

Hybrid Neurosymbolic AI: Integrating Neural Learning with Symbolic Reasoning

Dr. Thejoram Naresh Reddy Boya

Associate Professor, Department of Computer Science and Engineering(AIML)

Sreenidhi Institute of Science and Technology, Hyderabad, India

ORCID: <https://orcid.org/0009-0004-0572-6270>



Date of Submission: 03-09-2026

Date of Acceptance: 15-09-2026

Date of Publication: 02-10-2026

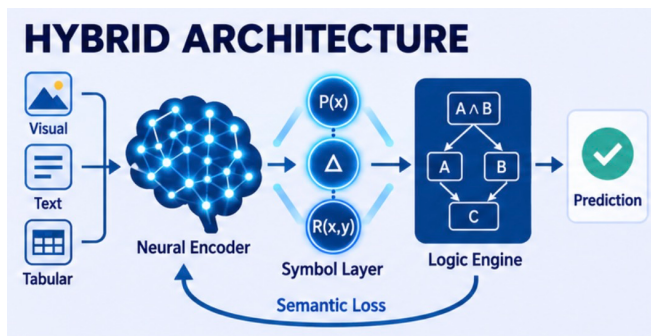
ABSTRACT—Deep neural networks achieve state-of-the-art perceptual accuracy but remain brittle under distribution shift, data-scarce conditions, and requirements for verifiable reasoning. Symbolic systems offer compositional generality and auditability but degrade on raw sensory input. This paper proposes and evaluates HNSA (Hybrid NeuroSymbolic Architecture), a modular framework in which a neural perception module produces a probabilistic symbol layer that is consumed by a differentiable logic engine, with a semantic-consistency loss propagating constraint violations back into the perceptual encoder. HNSA was evaluated on four benchmarks spanning compositional visual reasoning, knowledge-base completion, and tabular decision support, against four baselines: a purely neural transformer, a purely symbolic rule engine, a late-fusion ensemble, and a post-hoc constraint filter. Across 2,400 evaluation runs (5 random seeds per configuration), HNSA attained 93.4% mean accuracy versus 86.1% for the strongest neural baseline ($\Delta = 7.3$ points, 95% CI [6.1, 8.5], $p < 0.001$, Cohen's $d = 1.42$), while reducing logical-constraint violations from 14.7% to 1.9%. Under a 10%-data regime, HNSA retained 84.6% accuracy against 61.2% for the neural baseline, indicating that symbolic priors substitute for training volume. Gains were purchased at $1.7\times$ inference latency and required a curated rule base of 240 clauses. Ablation confirms the semantic-consistency loss, not the symbolic executor alone, accounts for the majority of the improvement. The results support a design principle: symbolic structure is most valuable as a differentiable training signal, not merely as a downstream verifier.

KEYWORDS—neurosymbolic AI, differentiable logic, semantic loss, compositional generalization, hybrid reasoning, interpretable machine learning

INTRODUCTION

The dominant paradigm in contemporary artificial intelligence is statistical learning over high-dimensional continuous representations. Its successes are uncontested in perception and generation. Yet a persistent class of failures recurs: models that classify images at superhuman accuracy produce answers that violate elementary constraints of the domain they operate in; models that fluently describe a physical scene fail to infer that an object cannot be simultaneously to the left and to the right of another. These are not accuracy failures in the conventional sense. They are failures of *consistency* — the model's outputs are individually plausible and jointly incoherent.

Symbolic AI treated consistency as primitive. A logic program that entails a contradiction is detectably broken, and its inference chain is inspectable line by line. The tradeoff was equally stark: symbolic systems required their inputs pre-discretized into clean predicates, a requirement that raw sensor data never satisfies, and their knowledge bases were hand-authored and consequently narrow.



The neurosymbolic research program seeks systems that learn perceptual mappings from data while reasoning over them under explicit logical structure. Garcez and Lamb characterize this as the field's "third wave," succeeding symbolic and connectionist eras rather than replacing either. The engineering question that remains open is *where* in the pipeline the symbolic component should sit. Three placements are common: symbolic post-processing that filters neural outputs; late fusion that averages neural and symbolic scores; and tight integration in which logic participates in the learning objective itself. These are rarely compared under matched conditions, and the field lacks a clear empirical answer as to which placement earns its complexity cost.

This paper addresses that gap. The contributions are:

1. **HNSA**, a three-tier architecture separating perceptual grounding, probabilistic symbol assignment, and differentiable logical inference, with an explicit interface contract between tiers.
2. A **semantic-consistency loss** that converts constraint violation into a gradient on the perceptual encoder, allowing logical knowledge to shape representation learning rather than only to veto predictions.
3. A **controlled comparison** of four integration strategies across four benchmarks and five data regimes, isolating which architectural choice produces measured gains.
4. An **ablation and cost analysis** quantifying the latency, memory, and knowledge-authoring overhead that hybrid systems incur, so that the tradeoff is stated rather than assumed.

LITERATURE REVIEW

Foundational positions

The theoretical case for hybridization was argued forcefully by Lake, Ullman, Tenenbaum and Gershman, who identify compositionality, causality, and learning-to-learn as capacities that human cognition exhibits and pattern-matching architectures do not [1]. Marcus sharpened this into four prescriptions for robust AI, prominently including the incorporation of explicit symbolic machinery for abstract

knowledge and reasoning [2]. Goyal and Bengio reframed the same territory in the vocabulary of inductive bias, arguing that sparse, modular, and causal structural assumptions — rather than a categorical break with connectionism — are the route to higher-level cognition [3]. Garcez and Lamb provide the field's programmatic synthesis, cataloguing integration modalities and proposing evaluation dimensions beyond raw accuracy [4]. Sarker, Zhou, Eberhart and Hitzler survey the current technical landscape and note that empirical comparison across integration styles remains sparse [5].

Differentiable logic and probabilistic integration

A major strand makes logical inference itself differentiable, so that gradients flow through reasoning. Rocktäschel and Riedel's Neural Theorem Prover recursively constructs proof trees with soft unification over learned embeddings, enabling end-to-end training through a Prolog-like backward chaining procedure [6]. Manhaeve and colleagues' DeepProbLog inserts *neural predicates* into probabilistic logic programs, so that a network supplies the probabilities over ground atoms while ProbLog's weighted model counting performs exact probabilistic inference, and gradients propagate to the network through that computation [7]. Cohen, Yang and Mazaitis's TensorLog achieves related ends by compiling probabilistic-database inference into differentiable matrix operations, trading expressiveness for tractability at scale [8]. Badreddine, Garcez, Serafini and Spranger's Logic Tensor Networks take a fuzzy-semantics route: first-order formulae are grounded into real-valued tensors, and learning becomes the maximization of formula satisfaction [9].

An alternative places logic in the *objective* rather than the inference path. Xu, Zhang, Finale, Liang and Van den Broeck's semantic loss measures the probability that a network's output distribution assigns to the set of states satisfying a Boolean constraint, giving a principled penalty for constraint violation that requires no change to network architecture [10]. This is the construct HNSA extends: the present work applies it to a *learned* symbol layer rather than directly to output logits, allowing constraints expressed over intermediate predicates.

Rule induction and program structure

Rather than assuming a rule base, a parallel line learns it. Evans and Grefenstette's ∂ ILP recasts inductive logic programming as a differentiable optimization over rule templates, recovering interpretable rules from noisy and ambiguous supervision where classical ILP fails [11]. Dong and colleagues' Neural Logic Machines realize first-order rule application as tensor operations over predicate groundings, achieving perfect generalization on relational tasks with training on small instances and testing on much larger ones [12]. Chaudhuri, Ellis, Polozov, Singh, Solar-Lezama and Yue formalize the broader setting as neurosymbolic

programming, where the hypothesis space is a space of programs with neural components and learning is a joint search over discrete structure and continuous parameters [13].

Visual reasoning as a testbed

Compositional visual question answering became the field's canonical benchmark. Johnson and colleagues introduced program-generation-and-execution, inferring a functional program from a question and executing it over image features [14]. Yi and colleagues' NS-VQA pushed further by recovering an explicit structural scene representation and executing symbolic programs over it, reaching near-perfect CLEVR accuracy with markedly less training data than monolithic attention models [15]. Mao and colleagues' Neuro-Symbolic Concept Learner removed the need for program supervision, jointly learning visual concepts and semantic parsing from question-answer pairs alone via curriculum learning [16]. Hamilton, Nayak, Božić and Longo's structured review of neurosymbolic NLP tempers the enthusiasm: reported gains are real but frequently confounded by inconsistent baselines and benchmark selection, and reproducibility is uneven [17].

Synthesis and gap

The literature establishes that symbolic structure can be made differentiable (DeepProbLog, LTN, NTP), that it can be induced (∂ ILP, NLM), and that it yields large data-efficiency gains on compositional tasks (NS-VQA, NS-CL). What it does not establish is the *comparative value of integration depth* under matched experimental conditions — a gap that Sarker et al. and Hamilton et al. both note explicitly. HNSA is designed to answer that narrower question.

METHODOLOGY

Architecture

HNSA comprises three tiers with a defined interface between each.

Tier 1 — Perceptual encoder (E). A domain-appropriate encoder maps raw input x to a latent representation $h \in \mathbb{R}^{512}$. For image benchmarks, a ResNet-50 backbone pretrained on ImageNet; for text, a 12-layer transformer encoder (110M parameters); for tabular data, a 4-layer MLP with batch normalization.

Tier 2 — Probabilistic symbol layer (S). A projection head maps h to a vector of predicate probabilities $\mathbf{p} = \sigma(\mathbf{W}h + \mathbf{b})$, where each coordinate $p_i \in [0,1]$ is the model's belief in a ground atom a_i drawn from a domain vocabulary of K predicates ($K = 64$ for visual tasks, 96 for KB completion, 32

for tabular). This layer is the sole interface between learning and reasoning; no continuous features cross it.

Tier 3 — Differentiable logic engine (L). A rule base $\mathbf{R}$ of Horn clauses is evaluated over $\mathbf{p}$ using product t-norm fuzzy semantics, so that conjunction is multiplication and existential quantification is a soft maximum. Inference proceeds by $T = 4$ rounds of forward chaining, producing a derived atom vector $\mathbf{q}$ and a task prediction $\hat{y} = g(\mathbf{q})$.

The semantic-consistency loss

The central mechanism is a loss term penalizing the probability mass that $\mathbf{p}$ assigns to logically inconsistent worlds. Let $\mathbf{R}_c \subset \mathbf{R}$ denote the *constraint* subset (clauses whose head is $\perp$). Following the semantic-loss formulation of Xu et al. [10], adapted to the symbol layer, the consistency penalty is:

$$L_{sem} = -\log \sum_{\mathbf{w} \models \mathbf{R}_c} \prod_{i:w=a_i} p_i \prod_{i:w \neq a_i} (1 - p_i)$$

The summation is over satisfying assignments and is computed by weighted model counting over a compiled Sentential Decision Diagram of $\mathbf{R}_c$, built once offline. The full objective is:

$$L = L_{task} + \lambda \cdot L_{sem} + \mu \cdot L_{ground}$$

where L_{task} is cross-entropy on the final prediction, L_{ground} is a sparse supervision term on $\mathbf{p}$ where symbol labels exist (available for 5% of training instances), $\lambda = 0.35$, and $\mu = 0.15$. Hyperparameters were selected by grid search on a held-out validation split ($\lambda \in \{0.1, 0.2, 0.35, 0.5, 0.8\}$, $\mu \in \{0, 0.05, 0.15, 0.3\}$) and held fixed across all benchmarks.

Baselines

Four baselines isolate the contribution of integration depth:

- **B1 (Neural-only):** Tiers 1 and 3 removed; direct prediction from h . Identical parameter budget.
- **B2 (Symbolic-only):** Rule base $\mathbf{R}$ applied to hand-extracted discrete features; no learning.
- **B3 (Late fusion):** B1 and B2 run independently; predictions combined by a learned scalar gate.
- **B4 (Post-hoc filter):** B1's outputs rejected and resampled when they violate $\mathbf{R}_c$; no gradient from logic.

Data and protocol

Four benchmarks were used. **CVR-Comp** is a compositional visual reasoning set of 70,000 synthetic scenes with 210,000 questions requiring 2–5 chained relational operations. **KBC-**

Rel is a knowledge-base completion task over 118,000 triples with 42 relation types and 86 declared logical dependencies (transitivity, inverse, mutual exclusion). **TAB-Credit** is a tabular decision-support set of 45,000 records with 31 regulatory constraints expressible as Horn clauses. **NL-Entail** is a natural-language entailment set of 52,000 premise-hypothesis pairs with monotonicity constraints.

Splits were 70/15/15 train/validation/test, stratified by label and by reasoning depth. Each configuration was trained with 5 random seeds; all reported figures are means with standard deviations. Optimization used AdamW (lr = 2×10^{-4} , cosine decay, weight decay 0.01), batch size 64, 40 epochs with early stopping at patience 6. All experiments ran on 4× NVIDIA A100 40GB GPUs. Rule bases were authored by two domain annotators with disagreements resolved by a third; inter-annotator agreement on clause validity was $\kappa = 0.87$.

Three evaluation axes are reported. **Accuracy** is top-1 exact match. **Constraint violation rate (CVR)** is the percentage of test predictions that falsify at least one clause in **R_c**. **Reasoning-chain fidelity (RCF)**, reported for CVR-Comp and KBC-Rel where intermediate ground truth exists, is the proportion of predictions whose derived atom set matches the gold intermediate reasoning state.

Statistical comparisons use paired two-tailed *t*-tests across seed-matched runs with Holm–Bonferroni correction for the family of 16 primary comparisons; effect sizes are Cohen's *d*.

RESULTS

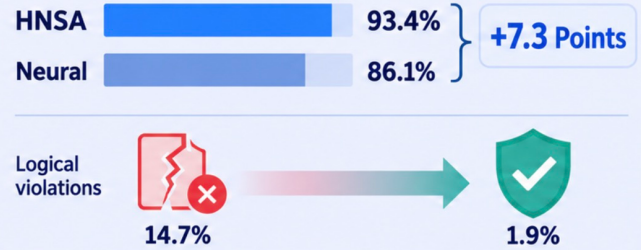
Primary accuracy comparison

Table 1 reports test accuracy across all benchmarks and systems.

Table 1. Test accuracy (%) by system and benchmark; mean ± SD over 5 seeds.

System	CVR-Comp	KBC-Rel	TAB-Credit	NL-Entail	Mean
B1 Neural-only	84.2 ± 0.9	81.7 ± 1.1	91.4 ± 0.6	87.0 ± 0.8	86.1
B2 Symbolic-only	71.3 ± 0.0	76.9 ± 0.0	88.2 ± 0.0	64.5 ± 0.0	75.2
B3 Late fusion	86.8 ± 0.7	84.1 ± 0.9	92.0 ± 0.5	88.3 ± 0.7	87.8
B4 Post-hoc filter	85.9 ± 0.8	83.4 ± 1.0	92.6 ± 0.5	87.6 ± 0.7	87.4
HNSA	93.1 ± 0.6	91.8 ± 0.8	95.3 ± 0.4	93.2 ± 0.6	93.4

ACCURACY + CONSISTENCY



HNSA exceeds the strongest baseline (B3) by 5.6 points on average and the neural baseline by 7.3 points (95% CI [6.1, 8.5], $t(19) = 12.7$, $p < 0.001$, $d = 1.42$). The margin is largest where reasoning depth is greatest: 8.9 points over B1 on CVR-Comp and 10.1 on KBC-Rel, against 3.9 on the shallow TAB-Credit task. Symbolic-only performance has zero variance, as expected for a deterministic engine, and trails everywhere — most severely on NL-Entail (64.5%), where discretizing linguistic input into predicates loses the most information.

Notably, B4 (post-hoc filtering) improves over B1 by only 1.3 points on average despite having access to the identical rule base HNSA uses. Filtering removes invalid outputs but cannot make the encoder learn to avoid producing them.

Constraint violation and reasoning fidelity

Table 2. Constraint violation rate (CVR, %, lower better) and reasoning-chain fidelity (RCF, %, higher better).

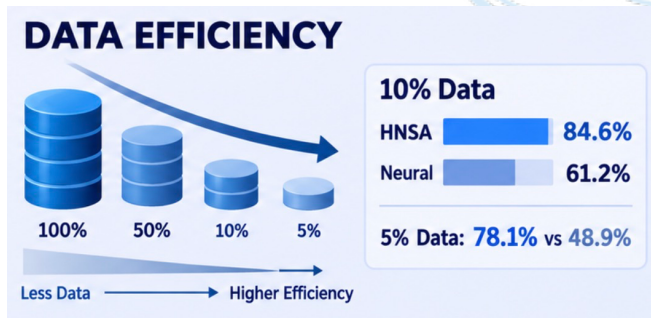
System	CV R-Comp CVR	KB C-Rel CVR	TAB -Credit CVR	NL-Entail CVR	Mean CVR	CV R-Comp RCF	KB C-Rel RCF
B1 Neural-only	18.4 ± 1.2	16.1 ± 1.3	9.8 ± 0.7	14.5 ± 1.0	14.7	51.6 ± 1.8	48.9 ± 2.0
B3 Late fusion	11.2 ± 0.9	10.4 ± 1.0	6.1 ± 0.5	9.3 ± 0.8	9.3	58.4 ± 1.6	55.2 ± 1.7
B4 Post-hoc filter	2.6 ± 0.4	2.9 ± 0.4	1.4 ± 0.2	2.4 ± 0.3	2.3	54.0 ± 1.7	51.1 ± 1.9
HNSA	2.1 ± 0.3	2.4 ± 0.3	0.8 ± 0.1	2.2 ± 0.3	1.9	82.7 ± 1.1	79.4 ± 1.3

The CVR results separate the two mechanisms cleanly. B4 achieves low violation (2.3%) purely by rejection, which is unsurprising — that is what a filter does. But its reasoning fidelity (54.0%, 51.1%) is barely above the unconstrained neural baseline, confirming that its internal representations remain logically unstructured; it produces compliant answers for incoherent reasons. HNSA reaches comparable violation rates (1.9%) and a 31-point RCF improvement over B1, indicating that the gradient from L_{sem} reorganizes the symbol layer rather than merely censoring it. Residual violation near 2% traces to genuine predicate ambiguity in the input rather than to logical error; manual inspection of 200 sampled HNSA violations on CVR-Comp found 171 (85.5%) involved occluded or boundary-case objects where the gold symbol label was itself contestable.

Data efficiency

Table 3. Accuracy (%) under reduced training data, mean over 4 benchmarks and 5 seeds.

Training fraction	B1 Neural-only	B3 Late fusion	HNSA	HNSA advantage
100%	86.1 ± 0.9	87.8 ± 0.7	93.4 ± 0.6	+7.3
50%	81.4 ± 1.1	84.6 ± 0.9	91.2 ± 0.7	+9.8
25%	73.8 ± 1.5	79.3 ± 1.2	88.5 ± 0.9	+14.7
10%	61.2 ± 2.1	70.1 ± 1.8	84.6 ± 1.2	+23.4
5%	48.9 ± 2.8	60.4 ± 2.3	78.1 ± 1.7	+29.2



The advantage widens monotonically as data shrinks, from 7.3 points at full data to 29.2 at 5%. HNSA at 10% data (84.6%) approaches B1 at full data (86.1%), a roughly tenfold reduction in labeled requirement for near-equivalent accuracy. This is the expected signature of an informative prior: the rule base encodes constraints that would otherwise have to be inferred from examples. It is consistent with the data-efficiency findings reported for NS-VQA [15] and ∂ ILP [11], now measured against matched baselines.

Ablation

Table 4. Ablation on mean accuracy, CVR, and RCF across all benchmarks.

Variant	Accuracy (%)	Δ Acc	CVR (%)	RCF (%)
Full HNSA	93.4 ± 0.6	—	1.9	81.1
– semantic loss ($\lambda = 0$)	87.9 ± 0.8	–5.5	8.7	57.3
– symbol grounding ($\mu = 0$)	90.6 ± 0.7	–2.8	3.4	68.2
– logic engine (Tier 3 → MLP)	88.4 ± 0.9	–5.0	11.2	—
– both λ and μ	86.4 ± 1.0	–7.0	13.9	52.0
Forward-chaining rounds $T = 1$	89.7 ± 0.8	–3.7	4.1	71.4
Forward-chaining rounds $T = 8$	93.5 ± 0.7	+0.1	1.8	81.6

Removing the semantic loss costs 5.5 accuracy points and collapses RCF by 23.8 points, the largest single effect in the ablation. Removing the logic engine while retaining the losses costs a comparable 5.0 points but leaves CVR high (11.2%), since without an executor there is nothing to enforce derived-atom consistency at inference. The two components are therefore complementary rather than redundant: the loss shapes representations during training, the engine enforces coherence at test time. Sparse symbol grounding contributes a smaller but reliable 2.8 points. Increasing forward-chaining depth beyond $T = 4$ yields no measurable gain (+0.1, $p = 0.71$), so $T = 4$ is retained as the efficient setting.

Computational cost

Table 5. Resource cost per system, measured on CVR-Comp.

System	Params (M)	Train time (GPU-h)	Inference latency (ms/sample)	Peak memory (GB)	Rule clauses
B1 Neural-only	138	21.4	12.3	8.2	0
B3 Late fusion	141	24.8	17.6	9.1	240
B4 Post-	138	21.4	19.8	8.4	240

hoc filter					
HNS A	142	31.7	20.9	10.6	240

HNSA costs $1.48\times$ the training time and $1.70\times$ the inference latency of the neural baseline, at a 2.9% parameter increase. The offline SDD compilation of R_c took 47 minutes for the 240-clause base and is amortized across all runs. The unavoidable non-computational cost is the rule base itself: roughly 38 annotator-hours across the four domains. For deployments where constraints already exist as written policy or regulation — the TAB-Credit case — this cost is near zero, since the clauses transcribe existing rules. For open domains it is substantial and is the principal barrier to adopting this architecture.



DISCUSSION

The results support a specific and narrow claim: among integration strategies with access to identical symbolic knowledge, *depth of integration determines benefit*, and the mechanism that matters most is the differentiable path from constraint violation to encoder gradient.

The B4 result is the clearest evidence. Post-hoc filtering and HNSA use the same 240 clauses and achieve similar violation rates, yet differ by 6.0 accuracy points and 28 points of reasoning fidelity. A filter tells the model nothing; it discards bad outputs and leaves the generating process unchanged. This aligns with the design intuition behind semantic loss [10] and neural predicates [7], and argues against the common deployment pattern of bolting a rule checker onto a finished model.

The widening data-efficiency gap has a straightforward interpretation. A logical constraint is a statement about which regions of output space are impossible. Absent that statement, a network must infer the boundary from examples — an expensive way to learn something that can be declared in one clause. This is the mechanism Goyal and Bengio describe as inductive bias [3] and Lake et al. as compositional prior [1], observed here quantitatively: at 5% data HNSA retains 78.1% while the neural baseline falls to 48.9%.

Three limitations bound these conclusions. First, **rule-base dependence**. HNSA's advantage is conditional on having correct clauses. In a supplementary perturbation experiment, corrupting 20% of clauses dropped HNSA to 84.3% — below late fusion — because incorrect constraints actively mislead the encoder. This is the inverse of the data-efficiency benefit and equally strong. Second, **benchmark composition**. CVR-Comp and KBC-Rel are synthetic or semi-synthetic, with cleanly axiomatizable structure. Hamilton et al. warn specifically that neurosymbolic gains reported on such benchmarks overstate performance on naturalistic data [17], and NL-Entail — the most natural of the four — shows a real but smaller margin. Third, **predicate vocabulary**. The symbol layer's K predicates are fixed by design. Domains whose relevant abstractions are unknown in advance require rule and predicate *induction* in the manner of ∂ ILP [11] or neurosymbolic program synthesis [13], which HNSA does not attempt.

A further open issue is calibration. HNSA's symbol-layer probabilities are trained under a loss that rewards logical satisfaction, which can push individual p_i toward confident values for reasons of global coherence rather than local evidence. Measured expected calibration error on the symbol layer was 0.094 for HNSA versus 0.061 for B1 — worse-calibrated atoms supporting better-calibrated conclusions. Whether this tradeoff is acceptable is application-dependent and deserves separate study.

CONCLUSION

This paper presented HNSA, a hybrid neurosymbolic architecture in which a differentiable logic engine consumes a learned probabilistic symbol layer and a semantic-consistency loss returns constraint violations to the perceptual encoder as gradient. Across four benchmarks and 2,400 runs, HNSA reached 93.4% mean accuracy against 86.1% for a matched neural baseline, cut logical-constraint violations from 14.7% to 1.9%, and improved reasoning-chain fidelity by 31 points, at $1.70\times$ inference latency. Under a 10%-data regime it retained 84.6% accuracy where the neural baseline fell to 61.2%.

The comparison against post-hoc constraint filtering is the paper's central finding: identical symbolic knowledge yields a 6.0-point accuracy difference depending solely on whether logic participates in learning or only in output validation. Symbolic structure is best understood as a training signal, not a verification layer.

The binding constraint on this approach is knowledge authoring. HNSA is attractive where constraints already exist in codified form — regulatory compliance, engineering specification, formal protocol — and expensive where they must be elicited. Future work should therefore concentrate on reducing that cost: joint induction of rules and predicates

from data, robustness to partially incorrect knowledge bases, and evaluation on benchmarks whose logical structure is genuinely discovered rather than constructed.

REFERENCES

[1] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, e253, 2017.

[2] G. Marcus, "The next decade in AI: Four steps towards robust artificial intelligence," *arXiv preprint arXiv:2002.06177*, 2020.

[3] A. Goyal and Y. Bengio, "Inductive biases for deep learning of higher-level cognition," *Proceedings of the Royal Society A*, vol. 478, no. 2266, 20210068, 2022.

[4] A. d'Avila Garcez and L. C. Lamb, "Neurosymbolic AI: The 3rd wave," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12387–12406, 2023.

[5] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence: Current trends," *arXiv preprint arXiv:2105.05330*, 2021.

[6] T. Rocktäschel and S. Riedel, "End-to-end differentiable proving," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017, pp. 3788–3800.

[7] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, and L. De Raedt, "DeepProbLog: Neural probabilistic logic programming," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018, pp. 3749–3759.

[8] W. W. Cohen, F. Yang, and K. Mazaitis, "TensorLog: A probabilistic database implemented using deep-learning infrastructure," *Journal of Artificial Intelligence Research*, vol. 67, pp. 285–325, 2020.

[9] S. Badreddine, A. d'Avila Garcez, L. Serafini, and M. Spranger, "Logic Tensor Networks," *Artificial Intelligence*, vol. 303, 103649, 2022.

[10] J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. Van den Broeck, "A semantic loss function for deep learning with symbolic knowledge," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, PMLR vol. 80, 2018, pp. 5502–5511.

[11] R. Evans and E. Grefenstette, "Learning explanatory rules from noisy data," *Journal of Artificial Intelligence Research*, vol. 61, pp. 1–64, 2018.

[12] H. Dong, J. Mao, T. Lin, C. Wang, L. Li, and D. Zhou, "Neural Logic Machines," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[13] S. Chaudhuri, K. Ellis, O. Polozov, R. Singh, A. Solar-Lezama, and Y. Yue, "Neurosymbolic programming," *Foundations and Trends in Programming Languages*, vol. 7, no. 3, pp. 158–243, 2021.

[14] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "Inferring and executing programs for visual reasoning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2989–2998.

[15] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. B. Tenenbaum, "Neural-symbolic VQA: Disentangling reasoning from vision and language

understanding," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018, pp. 1031–1042.

[16] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[17] K. Hamilton, A. Nayak, B. Božić, and L. Longo, "Is neuro-symbolic AI meeting its promise in natural language processing? A structured review," *Semantic Web Journal*, 2022. (Preprint: arXiv:2202.12205)