Neural-Guided Kripke Countermodel Synthesis with Symbolic Verification

Anonymous Author(s)

Anonymous Institution(s)

Keywords. Kripke semantics, modal logic, neuro-symbolic AI, automated reasoning

Counterexamples are often easier to check than to discover: once a candidate is given, verification usually reduces to substitution and evaluation. This paper treats that asymmetry as neural-guided model finding. Given a modal formula and requested frame conditions, the neural component proposes a finite Kripke countermodel; correctness is then certified by classical Boolean Kripke evaluation. The method is therefore not a replacement for theorem proving, but a speculative disproving module that can complement ATP/ITP workflows and model finders [2, 4, 5]: success yields a checkable countermodel certificate, while failure to find one makes no validity claim. We use standard propositional modal formulas [1] generated by the following syntax.

$$\varphi ::= \bot \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid \Box\varphi \mid \Diamond\varphi \mid p \quad (p \in \mathsf{Prop})$$

over Kripke structures $\mathcal{M} = (W, R, V)$ with accessibility relation $R \subseteq W \times W$ and valuation $V : \mathsf{Prop} \to 2^W$. Boolean connectives are interpreted classically, and the modal clauses are

$$\mathcal{M}, w \models \Box\varphi \iff \text{for every } v \in W, \text{ if } wRv \text{ then } \mathcal{M}, v \models \varphi,$$

$$\mathcal{M}, w \models \Diamond\varphi \iff \text{there exists } v \in W \text{ such that } wRv \text{ and } \mathcal{M}, v \models \varphi$$

Therefore, a counterexample to a formula α is a Kripke structure $\mathcal{M}$ and a world w that satisfy the conditions corresponding to the specified axiom system and for which $\mathcal{M}, w \not\models \alpha$. The goal of this study is to synthesize such an $\mathcal{M}$ and w directly with a neural model.

We propose a differentiable pipeline for neural-guided synthesis of finite Kripke countermodels, where continuous optimization is combined with symbolic semantic verification. During training and candidate generation, following matrix-based presentations inspired by multi-agent semantics [3], the relation R is represented by a $W \times W$ adjacency matrix and valuations by a $W \times \mathsf{Prop}$ matrix. The smooth formulas below are differentiable relaxations inspired by many-valued semantics [9] and recent neuro-symbolic differentiable-logic work [6, 7, 8]; they serve only as a training-time surrogate and continuous search objective. Proposed candidates are thresholded to crisp structures and ultimately verified under ordinary Boolean Kripke semantics; the finite bound simply specifies the model space being searched.

Let $S^w_\theta \in [0, 1]$ denote the relaxed truth value of a formula θ at each world w. Propositional constants, Boolean connectives, and modal operators are evaluated as follows:

$$S^w_\top = 1 \qquad S^w_\bot = 0 \qquad S^w_{p_i} = V^w_i \qquad S^w_{\neg\alpha} = 1 - S^w_\alpha$$

$$S^w_{\alpha\wedge\beta} = S^w_\alpha S^w_\beta \qquad S^w_{\alpha\vee\beta} = S^w_\alpha + S^w_\beta - S^w_\alpha S^w_\beta \qquad S^w_{\alpha\to\beta} = 1 - S^w_\alpha + S^w_\alpha S^w_\beta$$

$$S^w_{\Box\alpha} = \prod_{v\in W} (1 - A^w_v + A^w_v S^v_\alpha) \qquad S^w_{\Diamond\alpha} = 1 - \prod_{v\in W} (1 - A^w_v S^v_\alpha)$$

Here, $A_v^w \in [0,1]$ represents the strength of accessibility from world w to world v, and $V_i^w \in [0,1]$ represents the truth value of the propositional variable p_i at w. When A and V are crisp, these equations coincide with ordinary Kripke semantics. Thus the continuous operators guide search, while the reported countermodel condition is checked after thresholding by a classical evaluator.

The dataset is correct by construction: the generator samples a finite Kripke structure, applies requested frame closures, detects frame labels, and synthesizes a root-false formula. In the checked-in run, tensors are padded to $W_{\max} = 64$ worlds, the active-world penalty targets two worlds, and frame conditioning uses labels for the T, B, 4, 5, and D closures, with unconstrained K obtained by selecting none. The CNN uses token embeddings of dimension 128, hidden dimension 256, AdamW, and a 0.8/0.1/0.1 train–validation–test split; the exact proposition count, formula-length bound, and modal-depth bound are dataset hyperparameters that should be reported with the generated benchmark.

For symbolic certification, the relaxed matrices are thresholded to a finite relation and valuation, the requested frame closures are reapplied, and the formula is evaluated recursively at the root world using the Boolean clauses above. A successful output is therefore a small certificate—worlds, edges, valuations, and root—that can be checked independently of the neural model.

The main limitation is that this evaluation is a CNN-only feasibility study on a constructive data distribution. The dataset samples a Kripke structure first and then synthesizes a formula falsified at the root, rather than starting from arbitrary formulas and searching for minimal countermodels. Moreover, frame properties are supplied as conditioning labels rather than predicted by the model. Thus, the reported results should be interpreted as evidence for differentiable model search with symbolic certification, not as a complete replacement for classical countermodel search; future work should compare against public modal benchmark efforts such as QMLTP [10].

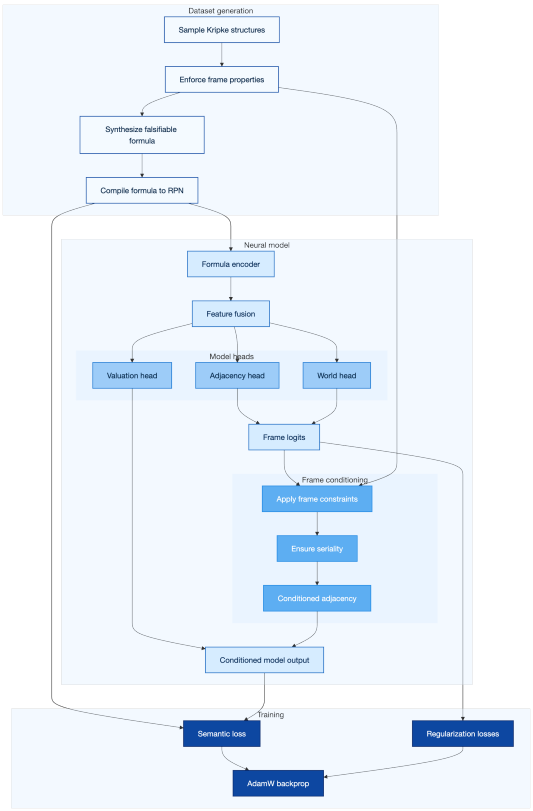

Figure 1: CounterLLM architecture: modal formulas are encoded by a CNN, decoded into candidate Kripke structures, conditioned by frame closures, and checked by relaxed semantics before symbolic verification of crisp countermodels.

Table 1: Final evaluation of the trained model. Semantic accuracy is the validation fraction for which the thresholded, frame-conditioned candidate falsifies the input formula at the root world under Boolean Kripke evaluation. Validation/total losses report the relaxed training objective; redundant raw edges are measured before frame conditioning; active worlds are normalized by $W_{\max} = 64$.

Metric	Final
Semantic accuracy	99.99%
Validation loss	1.07×10^{-4}
Total loss	1.20×10^{-4}
Redundant raw edges	0
Active worlds	0.03125 (2/64)

References

[1] P. Blackburn, J. van Benthem, and F. Wolter, editors. *Handbook of Modal Logic*, volume 3. Elsevier, 2006.

[2] U. Hustadt and R. A. Schmidt. MSPASS: Modal reasoning by translation and first-order resolution. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, LNCS 1847, pages 67–71. Springer, 2000. doi:10.1007/10722086_7.

[3] R. Hatano, K. Sano, and S. Tojo. Linear algebraic semantics for multi-agent communication. In *Proceedings of the International Conference on Agents and Artificial Intelligence – Volume 1*, pages 174–181. SCITEPRESS, 2015. doi:10.5220/0005219001740181.

[4] W. McCune. *Mace4 Reference Manual and Guide*. Technical Memo ANL/MCS-TM-264, Argonne National Laboratory, 2003.

[5] J. C. Blanchette and T. Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *International Conference on Interactive Theorem Proving*, pages 131–146. Springer, 2010. doi:10.1007/978-3-642-14052-5_11.

[6] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, S. Ikbal, H. Karanam, S. Neelam, A. Likhyani, and S. Srivastava. Logical neural networks. *arXiv preprint arXiv:2006.13155*, 2020.

[7] S. Badreddine, A. d'Avila Garcez, L. Serafini, and M. Spranger. Logic tensor networks: Design, implementation and applications. *Artificial Intelligence*, 303:103649, 2022. doi:10.1016/j.artint.2021.103649.

[8] A. Sulc. Modal logical neural networks. *arXiv preprint arXiv:2512.03491*, 2025.

[9] P. Hájek. *Metamathematics of Fuzzy Logic*, volume 4. Springer, 1998.

[10] T. Raths and J. Otten. The QMLTP Library: Benchmarking Theorem Provers for Modal Logics. Technical Report, University of Potsdam, 2011.