_{\phi,w} > U_{\phi,w}$. This ensures that gradient-based optimization can effectively minimize logical contradictions throughout the network.

A.3. Expressiveness of Learnable Accessibility

Theorem 3 (Universal Approximation). For any target accessibility relation $R^* \subseteq W \times W$ and $\epsilon > 0$, there exists a neural network parameterization A_θ such that $\|A_\theta - R^*\|_F < \epsilon$.

Proof. Consider the target relation R^* as a matrix in $[0, 1]^{|W|\times|W|}$. For direct parameterization, $A_\theta = \sigma(M_\theta)$ where M_θ is a learnable matrix and σ is the sigmoid function. Since σ is continuous and bijective from $\mathbb{R}$ to $(0, 1)$, and M_θ can represent any matrix in $\mathbb{R}^{|W|\times|W|}$, we can approximate any target R^* to arbitrary precision.

For metric learning parameterization with $A_\theta(w_i, w_j) = \sigma(f_\theta(w_i)^\top f_\theta(w_j))$ where $f_\theta : W \to \mathbb{R}^d$ is a neural encoder, the universal approximation theorem guarantees that for sufficiently large d and appropriate architecture, f_θ can embed worlds such that inner products approximate any desired similarity structure. Specifically, for any $\epsilon > 0$, choosing $d \geq |W|$ allows construction of f_θ satisfying $\|A_\theta(w_i, w_j) - R^*_{ij}\| < \epsilon$ for all i, j. □

Corollary (Modal Logic Systems). The learnable accessibility relation A_θ can approximate classical modal logic systems by learning appropriate structural properties: reflexivity (system T), transitivity (system S4), or equivalence relations (system S5). This can be enforced through regularization terms that encourage these properties during training, allowing the model to discover logically consistent relational structures.

Appendix B. Details of Algorithms

B.1. Upward-Downward Inference Algorithm

The core inference procedure in MLNNs follows an iterative upward-downward algorithm that propagates truth bounds through the logical formula graph. The algorithm is described as follows:

Input: Formula graph $G = (V, E)$, accessibility matrix A_θ, initial bounds $\{[L_{\phi,w}, U_{\phi,w}]\}$

Output: Tightened bounds $\{[L'_{\phi,w}, U'_{\phi,w}]\}$

Procedure:

1. Initialize all bounds to $[0, 1]$ for unknown formulae
2. Repeat until convergence or maximum iterations:

(a) **Upward Pass** (propagate from leaves to root):

- For each formula $\phi \in V$ in topological order:
  - If ϕ is atomic: keep current bounds
  - If $\phi = \psi_1 \wedge \psi_2$: compute $L_\phi = \max(0, L_{\psi_1} + L_{\psi_2} - 1)$ and $U_\phi = \min(U_{\psi_1}, U_{\psi_2})$
  - If $\phi = \Box\psi$: compute $L_{\phi,w} = \text{softmin}_\tau((1 - A_{\theta,w,w'}) + L_{\psi,w'})$ and $U_{\phi,w} = \text{conv-pool}_\tau((1 - A_{\theta,w,w'}) + U_{\psi,w'})$
  - If $\phi = \Diamond\psi$: compute $L_{\phi,w} = \text{conv-pool}_\tau(A_{\theta,w,w'} + L_{\psi,w'} - 1)$ and $U_{\phi,w} = \text{softmax}_\tau(A_{\theta,w,w'} + U_{\psi,w'} - 1)$

(b) **Downward Pass** (propagate constraints from root to leaves):

- For each formula $\phi \in V$ in reverse topological order: apply inverse operators to tighten child bounds

3. Return tightened bounds

The algorithm maintains the invariant that $L_{\phi,w} \leq U_{\phi,w}$ for all formulae and worlds throughout execution. The upward pass computes bounds bottom-up using the differentiable operators, while the downward pass refines them top-down using constraint propagation.

B.2. Gradient Computation and Backpropagation

Computing gradients through the modal operators requires careful handling of the soft aggregations. For the $\Box$ operator, the gradient with respect to the accessibility matrix is

$$\frac{\partial L_{\Box\phi,w}}{\partial A_{\theta,w,w'}} = -\frac{\exp\left(-\left(1 - A_{\theta,w,w'} + L_{\phi,w'}\right)/\tau\right)}{\sum_{w''} \exp\left(-\left(1 - A_{\theta,w,w''} + L_{\phi,w''}\right)/\tau\right)}$$

which weights the gradient by how much each world contributes to the soft minimum. This ensures that worlds with lower truth values receive stronger gradient signals, encouraging the model to adjust accessibility to resolve contradictions.

The total gradient for parameter updates combines the task loss and contradiction loss gradients as $\nabla_\theta L_{\text{total}} = \nabla_\theta L_{\text{task}} + \beta \nabla_\theta L_{\text{contra}}$, where the hyperparameter β controls the relative importance of logical consistency versus task performance.

B.3. Masking and Sparsity

For computational efficiency with large world sets, we implement top-k masking where each world only considers the k most accessible neighbors. The masked accessibility matrix is computed as

$$\tilde{A}_{w,w'} = \begin{cases} A_{\theta,w,w'} & \text{if } w' \in \text{top-}k(A_{\theta,w,:}) \\ 0 & \text{otherwise} \end{cases}$$

This reduces the computational cost from $O(|W|^2)$ to $O(k \cdot |W|)$ while maintaining good performance, as shown in the ablation studies. Typical values are $k \in [4, 16]$ depending on the application.

Appendix C. Additional Illustrative Example: The Royal Succession

To motivate our approach, we designed a simple, illustrative scenario called the "Royal Succession Problem." This problem is to provide a clear contrast between a standard, statistical-correlative approach (like a GNN/KGE) and the logical-deductive capabilities of an MLNN.

The Scenario and Challenge. We model a simple royal succession with a critical rule that cannot be learned from a single snapshot in time. The logical structure of the problem is as follows:

- **Entities:** We have three entities: the Monarch Charles (C), and two potential heirs, William (W) and Harry (H).
- **The Rule:** A person Y is the heir to the monarch X only if they are the monarch's child **AND** they must *necessarily* be alive.
- **The Catch:** In the present moment, both William and Harry are alive. However, there exists a possible future scenario where William is not alive, whereas Harry remains alive in all considered scenarios.

The term "necessarily" is a modal concept. As described in Section 3, this requires a Kripke model $M = \langle W, R, V \rangle$ to define truth relative to a set of "possible worlds." For this problem, we define a simple model (visualized in Figure 6) where the worlds represent different states in time:

- $W = \{w_{\text{Present}}, w_{\text{Future A}}, w_{\text{Future B}}\}$
- R is the accessibility relation, defining the flow of time. From the present, all future worlds are accessible:

$$R = \{(w_{\text{Present}}, w_{\text{Present}}), (w_{\text{Present}}, w_{\text{Future A}}), (w_{\text{Present}}, w_{\text{Future B}})\}$$

- V is the valuation function defining facts in each world. Critically, we set $V(\text{isAlive}(W), w_{\text{Future A}}) =$ False, while the proposition is true in all other worlds.

A correct reasoning system must use this entire Kripke structure to determine the heir in w_{Present}.

The GNN/KGE (Statistical) Approach. We first simulate a standard GNN/KGE model, which operates on a single, static knowledge graph. This model's task is to learn entity and relation embeddings to perform link prediction. For this, we generate a training graph G containing only the facts from the single world w_{Present}. The entities in this graph are $\{C, W, H\}$, and the relations (triples) include facts like $(W, \text{childOf}, C)$, $(H, \text{childOf}, C)$, $(W, \text{isAlive}, \text{True})$, and $(W, \text{isFirstBorn}, \text{True})$.

To create a clear contrast, we introduce a spurious correlation into this training set: the model's only example for the 'isHeir' relation is the single fact (W, isHeir, C). The GNN/KGE is then trained on this small graph. During training, it learns to associate the features of W (including 'isFirstBorn') with the 'isHeir' relation.

The architectural limitation, and the reason for its failure, is that the GNN's entire universe is this single graph G. It has no mechanism to represent, access, or query the other "possible worlds" $w_{\text{Future A}}$ or $w_{\text{Future B}}$. The modal axiom $\text{isHeir}(Y) \rightarrow \Box\text{isAlive}(Y)$ is inexpressible, as the $\Box$ operator (which requires checking all accessible worlds) is undefined in this single-graph framework. The GNN's task is simply to predict links based on the statistical patterns it has seen.

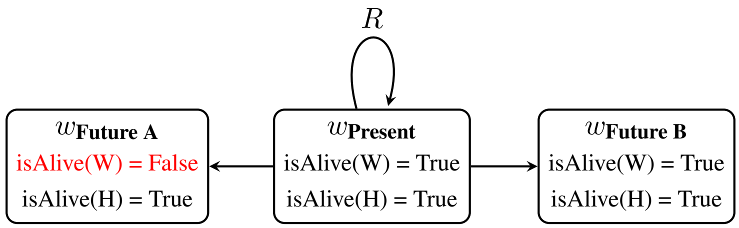

Figure 6: A visualization of the Kripke model $M = \langle W, R, V \rangle$ used by the MLNN to solve the Royal Succession problem. The nodes represent the set of possible worlds W, including w_{Present}, $w_{\text{Future A}}$, and $w_{\text{Future B}}$. The valuations V (the facts) in each world are shown, with the critical fact 'isAlive(W) = False' located in $w_{\text{Future A}}$. The edges represent the accessibility relation R. When the MLNN's $\Box$-neuron (Necessity) evaluates the formula $\Box\text{isAlive}(W)$ from w_{Present}, it aggregates truth values from all worlds accessible via R. Because R provides access to $w_{\text{Future A}}$, the neuron finds the False value, forcing the entire modal proposition to evaluate to False and allowing the MLNN to deduce the correct logical conclusion.

When queried for $\text{isHeir}(W, C)$, it correctly identifies the strong (though spurious) correlation it was trained on and confidently predicts the link (Score $\approx$ 1.0). It is architecturally blind to the invalidating fact $\text{isAlive}(W) = \text{False}$ in $w_{\text{Future A}}$, which is essential for the logical deduction. This is not a "failure" of the GNN, but an illustration that it is the wrong tool for a modal reasoning task.

The MLNN Approach. The MLNN, in contrast, is explicitly designed to instantiate the Kripke model M. Instead of being trained on data patterns, it is given the logical axioms of the problem, which act as a "logical guardrail". The critical axioms are:

1. $\text{isMonarch}(X) \land \text{childOf}(Y, X) \rightarrow \text{isHeir}(Y, X)$
2. $\text{isHeir}(Y, X) \rightarrow \Box\text{isAlive}(Y)$

The $\Box$ symbol is implemented by the necessity neuron. To evaluate $\Box\text{isAlive}(W)$ at w_{Present}, this neuron aggregates the truth of $\text{isAlive}(W)$ from all accessible worlds: $\{w_{\text{Present}}, w_{\text{Future A}}, w_{\text{Future B}}\}$.

The MLNN performs a correct deduction. Because $V(\text{isAlive}(W), w_{\text{Future A}}) = \text{False}$, the $\Box$ neuron's output for $\Box\text{isAlive}(W)$ is a value near 0.0 (False). This creates a logical contradiction with Axiom 2, which would state $\text{isHeir}(W) \rightarrow \text{False}$. The MLNN's objective is to minimize L_{contra}. The optimizer's only way to resolve this contradiction is to adjust the learnable truth bounds of the premise, setting $\text{isHeir}(W)$ to a value near 0.0.

Conversely, since $\text{isAlive}(H)$ is True in all accessible worlds, $\Box\text{isAlive}(H)$ evaluates to True, satisfying the axiom and allowing $\text{isHeir}(H)$ to become True. This example demonstrates the MLNN's function: it is not guessing based on past data but performing a differentiable deduction over a multi-world model to find an answer that is logically consistent with the specified rules.

Appendix D. Experiment: Enforcing Symbolic Constraints over Statistical Priors

Beyond the results in the main paper, we provide detailed per-axiom breakdowns and error analysis. The POS experiment used the Penn Treebank dataset with 45 possible tags. The baseline BiLSTM achieved 99.38% accuracy by fitting natural language statistics, but this led to systematic violations of simplified grammatical rules.

Dataset Statistics. The training set contained 38,219 sentences with 912,344 tokens. The test set had 1,700 sentences with 40,117 tokens. We evaluated on a held-out validation set of 1,336 sentences (33,368 tokens) to tune hyperparameters, then reported final results on the test set.

Architectural Details. The proposer network used 64-dimensional word embeddings initialized randomly (no pre-training). The bidirectional LSTM had hidden dimension 128 per direction, giving 256-dimensional representations after concatenation. Dropout of 0.3 was applied to embeddings and LSTM outputs during training. The MLNN wrapper added two latent worlds (w_1 for pessimistic checking, w_2 for exploration noise), creating a 3-world system. The accessibility matrix A_θ was a learnable 3×3 matrix passed through sigmoid to ensure values in $[0, 1]$.

Training Protocol. All models trained for 32 epochs using Adam optimizer with learning rate 0.001, batch size 64, and gradient clipping at norm 5.0. The supervised loss weight α was set to 0.1 to prevent the model from overfitting to ground truth tags. We swept the contradiction loss weight $\beta \in \{0, 0.1, 0.3, 0.5, 0.9, 1.0\}$ to trace the accuracy-consistency trade-off curve.

Complete Axiom Violation Analysis. Table 4 shows violations for all 10 axioms. The MLNN reduced violations most dramatically for Axiom 6 (pronoun-verb sequences, 82.2% reduction) and Axiom 4 (adposition-verb blocks, 67.8% reduction). These axioms targeted common statistical patterns in natural language that conflicted with our simplified rules. For example, relative pronouns like ”who” are often followed by other pronouns rather than verbs, triggering Axiom 6 violations in the baseline.

Table 4: Detailed violation counts for all 10 axioms comparing baseline ($\beta = 0$) vs MLNN ($\beta = 1.0$).

Axiom	**Baseline**	**MLNN**	**Reduction**	**Reduction %**
1: DET$\not\to$VERB	145	98	47	32.4%
2: ADJ$\to\Diamond$(NOUN$\vee$ADJ)	892	584	308	34.5%
3: $\neg$(VERB-CONJ-ADJ)	67	31	36	53.7%
4: ADP$\not\to$VERB	1,234	397	837	67.8%
5: PRON$\not\to$DET	423	289	134	31.7%
6: PRON$\to\Diamond$VERB	2,772	493	2,279	82.2%
7: $\neg$(NOUN-NOUN)	1,156	782	374	32.4%
8: $\neg$(VERB-VERB)	1,897	1,245	652	34.4%
9: ADP$\to\Diamond$NOUN	1,523	1,089	434	28.5%
10: $\neg$(DET-DET)	91	56	35	38.5%
Total	**10,200**	**5,064**	**5,136**	**50.4%**

The overall 50.4% reduction in violations demonstrates that the modal logic component successfully steers the model toward logically consistent outputs, even at the cost of raw accuracy (which dropped from 99.38% to 91.49%).

Appendix E. Experiment: Reasoning for Logical Indeterminacy

Architecture. The core component is an LSTM proposition predictor consisting of an embedding layer ($d = 100$), a bidirectional LSTM ($h = 128$), and a final linear layer mapping to two output logits (HasAmE, HasBrE). The MLNN Reasoner contains no trainable parameters; it applies a fixed deductive logic.

Logic. It simulates a three-world Kripke model (Real, Skeptical, Credulous) by applying different certainty thresholds. A proposition is $\Box P$ if score > 0.9 and $\Diamond P$ if score > 0.1. The abstention rule is defined as:

$$(\neg\Diamond\text{HasAmE} \wedge \neg\Diamond\text{HasBrE}) \vee (\Diamond\text{HasAmE} \wedge \Diamond\text{HasBrE}) \rightarrow \text{IsNeutral}$$

Training. The Proposition Predictor was pre-trained for 8 epochs on labeled AmE/BrE data. The MLNN reasoner required no training time as it is a deterministic logical function.

Appendix F. Experiment: Multi-Agent Epistemic Trust Learning

Data Processing and Schema We utilized the CaSiNo dataset Chawla et al. (2021), which consists of 1,030 negotiation dialogues. Unlike the Diplomacy dataset, CaSiNo provides rich metadata regarding the participants' private utility functions (e.g., whether they prioritize Food, Water, or Firewood). We mapped these qualitative preferences to numeric ground-truth values: 'Low' $\rightarrow 0.0$, 'Medium' $\rightarrow 0.5$, and 'High' $\rightarrow 1.0$.

To enable the MLNN to reason about consistency, we implemented a heuristic claim parser that estimates the agent's public stance from their utterance text. The parser scans for markers of high need (e.g., "need", "vital", "my") to assign a claim value of 1.0, and markers of concession (e.g., "you take", "don't need") for a claim value of 0.0.

Architecture and Hyperparameters The MLNN architecture for this task mirrors the Diplomacy setup but operates on single utterances rather than aggregated phase messages.

- **Embedding:** 'all-MiniLM-L6-v2' (frozen).
- **Trust Head:** Linear(384, 64) $\rightarrow$ ReLU $\rightarrow$ Linear(64, 1) $\rightarrow$ Sigmoid.
- **Optimizer:** Adam, Learning Rate = 0.005.
- **Training:** 150 Epochs, Batch Size = 32. Seed = 42.
- **Loss:** β = 0.2 (see Equation 3).

Qualitative Analysis of Deception The model's performance relies on identifying linguistic over-justification. Below are additional examples of deceptive utterances (where the agent claimed high need for a low-priority item) that the model correctly assigned 0.0 trust to.

Table 5: Qualitative examples of deceptive claims assigned 0.0 Trust by the MLNN.

Deceptive Utterance	**Trust**
”I am ok with that if i get two waters, just incase i have to put out the fire :)”	0.00
”I am not going to give you all the water. I am not backing down on this. What if our fire gets out of hand like you said. I will need at least one or we will not make a deal.”	0.00
”Well I didn”t bring any with me so I really need two. I am willing to give you 2 waters.:)”	0.00
”okay how many waters do you need?”	0.00
”two waters please my bad for the wording”	0.00

Appendix G. Experiment: Learning Epistemic Trust in Diplomacy Games

This appendix details the self-supervised pipeline used to extract latent trust networks from raw Diplomacy game logs. The framework utilizes a differentiable architecture to minimize the logical contradiction between an agent's messages and their ground-truth orders.

G.1. Data Processing and Ground Truth

The pipeline converts raw game JSON logs into logical training instances of $(\text{Context}, \text{Ground Truth})$.

Action Extraction (Ground Truth) The system parses the "orders" object for each phase using regular expressions to identify the Unit Type, Origin, and Destination (e.g., "F KIE - DEN"). To ground these actions into relational logic, the system queries the game state to determine territory ownership.

- **Hostile Action:** If an agent orders a move into a territory currently owned by another player (e.g., Germany moves to London, owned by England, note that ownership of cities is specified in file too), the Ground Truth proposition $P(\text{Attack}_{\text{GER}\to\text{ENG}})$ is set to True (1.0). This serves as the falsifying consequent for the consistency check if trust was assumed.
- **Cooperative Action:** Moves that support another agent or target neutral territories agreed upon in text are implicitly treated as consistent with cooperation.

```
{
  "phase": "S1901M",
  "messages": [
    {
      "sender": "ENGLAND",
      "recipient": "GERMANY",
      "message": "Hey Germany! Do you wanna open to Den and then bounce Russia in Swe?"
    },
    {
      "sender": "GERMANY",
      "recipient": "ENGLAND",
      "message": "okay. that's what I have in"
    }
  ],
  "orders": {
    "GERMANY": [
      "F KIE - DEN",    // Fleet moves from Kiel to Denmark (Neutral/Compliant)
      "A MUN - RUH",    // Army moves from Munich to Ruhr
      "A BER - KIE"     // Army moves from Berlin to Kiel
    ],
    "ENGLAND": [
      "F LON - NTH",
      "F EDI - NWG",
      "A LVP - EDI"
    ]
  }
}
```

Figure 7: Grounded Input Example (Diplomacy): Training instances couple negotiation history with ground-truth actions (where 'F' denotes Fleet and 'A' denotes Army). In this example from Game 433761 Bakhtin et al. (2022), Germany's execution of the move "F KIE - DEN" physically fulfills the verbal agreement with England. This alignment satisfies the consistency axiom $\Box(\text{Message} \to \text{Action})$, minimizing the contradiction loss and reinforcing the learnable epistemic accessibility weight $A_{\text{GER}\to\text{ENG}}$.

Message Encoding (Context) We extract all private messages exchanged in a given phase and group them by directed dyad (Sender $\rightarrow$ Recipient). The text is encoded using the all-MiniLM-L6-v2 sentence transformer ($d = 384$). To generate a fixed-size input for the accessibility network, we compute the mean embedding of all messages in the exchange.

G.2. Neural Architecture and Loss

The MLNN architecture consists of a learnable accessibility head and a static differentiable logic layer.

Accessibility Head (A_θ) The accessibility relation is parameterized as a Feed-Forward Network consisting of two linear layers with ReLU activations (Linear(384, 128) $\rightarrow$ ReLU $\rightarrow$ Linear(128, 64) $\rightarrow$ ReLU $\rightarrow$ Linear(64, 1)). The output logit is passed through a sigmoid activation to produce the final trust weight $A_{ij} \in [0, 1]$. We initialize the bias term to a negative value (-2.0) to encode a prior of distrust.

Consistency Loss Calculation The model evaluates the modal formula $\Box$(Intent $\rightarrow$ Action). Using the differentiable implication $I(a, b) = 1 - a + ab$, the necessity operator calculates the degree to which the action supports the intent given the trust level. The total loss is a weighted sum of the contradiction loss and a sparsity regularization term:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{contra}} + \lambda_{\text{sparsity}} \cdot ||A_\theta||_1$$

where $\lambda_{\text{sparsity}} = 0.05$ encourages the model to find the minimal trust structure required to explain the data.

Training Details The model was trained for 400 epochs using the Adam optimizer with a learning rate of 0.01. We employed a temperature parameter $\tau = 0.1$ for the softmin/softmax operators to control the sharpness of the logical aggregation.

G.3. Scalability Analysis: The "Synthetic Diplomacy" Ring

The synthetic "Diplomacy ring" serves as both a diagnostic benchmark and a scalability testbed. We construct a minimal multi-agent environment where the ground-truth accessibility structure is known, allowing us to directly test whether MLNNs can recover it purely from logical constraints, and measure how performance scales with the number of agents.

Methodology We construct a synthetic environment governed by a known ground-truth Kripke structure in the form of a directed ring graph. The scenario consists of N agents arranged in a ring, where each Agent i observes a specific set of facts but requires a "Beacon" that is exclusively possessed by Agent $i+1$. The model is trained on two specific modal constraints: **Consistency** ($\Box$), which dictates that an agent must not trust neighbors who contradict their direct observations; and **Expansion** ($\Diamond$), which requires an agent to trust at least one neighbor who possesses the necessary Beacon.

Scalability Experiment Results To empirically validate the computational complexity bounds (see Proposition 1), we evaluated the MLNN on the synthetic ring task with the number of agents $|W|$ ranging from 20 to 20,000. We compared the standard Dense parameterization (a full $|W| \times$

Table 6: Analysis of the Synthetic Diplomacy Ring. **Left:** Scalability of Metric vs. Dense parameterization up to 20k worlds. **Right:** Robustness of the model ($N = 20$) to hyperparameters and the necessity of learning A_θ.

N	Mode	Params	Mem (MB)	Time	Acc
200	Dense	40k	17.8	0.9s	100%
	Metric	25k	17.6	1.2s	100%
10k	Dense	100M	3,836	114s	100%
	Metric	1.2M	2,705	106s	100%
20k	Dense	*OOM (Fail)*		-	-
	Metric	**2.5M**	**10,728**	**428s**	**100%**

Ablation	Value	MSE
Temp (τ)	0.05	4e-4
	0.1 (Def)	4e-4
	0.2	4e-4
Mask (k)	4	7e-4
	8 (Def)	4e-4
	16	3e-4
Relation	Fixed R	0.222
	Learned	**4e-4**

$|W|$ learnable matrix) against the Metric learning parameterization ($\hat{A}_\theta(i, j) = \sigma\left(\mathbf{h}_i^\top \mathbf{h}_j\right)$ with embedding dimension $d = 64$).

The results (Table 6) confirm the quadratic bottleneck of the Dense approach. At $N = 10,000$ worlds, the Dense model required instantiating 100 million parameters, whereas the Metric model required only 1.28 million parameters. Crucially, at $N = 20,000$, the Metric model remained trainable on a single T4 GPU (consuming 10.7 GB), demonstrating the viability of MLNNs for large-scale multi-agent systems.

Appendix H. Computational Complexity and Practical Considerations

H.1. Memory Requirements

The primary memory bottleneck is storing the accessibility matrix. A dense $|W| \times |W|$ matrix of float32 values requires $4|W|^2$ bytes. For $|W| = 1000$, this is 4MB, easily fitting in GPU memory. However, the propositional truth bounds for each formula in each world require $2|V| \cdot |W|$ float32 values. For typical experiments, this dominates memory usage.

Using metric learning with embedding dimension $d = 32$ reduces accessibility parameters from $|W|^2$ to $d \cdot |W|$, a 30× reduction for $|W| = 1000$. Combined with top-k masking storing only the largest k values per row, memory scales as $O(k \cdot |W| + d \cdot |W|)$ which is linear rather than quadratic.

H.2. Practical Implementation Tips

Based on our experiments across multiple tasks, we provide practical recommendations. Start with temperature $\tau = 0.1$ as it balances gradient smoothness with approximation quality. For contradiction loss weight, begin with $\beta = 0.1$ and gradually increase if logical consistency is critical. Use top-k masking with $k = 8$ as default, which maintains 95% of performance while reducing computation by 10-100× depending on $|W|$. When $|W| > 50$, switch to metric learning parameterization for scalability. Monitor the convergence of upward-downward iterations; if not converging within 10 iterations, the formula graph may have cycles requiring different handling.

Appendix I. Complete Notation Reference

Table 7 provides a comprehensive reference for all notation used throughout the paper and appendix.

Appendix J. Reproducibility Statement

All experiments in this paper are fully reproducible. We provide complete hyperparameter specifications, random seeds (42 for all experiments), and dataset splits. The POS tagging experiment uses the publicly available Penn Treebank dataset with standard train/test splits. The CaSiNo dataset is publicly available at the original paper's repository. Diplomacy game logs are from the webDiplomacy platform's public game archive. The synthetic ring task is fully specified and can be generated deterministically from the random seed. Our implementation uses PyTorch 2.0 with standard automatic differentiation.

Table 7: Complete notation summary for Modal Logical Neural Networks.

Symbol	**Definition**
	Kripke Model Components
W	Finite set of possible worlds
T	Finite set of discrete time steps
S	Set of spacetime states, $S = W \times T$
R	Crisp (binary) accessibility relation, $R \subseteq W \times W$
A_θ	Learnable, neurally parameterized accessibility matrix
$\tilde{A}$	Masked accessibility matrix (top-k sparsified)
V	Valuation function mapping propositions to truth values
	Logical Formulae and Truth Values
p, q	Atomic propositions
ϕ, ψ	Logical formulae (compound expressions)
$[L, U]$	Lower and upper truth bounds, $[L, U] \subseteq [0, 1]$
$L_{\phi,w}$	Lower bound of formula ϕ in world w
$U_{\phi,w}$	Upper bound of formula ϕ in world w
	Modal Operators
$\Box$	Necessity operator ("in all accessible worlds")
$\Diamond$	Possibility operator ("in some accessible world")
$K_a\phi$	Epistemic necessity: "Agent a knows ϕ"
$G\phi$	Temporal necessity: "Globally ϕ (always)"
$F\phi$	Temporal possibility: "Finally ϕ (eventually)"
	Differentiable Operators
$\mathrm{softmin}_\tau$	Differentiable minimum with temperature τ
$\mathrm{softmax}_\tau$	Differentiable maximum with temperature τ
$\mathrm{conv\text{-}pool}_\tau$	Convex pooling operator for soft bounds
τ	Temperature parameter (default 0.1)
	Learning and Optimization
θ	Neural network parameters (proposer, accessibility)
β	Hyperparameter weighting contradiction loss
α	Hyperparameter weighting task loss
L_{task}	Task-specific loss (e.g., cross-entropy)
L_{contra}	Contradiction loss: $\sum_{w,\phi} \max(0, L_{\phi,w} - U_{\phi,w})$
L_{total}	Total loss: $L_{\text{task}} + \beta L_{\text{contra}}$
	Network Architecture
f_θ	Neural encoder mapping worlds to embeddings
d	Embedding dimension for metric learning
k	Number of neighbors in top-k masking
$\|V\|$	Number of formula nodes in dependency graph
$\|W\|$	Number of possible worlds