

Neural Structural Reasoner: A Brain-inspired Architecture for Reasoning over Structured Knowledge

Zixing Jia[1,2,3,4*] **Yuhang Pan**[1,2,3*] **Ni Ji**[1,2,3†]

[1]Beijing Institute for Brain Research, Chinese Academy of Medical Sciences & Peking Union Medical College, Beijing, 102206, China
[2]Chinese Institute for Brain Research, Beijing; Beijing, 102206, China
[3]Beijing Key Laboratory of Brain Science and Brain-Machine Interface
[4]Sun Yat-sen University, Guangzhou, China

J3540493668@outlook.com panyuhang@cibr.ac.cn niji@cibr.ac.cn

Abstract

Structural reasoning, the ability to recognize and make inferences over the relational structure between objects and concepts, is a hallmark of human cognition, yet prevailing methods often collapse relational topology into flat embeddings, cannot discover hidden structure and lack interpretability. We introduce Neural Structural Reasoner (NSR), a brain-inspired network that preserves relational structure directly in the connectivity and dynamics of coupled neuronal populations. NSR draws inspiration from three biological mechanisms: multi-layered architecture for encoding hierarchical knowledge, stable representations of entity and concepts, and path integration for input-driven state inference. At query time, NSR parallelizes computation over candidate relational structures and leverages confidence-weighted scores to perform link prediction. Across standard knowledge-graph benchmarks, NSR achieves competitive accuracy without leading on every dataset, and has lower reported training times than several neural baselines. Because reasoning is implemented through sequences of human-readable neuron activations, NSR affords native interpretability by tracking intermediate inference steps. The model further extracts latent relational hierarchies and compositional rules, demonstrating the brain-inspired architecture as an effective, efficient, and highly interpretable substrate for structural reasoning.

1 Introduction

Knowledge in the real world is inherently structured—organized by relations such as order, hierarchy, and connectivity—and humans routinely rely on these relations, rather than the entities alone, to reason about the world [19, 22]. This ability to understand and reason about data relationships, which we call *structural reasoning*, has long been regarded as a desideratum for human-like machine intelligence [28, 56]. Machine learning pursues it from several angles: knowledge-graph embeddings map entities and typed relations into continuous spaces for multi-hop queries [24, 25, 55]; graph neural networks impose relational inductive biases via message passing [4, 44, 53]; and large language models, now widely deployed as general reasoners, tackle structural tasks by verbalizing

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

relations and chaining inferences in natural language [60, 23, 17, 38]. Despite their differences, these methods share two recurring weaknesses: they tend to collapse rich structural information into flat vector representations and produce answers without exposing how they were reached—making their reasoning hard to verify and prone to break on deeper, multi-step queries [3, 62].

Biology offers an alternative starting point. Structural reasoning in the brain arises from the coordinated activity of several systems: the hippocampal–entorhinal circuit builds cognitive maps—relational organizations of entities that support inference beyond direct experience across spatial and non-spatial domains [5, 58], and does so predictively and in a manner causally required for structure-based inference [47, 35]; the prefrontal cortex abstracts relations into hierarchical, schema-like representations that generalize across contexts [43]. Existing biologically inspired models, including the Tolman–Eichenbaum Machine [58], Vector-HaSH [10], and related associative-memory architectures [48, 50, 42, 26], draw on these neural principles to represent and retrieve structured knowledge. However, to our knowledge, there still remains a lack of a modular, brain-inspired architecture that can both encode and perform reasoning over structured knowledge at scale.

We address this gap with Neural Structural Reasoner (NSR), a brain-inspired network architecture for structural reasoning. The main advances are four-fold:

- **Network structure as relational structure.** NSR uses network connectivity to represent relational structure, and exploits efficient network dynamics to implement relational reasoning.
- **Parallel reasoning with path-support scores.** NSR computes in parallel over multiple candidate relational structures, returning ranked candidate answers with confidence scores.
- **Competitive performance with high computational efficiency.** NSR achieves competitive, dataset-dependent accuracy across benchmarks while requiring relatively less training time.
- **Interpretability from dynamics.** NSR yields human-readable reasoning paths by tracking network dynamics as reasoning steps at inference.

2 Background and Related Works

2.1 Structural reasoning across machine learning

Structural reasoning has long been studied under Knowledge Representation and Reasoning (KRR) [7], which formalizes knowledge through structured objects—logic rules, ontologies, and graphs—and defines inference as a process operating on that structure. Among these, knowledge graphs (KGs) offer an instantiation that is both scalable in practice and conducive to systematic empirical study. Formally, a KG is a tuple $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ with entities $\mathcal{E}$, relations $\mathcal{R}$, and triples $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where each $(h, r, t) \in \mathcal{T}$ asserts that entity h stands in relation r to entity t [24, 25]. The canonical task is link prediction: given a query $(h, r, ?)$, score candidate tails and recover the missing entity from the relational structure already in $\mathcal{G}$. This makes structural reasoning concrete—inference over typed relations among discrete entities, isolated from perceptual or linguistic confounds.

A central object of study is the structure of relations themselves, since real KGs are pervaded by canonical patterns like symmetry (e.g., `sibling_of`), inversion ($r_1(x, y) \Rightarrow r_2(y, x)$, e.g., `teacher_of` $\leftrightarrow$ `student_of`), and composition ($r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z)$, e.g., `father_of` $\circ$ `father_of` $\Rightarrow$ `grandfather_of`) [49]. Composition is especially important: nearly every relation in benchmark KGs participates in some compositional pattern [33]. Existing methods address these regularities through distinct representational strategies, each with characteristic limitations. **Geometric embedding and tensor-factorization models** encode relations as translations, rotations, bilinear maps, or low-rank interactions, enabling scalable link prediction but leaving composition implicit in vector geometry [6, 49, 31, 52]. **Deep neural models**, including ConvE [15] and message-passing GNNs [45, 63, 21], add nonlinear computation and neighborhood aggregation, but often lack transparent inference trajectories and struggle with long-range relational chains. **Rule-mining and path-based systems** expose logical chains, but rely on combinatorial search and are brittle under missing or noisy edges [61, 14]. LLM-based methods can verbalize triples or retrieve subgraphs, yet their outputs are prompt-sensitive and may not faithfully track executable graph operations. Thus, current methods generally lack a native mechanism for representing relational structure as explicit, reusable, and traceable dynamics over a structured network.

2.2 Neural substrates of structural reasoning

Structural reasoning in the brain appears to rely on a distributed network of brain regions. Conceptual knowledge is often linked to the anterior temporal lobe (ATL), a proposed transmodal semantic hub that integrates modality-specific features into **stable, context-general concepts** [37, 41]. Relational operations recruit prefrontal mechanisms, especially rostrolateral/frontopolar prefrontal cortex, whose **activity increases with relational integration** demands and which is implicated in analogical and multi-relational reasoning [12, 8, 57, 54]. **Flexible composition over relations** further depends on medial temporal lobe circuitry: the hippocampus binds overlapping experiences into relational structures that support novel inference [16, 39, 18], while entorhinal grid-like codes may provide a metric format for cognitive maps spanning physical and conceptual spaces [13, 36]. Through **path integration**, hippocampal–entorhinal circuits update an internal state estimate by accumulating transitions through space [5]; by extension, analogous dynamics may support abstract reasoning by tracking trajectories through relational or conceptual spaces (though direct evidence remains scarce). Thus, structural reasoning in the brain likely emerges from specialized but interconnected regions that support stable entity and relation representations, relation-specific binding of entities, and compositional generalization of relational structures.

2.3 Biologically inspired models of structural reasoning

Some biologically inspired architectures offer important ingredients for relational cognition, though they have not yet fully addressed structural reasoning in the sense of §2.1. The Tolman–Eichenbaum Machine (TEM) [58, 59] repurposes hippocampal–entorhinal path integration for abstract inference, treating relations as velocity-like signals over a learned structural scaffold; however, its entorhinal module is implemented as a backpropagation-trained RNN, leaving open how such computations arise from grid-cell-like dynamics. Vector–HaSH [10] provides a more biologically grounded HPC–EC loop and supports sequence memory via path integration, but its role in abstract relational reasoning remains unexplored. Continuous-attractor models of grid cells [9] offer a mechanistic account in which velocity-related inputs drive state transition to encode movement in space, yet their application is often limited to low-dimensional navigation rather than high-dimensional relational domains. The Brain Bandit Network (BBN) [26] further demonstrates how biologically grounded recurrent dynamics can implement Bayesian inference and uncertainty-guided decision-making. A deeper limitation of many HPC-EC inspired models is that the network dynamics, built for 2D navigation, are confined to a 2D torus, while **abstract relations require input-driven transitions on high-dimensional manifolds**—and biologically grounded mechanisms for such high-dimensional, relation-specific transitions remain underexplored. Together, these works motivate a bio-inspired architecture that can encode and perform reasoning in structured, high-dimensional relation space.

3 The Neural Structural Reasoner Model

3.1 Model Architecture

Neural Structural Reasoner (NSR) maps relational structure between entities onto a hierarchical multi-layered network, where connectivity directly encodes associations between entities, relations and relation compositions (Fig. 1a). The core functional layers are:

Entity Layer (L_E): contains neurons $\{e_i\}^N$ encoding the N entities in the data.

Relations Layer (L_R): contains a total of $2M$ neurons where the first half, $\{r_i\}^M$, encode the M relations and the second half, $\{r_{i+M}\}^M$, encode the inverse relations $\{r^{-1}\}^M$ (e.g., $\texttt{father_of}^{-1}$ is learned to associate with `son_of` via the Hebbian rule in §3.3).

Relational Reasoning Layer (L_Z): contains up to $N \times (2M)$ neurons; a given neuron z_i^k encodes the association between the head entity h_i and the relation r_k.

Relation Composition Layer (L_C): contains a flexible number of neurons encoding the compositional structures (e.g. chains) among relations.

We assess the functional contribution of the mechanisms implemented by these layers in Appendix C.5, through ablations of inverse-relation encoding, relation-equivalence retrieval, compositional inference, and Hebbian updates.

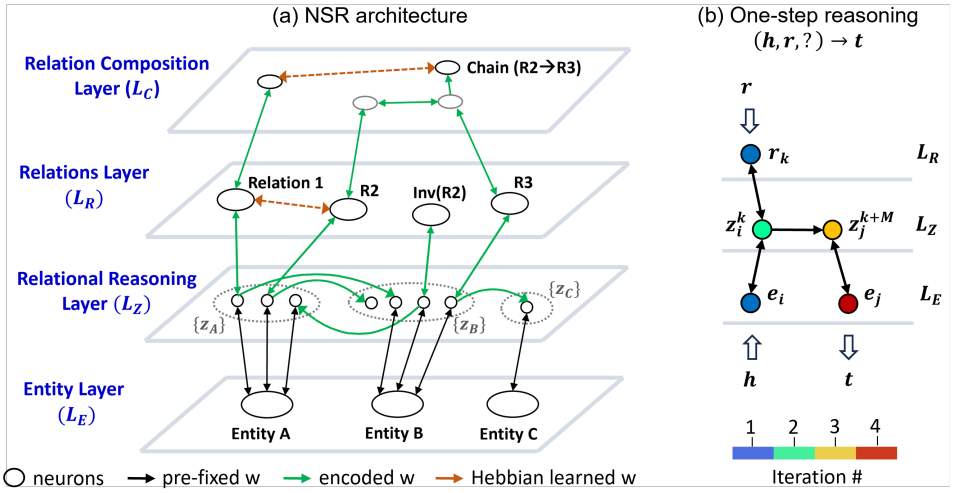


Figure 1: **The Neural Structural Reasoner Architecture.** (a) Four core layers of NSR: L_E encode entities; L_Z instantiates relation-driven state transition; L_R and L_C encode relations and their compositions (e.g., chains). Connections in black are fixed a priori, those in green encode observed triples, those in orange are learned via Hebbian updates to capture latent relational structure. (b) Sequence of neuron activation that implements one-step reasoning over $(h, r, ?) \mapsto t$.

3.1.1 Network Connectivity and Dynamics

The four layers are connected in a bi-directional hierarchy: $L_E \leftrightarrow L_Z \leftrightarrow L_R \leftrightarrow L_C$. L_E and L_Z neurons encoding the same entity are bidirectionally coupled at initialization and held fixed:

$$\forall i \in [N],\ \forall k \in [2M]: \quad W_{E \leftrightarrow Z}(e_i, z_i^k) = 1 \tag{1}$$

All other weights are initialized at 0 and are learnable. The activation state of each layer is updated in a single step by summing up intra- and inter-layer inputs and external query input I, and transformed through sigmoidal nonlinearity σ.

$$\begin{cases} L_E : X_E(t+1) = \sigma_E(W_{E \leftrightarrow Z} X_Z(t)) + I_E, \\ L_Z : X_Z(t+1) = \sigma_Z\big(W_{ZZ} H\big(X_Z(t) - \eta_Z\big) + W_{Z \leftrightarrow E} X_E(t) + W_{Z \leftrightarrow R} H\big(X_R(t) - \eta_R\big)\big) + I_Z, \\ L_R : X_R(t+1) = \sigma_R\big(W_{RR} X_R(t) + W_{R \leftrightarrow Z} H\big(X_Z(t) - \eta_Z\big) + W_{R \leftrightarrow C} X_C(t)\big) + I_R, \\ L_C : X_C(t+1) = \sigma_C\big(W_{CC} X_C(t) + W_{C \leftrightarrow R} H\big(X_R(t) - \eta_R\big)\big) + I_C \end{cases} \tag{2}$$

Neuronal interactions within L_Z are rectified by the Heaviside function H, ensuring that activity propagates only in response to convergent inputs from the associated entity and relation. Similar gating exists for projections $L_Z \rightarrow L_R$ and $L_R \rightarrow L_C$ to conditional activation of co-occurring relations and relation compositions in the learning phase (§3.3). When performing reasoning, the relevant entities and relations obtained in intermediate steps can be read out directly from neuronal activations in L_E, L_R or L_C.

3.2 Encoding of Relation Triples

For each triple (h_i, r_k, t_j) and its inverse (t_j, r_k^{-1}, h_i), we let neurons r_k and r_{k+M} to encode the relation and its inverse, and connect them with the entity-encoding neurons by assigning the following weights :

$$\begin{cases} W_{ZZ}(z_i^k, z_j^{k+M}) = W_{ZZ}(z_j^{k+M}, z_i^k) = 1, \\ W_{Z\leftrightarrow R}(z_i^k, r_k) = 1, \quad W_{Z\leftrightarrow R}(z_j^{k+M}, r_{k+M}) = 1 \end{cases} \tag{3}$$

3.2.1 Reasoning over $(h_i, r_k, ?)$ through Path-Integration-Inspired Dynamics

When queried with the relation triple $(h_i, r_k, ?)$ (Fig. 1b), NSR first activates the entity neuron e_i and the relation neuron r_k. Iterating over (2) then activates neurons z_i^k and z_j^{k+M} through intra-layer dynamics in L_Z. This in turn activates neuron e_j, which allows the readout of the tail entity t_j. This computational logic is inspired by path integration models proposed for entorhinal grid cells [9, 10].

3.3 Learning Compositional Rules among Relations

Following the encoding phase, NSR uses Hebbian-like associative learning to extract the following compositional structure and equivalence rules among relations:

Relational Equivalence: When queried with two relation triples with the same head entity, e.g. (h_i, r_k, t_j) and (h_i, r_l, t_w), L_Z neurons encoding the tail entities, z_j^{k+M} and z_w^{l+M} will be activated. If in fact $t_j = t_w$, then the activity of z_j^{k+M} and z_j^{l+M} can be elevated above the threshold η_z through mutual coupling with e_j. Iterating over (2) leads to co-activation of r_{k+M} and r_{l+M} and of r_k and r_l. The connections between co-activated relation neurons are updated via a symmetrized version of Oja's rule:

$$\Delta w_{ij}^{RR} = \delta_{RR}(r_i * r_j) - \mu_{RR}(r_i^2 + r_j^2)w_{ij}^{RR} \tag{4}$$

Where δ_{RR} and μ_{RR} are learning rates. In practice, all relation triples with the same head entity can be queried at once, and weight update can occur simultaneously for all co-activating r_ks. This ultimately leads to strong coupling between neurons encoding semantically analogous relations.

Symmetric and Inverse Relations: Same network dynamics and learning rule can be applied to extract symmetric and inverse relations. This is achieved by letting $r_l = r_l^{-1}$ in the case of symmetry, and $r_l = r_m^{-1}$ in the case of inverse relations.

Composite relations: We focus on detecting the equivalence between multi-hop relation chains and single triples. Take two-hop chains for example, this amounts to detecting

$$r_p(h_i, x_p) \wedge r_q(x_p, x_q) \Rightarrow r_k(h_i, x_q) \Leftrightarrow r_k^{-1}(x_q, h_i). \tag{5}$$

where the bidirectional equivalence follows from Eq. 3: every triple (h, r, t) is encoded together with its inverse (t, r^{-1}, h). The network discovers such rules by checking whether a two-hop chain returns to the original head entity. Concretely, suppose (h_i, r_p, x_p) is already encoded. NSR computes three inference steps:

Step 1: Activate e_{x_p} (the tail of the known triple) and all relation neurons $\{r_m\}_{m=1}^{2M} \setminus \{r_{p+M}\}$ (all relations except the inverse r_p^{-1}). Iterating dynamics in L_Z activates neurons z_q^m representing candidate tails x_q reachable from x_p via each relation r_m.

Step 2: For each discovered entity x_q, treat it as a new head entity and repeat Step 1, querying $(x_q, r_m, ?)$. This yields a second-hop set of tails $\{x_s\}$ encoded by neurons z_s^m.

Step 3: If any x_s coincides with the original head entity h_i, the network has closed a three-step loop:

$$(h_i, r_p, x_p) \wedge (x_p, r_q, x_q) \wedge (x_q, r_k^{-1}, h_i). \tag{6}$$

Because the loop returns to h_i through r_k^{-1}, the two-hop composition $r_p \circ r_q$ connects h_i to x_q in the direction of r_k, supporting the rule $r_p \circ r_q \Rightarrow r_k$ (Eq. 5). Owing to the bidirectional encoding in Eq. 3, the presence of (x_q, r_k^{-1}, h_i) is equivalent to the presence of (h_i, r_k, x_q); hence the discovered chain is registered in L_C as a compositional association between $r_p \rightarrow r_q$ and r_k.

Whenever such a closed loop is detected, assign a L_C neuron c_k to represent the relation r_k, and neuron $c_{p\rightarrow q}$ to detect the sequential activation of $r_p \rightarrow r_q$. Update weights according to:

$$\Delta w_{k-pq}^{CC} = \delta_{CC}(c_k * c_{pq}) - \mu_{CC}(c_k^2 + c_{pq}^2) w_{k-pq}^{CC} \tag{7}$$

While these procedures are designed for online sequential learning, for static datasets, the existence of composite structures can be detected statistically and then encoded directly in network weights. We present details of these simplifications along with pseudocodes in Appendix A.2.

3.4 Reasoning Phase

Given a query $(h_i, r_k, ?)$ where $(h_i, r_k, t^*) \notin \mathcal{D}_{\text{train}}$, NSR uncovers t^* by first retrieving equivalent relations r_e or relation compositions C_e to r_k, and then performing relational reasoning over these equivalence sets to obtain candidate tail entities $\tilde{\mathcal{T}} = \{\tilde{t}\}$ with path-support scores used for ranking:

Step 1 **Retrieval of equivalent relations:** This is achieved by first activating neuron r_k encoding the query relation, and iterating the dynamics of L_R in (2) once to activate neurons $\{r_e\}$ encoding the set of equivalent relations. A threshold `T_thresh` gates the propagation, retaining only neurons with activation at least `T_thresh` ; their activation levels, which represent how strongly two relations are associated, serve as path-support scores (i.e., $S_R(r_e) = x_e$).

Step 2 **Relational reasoning with $\{r_e\}$:** This is implemented by activating the neuron e_i encoding the head entity, while holding $\{r_e\}$ active. Iterating over the dynamics of L_Z according to (2) implements $(h_i, r_e, ?)$ and subsequently activates neurons $\{\tilde{e}\}$ in L_E. These neurons encode a set of candidate tail entities $\tilde{\mathcal{T}}^{(1)} = \{\tilde{t}\}$.

Step 3 **Retrieval of equivalent relation compositions:** Upon re-initializing the network, activate the query relation neuron r_k and allow its activity to propagate to L_C. Iterating L_C dynamics based on (2) activates neurons encoding relation compositions that are closely associated with r_k. Applying a gate function yields the top-K activated neurons $\{c_e\} = \mathrm{TopK}(X_C)$ corresponding to the top equivalent compositions. The activation of $\{c_e\}$ serves as the path-support scores S_C.

Step 4 **Relational reasoning with $\{c_e\}$:** Sample and activate a single neuron in $\{c_e\}$, which then initiates the sequential activation of L_R neurons $r_{p_1}, r_{p_2}, \ldots, r_{p_\ell}$ in the relation chain. Activate the entity neuron e_i together with r_{p_1} to initiate multi-hop reasoning in L_Z. At the end of the chain traversal, retain the contributing grounded paths together with their terminal entities in L_E. Each path receives the support score $S_C(c_e)$ of its corresponding composition neuron. Repeating this procedure for all selected composition neurons yields the compositional path collection and its candidate tail set $\tilde{\mathcal{T}}^{(2)}$. Distinct groundings of the same relation chain are retained separately for aggregation in Step 5.

Step 5 **Aggregating path-support scores for candidate entities:** For each candidate tail entity t_j, let $\mathcal{P}(t_j)$ be the set of grounded equivalent-relation and compositional paths reaching it, and let $s(p)$ be the support score assigned to path p. We compute

$$\mathrm{Score}_a(t_j) = \begin{cases} \max\limits_{p \in \mathcal{P}(t_j)} s(p), & a = \max, \\ \sum\limits_{p \in \mathcal{P}(t_j)} s(p), & a = \mathrm{sum}, \end{cases} \tag{8}$$

where the aggregation mode a is selected on the validation set. Max aggregation may overvalue a single spurious path, whereas sum aggregation may overcount correlated paths. These scores are used to rank candidates, not as calibrated probabilities; the contributing paths remain inspectable.

4 Experiments

4.1 Experimental setup

Datasets. Our main-table evaluation uses four benchmarks: Nations [27], Kinship [28], YAGO3-10 [15], and FB15k-237 [51], accessed through PyKEEN [2]. These benchmarks span compact relational graphs and larger knowledge graphs. We additionally report experiments on two standard benchmarks, Countries S3 [32] and WN18RR [15], and on our constructed Kinship1990_EXTENDED dataset in Appendix C.4. The construction of Kinship1990_EXTENDED is described in Appendix C.2. For the six public benchmarks, all locally evaluated methods use the provided train/validation/test splits.

Table 1: Link-prediction performance on Nations and Kinship. Hits@1 and Hits@3 are percentages. Higher ranking scores and lower training times are better; the best value in each column is bold.

		Nations			Kinship			Training time	
Category	Method	MRR	H@1	H@3	MRR	H@1	H@3	Nations	Kinship
Embedding-based	ConvE	0.8029	69.05	89.45	**0.7927**	**68.11**	**88.45**	26 s	64 s
	RotatE	0.5351	33.23	66.02	0.7598	63.37	86.26	24 s	84 s
Symbolic rule learning	AnyBURL	0.7994	69.15	89.55	0.6768	54.10	76.63	64 s	63 s
	AMIE	**0.8559**	**77.11**	**92.54**	0.6767	55.03	76.26	0.6 h	**2 s**
Neural rule learning	NeuralLP	0.6841	52.74	81.59	0.6072	47.30	68.06	40 s	26 s
	NCRL	0.4571	25.37	55.22	0.6050	45.81	68.99	134 s	99 s
	RNNLogic	0.7216	61.65	83.06	0.6690	54.94	77.60	32 s	82 s
Path-based reasoning	PRA / PathRank	0.5933	38.31	77.61	0.6296	46.83	73.56	34 s	86 s
	MINERVA	0.5865	44.28	73.13	0.6253	46.37	73.46	0.9 h	1.2 h
GNN-based	NBFNet	0.7479	62.19	84.08	0.7445	62.38	83.61	29 s	77 s
Brain-inspired	NSR	0.8142	71.64	88.16	0.6515	54.21	70.67	**3 s**	11 s

Evaluation. We adopt the filtered link-prediction setting [6], reporting Mean Reciprocal Rank (MRR) and `Hits@K` ($K \in \{1, 3\}$). In addition to predictive accuracy, we measure training time to assess computational efficiency.

4.2 Empirical performance

Tables 1 and 2 compare NSR with ten baselines spanning major approaches to knowledge-graph reasoning: embedding models (ConvE [15] and RotatE [49]), symbolic rule mining (AnyBURL [30] and AMIE [20]), path ranking (PRA/PathRank [29]), neural rule learning (NeuralLP [61], NCRL [11], and RNNLogic [40]), reinforcement-learning path search (MINERVA [14]), and graph neural reasoning (NBFNet [63]). We report filtered MRR and Hits@1/3 on Nations, Kinship, YAGO3-10, and FB15k-237. Locally evaluated methods use the same dataset splits and filtered-tail evaluation protocol.

Across the four benchmarks, NSR delivers competitive predictive performance while maintaining strong training efficiency. It ranks near the top on Nations and YAGO3-10 and remains competitive on FB15k-237, although its performance is weaker on Kinship, indicating dataset-dependent strengths rather than uniformly superior accuracy. At the same time, NSR trains substantially faster than several neural baselines and remains efficient even on the larger graphs. Overall, these results suggest that NSR offers a favorable trade-off between reasoning performance and computational cost, making it a competitive structural reasoner for knowledge-graph tasks where both accuracy and training efficiency matter.

4.3 Traceable reasoning steps

Unlike embedding baselines that rely on post-hoc attention or gradient saliency to explain opaque scalar scores, NSR's interpretability is intrinsic: the activation trajectory *is* the computation. For the query `(Brazil, embassy, ?)` (Figure 2), Step 1 of §3.4 clamps `embassy` onto the L_R layer and iterates the L_R dynamics once through W_{RR}. The Hebbian couplings learned during training (Eq. (4)) activate $\texttt{weightedunvote}^{-1}$ as the dominant equivalent relation, and its activation level serves as the confidence score S_R. Step 2 then activates the head entity `Brazil` in L_E while holding $\texttt{weightedunvote}^{-1}$ active in L_R; the joint input projects onto the L_Z relation map via the $W_{E \leftrightarrow Z}$ and $W_{Z \leftrightarrow R}$, forming the initial state $\boldsymbol{X}_Z^{\text{init}}$. Recurrent dynamics through W_{ZZ} concentrate activation onto the neuron $z_{\text{UK}}^{\text{weightedunvote}^{-1}}$, which is read out in L_E as the tail entity `UK`. Because each active unit represents a discrete, human-readable proposition, errors are auditable by inspecting which relation was activated in L_R or where the L_Z dynamics diverged. The resulting activation sequence provides

Table 2: Link-prediction performance on YAGO3-10 and FB15k-237. Hits@1 and Hits@3 are percentages. Higher ranking scores and lower training times are better; the best available value in each column is bold. A dash indicates a missing entry.

		YAGO3-10			FB15k-237			Training time	
Category	Method	MRR	H@1	H@3	MRR	H@1	H@3	YAGO3-10	FB15k-237
Embedding-based	ConvE	**0.6365**	**59.03**	**71.28**	0.4095	31.61	44.89	10.3 h	991 s
	RotatE	0.1812	9.98	21.04	0.3368	27.07	41.51	1.5 h	1.5 h
Symbolic rule learning	AnyBURL	0.5589	50.78	60.16	0.332*	24.7*	—	1000 s	1000 s*
	AMIE	0.5473	49.58	59.27	0.2170	16.57	23.07	**83 s**	**8 s**
Neural rule learning	NeuralLP	—	—	—	0.3166	24.45	34.31	—	11.9 h
	NCRL	0.380*	27.40*	—	0.300*	20.90*	—	—	—
	RNNLogic	0.5625	49.56	61.80	0.3276	23.49	38.60	4.5 h	0.6 h
Path-based reasoning	PRA / PathRank	0.4678	37.39	55.22	0.0972	6.35	10.72	239 s	229 s
	MINERVA	—	—	—	0.2734	19.77	30.34	—	3.3 h
GNN-based	NBFNet	0.4946	37.96	55.54	**0.5114**	**41.64**	**55.95**	2.3 h	10.4 h
Brain-inspired	NSR	0.5893	52.80	64.32	0.3649	28.45	39.74	0.3 h	0.6 h

*Values reported in prior work or official releases. The AnyBURL FB15k-237 results and its 1000 s rule-learning budget come from the official results page (`https://web.informatik.uni-mannheim.de/AnyBURL/`); NCRL values come from [11]. Published and cross-hardware timings are indicative rather than strictly matched; The reported NSR training times correspond to the accelerated implementation for static knowledge graphs described in Appendix A.2.1.

an inspectable computational trace, inspired by evidence of non-spatial task-state replay in the human hippocampus [46]. Symmetric trace is given in Appendix D.1.

4.4 Learning of latent structures

Latent relational hierarchy. NSR recovers the latent compositional structure of the relational space rather than memorizing pairwise co-occurrences. As shown in Figure 3, the model learns which relation sequences form reliable multi-hop pathways. For example, the chain `exportbooks` + `releconomicaid` $\rightarrow$ `embassy` attains confidence 0.93 because the intermediate transition constitutes a coherent diplomatic pathway, whereas merely repeating `embassy` twice fails to yield a high-confidence rule despite the relation's high frequency. This selectivity shows that composition is context-sensitive and depends on whether a relation pair creates a semantically stable bridge. Notably, on Kinship1990_EXTENDED the model can assign confidence 1 to deterministic kinship rules after a single observation by integrating prior logical constraints through hyperparameters, demonstrating that NSR is not limited to statistical induction.

Relational organization of entities. NSR encodes entities not as isolated symbols but as nodes characterized by their recurrent connections L_Z. As illustrated in Figure 4, projecting entities based on their relational connections (W_{ZZ}) induces structurally meaningful clusters: the Western capitalist core (UK, USA), the Eastern socialist core (USSR, China), and distinct non-aligned blocs (Jordan, Egypt, India, Brazil) each occupy separate regions of the latent space. Taken together, NSR learns a dual latent structure: a compositional hierarchy governing how relations combine, and a relational topology governing how entities are organized.

5 Conclusion

We presented Neural Structural Reasoner (NSR), a brain-inspired architecture that integrates hierarchical representation with relation-driven state transition and Hebbian-style learning to accomplish structural reasoning over knowledge graphs. By preserving relational structure in layered network connectivity rather than collapsing it into flat embeddings, NSR achieves competitive, dataset-dependent accuracy on standard benchmarks, while its inference dynamics natively expose human-readable reasoning traces. NSR also spontaneously discovers interpretable latent structures such as composi-

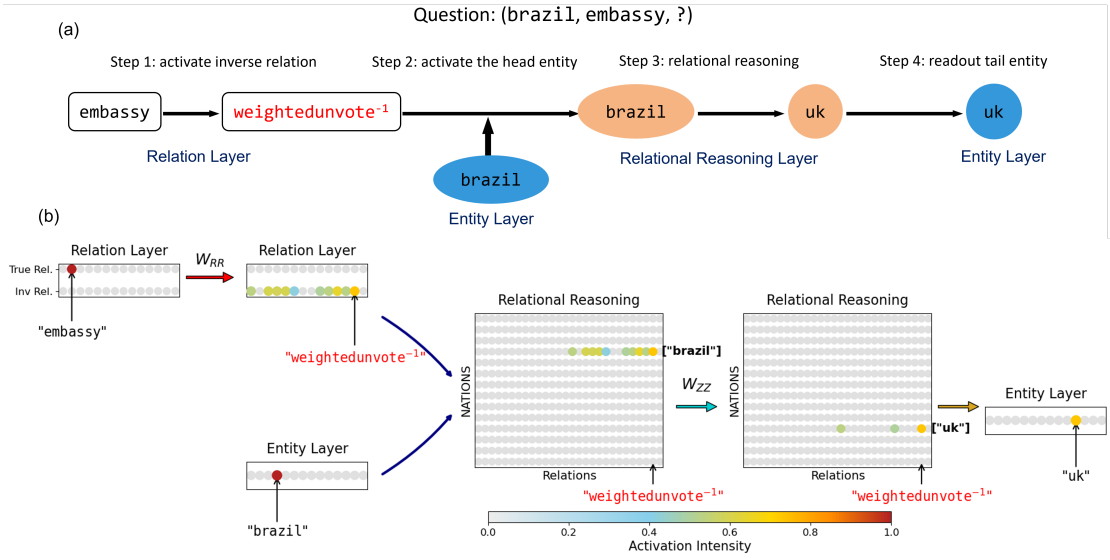


Figure 2: **Traceable reasoning in NSR. (a)** A schematic of the inferred logical path: `embassy` triggers its inverse to map `Brazil` to `UK`. **(b)** Neuronal activations at each stage of the query `(Brazil, embassy, ?)`. Activating the relation neuron `embassy` triggers the activation of candidate equivalent relations (e.g., $\texttt{weightedunvote}^{-1}$ is an associated inverse relation of `embassy`) in the L_R layer. The selected relation and head entity `Brazil` jointly project onto the L_Z layer. State transition in L_Z through recurrent dynamics leads to activation of neurons encoding the tail entity `UK`.

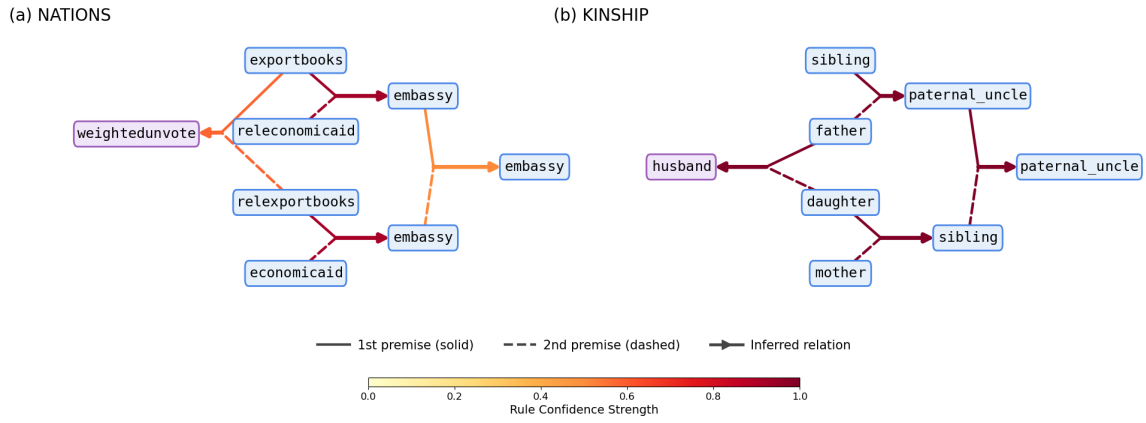


Figure 3: **Latent relational hierarchy.** Compositional rules learned by NSR, satisfying $(h, r_1, x) \wedge (x, r_2, t) = (h, r_3, t)$, on (a) Nations and (b) Kinship1990_EXTENDED. Solid lines denote r_1, dashed lines denote r_2, and color intensity indicates rule confidence.

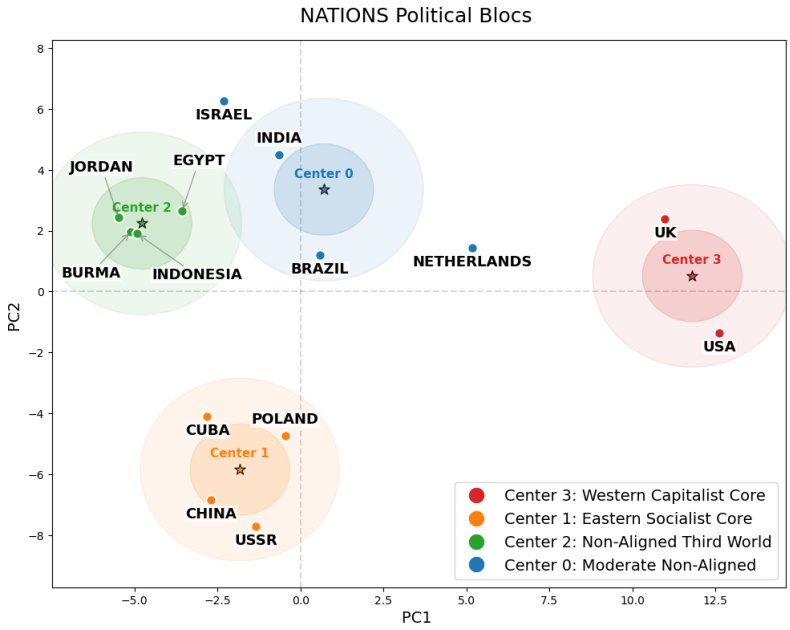


Figure 4: **Learned relational organization in Nations.** PCA projection of nations based on their relational connections in L_Z. Four clusters emerge corresponding to Cold War political blocs.

tional rules among relations and relational similarity among entities. These properties make NSR a promising step towards building efficient and interpretable models for structural reasoning.

NSR is currently evaluated on discrete triplet-structured knowledge graphs, which provide a controlled setting for structural reasoning but leave open whether the same mechanisms generalize to temporal, hyper-relational, event-based, or noisy real-world knowledge. The model also assumes that entities and relations are already symbolically specified, so future work should couple NSR with neural modules that extract relational structure from text, perception, or episodic experience. While NSR is efficient on current benchmarks, scaling compositional-rule discovery to larger graphs and longer multi-hop chains will require sparse activation, approximate retrieval, or learned proposal mechanisms. Future work should also test robustness to missing or contradictory triples and develop more biologically grounded implementations of sequence detection, gating, and confidence aggregation.

Code Availability

The source code and instructions for reproducing our experiments are publicly available at `https://github.com/navbrainlab/Neural-Structural-Reasoner-NSR`.

Acknowledgments and Disclosures

We thank Zexi Su, Hehao Qin for insightful discussions and helpful feedback on this work.

Funding. This work was supported by the National Science and Technology Major Project on Brain Science and Brain-Like Intelligence Technology (Project No. 2025ZD0217400), the CAMS Innovation Fund for Medical Sciences (CIFMS; No. 2024-RC180-02), the Beijing Key Laboratory of Brain Science and Brain-Machine Interface, and the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM504).

Competing Interests. The authors declare no competing interests.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: a python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research*, 22(82):1–6, 2021.

[3] David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 511–520. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/barrett18a.html`.

[4] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018. URL `https://arxiv.org/abs/1806.01261`.

[5] Timothy E.J. Behrens, Timothy H. Muller, James C.R. Whittington, Shirley Mark, Alon B. Baram, Kimberly L. Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2018.10.002. URL `https://www.sciencedirect.com/science/article/pii/S0896627318308560`.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[7] Ronald Brachman and Hector Levesque. *Knowledge representation and reasoning*. Elsevier, 2004.

[8] Silvia A Bunge, Carter Wendelken, David Badre, and Anthony D Wagner. Analogical reasoning and prefrontal cortex: evidence for separable retrieval and integration mechanisms. *Cerebral cortex*, 15(3):239–249, 2005.

[9] Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology*, 5(2):e1000291, 2009.

[10] Sarthak Chandra, Sugandha Sharma, Rishidev Chaudhuri, and Ila Fiete. Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature*, 638(8051):739–751, 2025.

[11] Kewei Cheng, Nesreen K Ahmed, and Yizhou Sun. Neural compositional rule learning for knowledge graph reasoning. *arXiv preprint arXiv:2303.03581*, 2023.

[12] Kalina Christoff, Vivek Prabhakaran, Jennifer Dorfman, Zuo Zhao, James K Kroger, Keith J Holyoak, and John DE Gabrieli. Rostrolateral prefrontal cortex involvement in relational integration during reasoning. *Neuroimage*, 14(5):1136–1149, 2001.

[13] Alexandra O Constantinescu, Jill X O'Reilly, and Timothy EJ Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

[14] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*, 2017.

[15] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[16] Jeffery A Dusek and Howard Eichenbaum. The hippocampus and memory for orderly stimulus relations. *Proceedings of the National Academy of Sciences*, 94(13):7109–7114, 1997.

[17] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang (Lorraine) Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 70293–70332. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/deb3c28192f979302c157cb653c15e90-Paper-Conference.pdf.

[18] Howard Eichenbaum and Neal J Cohen. *From conditioning to conscious recollection: Memory systems of the brain*. Number 35. Oxford university press, 2004.

[19] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(88)90031-5. URL https://www.sciencedirect.com/science/article/pii/0010027788900315.

[20] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422, 2013.

[21] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562*, 2023.

[22] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7 (2):155–170, 1983. ISSN 0364-0213. doi: https://doi.org/10.1016/S0364-0213(83)80009-3. URL https://www.sciencedirect.com/science/article/pii/S0364021383800093.

[23] Yu He, Yingxi Li, Colin White, and Ellen Vitercik. Can llms reason structurally? benchmarking via the lens of data structures, 2026. URL https://arxiv.org/abs/2505.24069.

[24] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4), July 2021. ISSN 0360-0300. doi: 10.1145/3447772. URL https://doi.org/10.1145/3447772.

[25] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022. doi: 10.1109/TNNLS.2021.3070843.

[26] Chen Jiang, Jiahui An, Yating Liu, and Ni Ji. Brain bandit: A biologically grounded neural network for efficient control of exploration. In *International Conference on Learning Representations*, volume 2025, pages 64153–64184, 2025.

[27] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

[28] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. doi: 10.1017/S0140525X16001837.

[29] Ni Lao, Tom Mitchell, and William Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 529–539, 2011.

[30] Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckenschmidt. Anytime bottom-up rule learning for large-scale knowledge graph completion: C. meilicke et al. *The VLDB Journal*, 33(1):131–161, 2024.

[31] Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 3104482–3104584, 2011.

[32] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[33] Guanglin Niu. Knowledge graph embeddings: A comprehensive survey on capturing relation properties. *arXiv preprint arXiv:2410.14733*, 2024.

[34] A. Paccanaro and G.E. Hinton. Learning distributed representations of concepts using linear relational embedding. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):232–244, 2001. doi: 10.1109/69.917563.

[35] Deng Pan, Simone D'Ambrogio, Naomi Kingston, Miruna Rascu, Pranav Sankhe, Shuyi Luo, Miriam C Klein-Flügge, Ali Mahmoodi, and Matthew FS Rushworth. Causal necessity of human hippocampus for structure-based inference in learning. *bioRxiv*, pages 2025–08, 2025.

[36] Seongmin A Park, Douglas S Miller, and Erie D Boorman. Inferences on a multidimensional social hierarchy use a grid-like code. *Nature neuroscience*, 24(9):1292–1301, 2021.

[37] Karalyn Patterson, Peter J Nestor, and Timothy T Rogers. Where do you know what you know? the representation of semantic knowledge in the human brain. *Nature reviews neuroscience*, 8 (12):976–987, 2007.

[38] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

[39] Alison R Preston, Yael Shrager, Nicole M Dudukovic, and John DE Gabrieli. Hippocampal contribution to the novel use of relational information in declarative memory. *Hippocampus*, 2004.

[40] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. {RNNL}ogic: Learning logic rules for reasoning on knowledge graphs. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tGZu6DlbreV.

[41] Matthew A Lambon Ralph, Elizabeth Jefferies, Karalyn Patterson, and Timothy T Rogers. The neural and computational bases of semantic cognition. *Nature reviews neuroscience*, 18(1): 42–55, 2017.

[42] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd.

[43] Veronika Samborska, James L Butler, Mark E Walton, Timothy EJ Behrens, and Thomas Akam. Complementary task representations in hippocampus and prefrontal cortex for generalizing the structure of problems. *Nature Neuroscience*, 25(10):1314–1326, 2022.

[44] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/e6acf4b0f69f6f6e60e9a815938aa1ff-Paper.pdf.

[45] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[46] Nicolas W Schuck and Yael Niv. Sequential replay of nonspatial task states in the human hippocampus. *Science*, 364(6447):eaaw5181, 2019.

[47] Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643–1653, 2017.

[48] Julia Steinberg and Haim Sompolinsky. Associative memory of structured knowledge. *Scientific Reports*, 12(1):21808, 2022.

[49] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkgEQnRqYQ.

[50] Hamza Tahir Chaudhry, Jacob A Zavatone-Veth, Dmitry Krotov, and Cengiz Pehlevan. Long sequence hopfield memory. *Journal of Statistical Mechanics: Theory and Experiment*, 2024 (10):104024, 2024.

[51] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66, 2015.

[52] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

[53] Petar Velivckovi'c, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Mikhail Dashevskiy, Raia Hadsell, and Charles Blundell. The clrs algorithmic reasoning benchmark. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:249210177.

[54] James A Waltz, Barbara J Knowlton, Keith J Holyoak, Kyle B Boone, Fred S Mishkin, Marcia de Menezes Santos, Carmen R Thomas, and Bruce L Miller. A system for relational reasoning in human prefrontal cortex. *Psychological science*, 10(2):119–125, 1999.

[55] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29 (12):2724–2743, 2017. doi: 10.1109/TKDE.2017.2754499.

[56] Taylor W. Webb, Steven M. Frankland, Awni Altabaa, Simon Segert, Kamesh Krishnamurthy, Declan Campbell, Jacob Russin, Tyler Giallanza, Randall O'Reilly, John Lafferty, and Jonathan D. Cohen. The relational bottleneck as an inductive bias for efficient abstraction. *Trends in Cognitive Sciences*, 28(9):829–843, 2024. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2024.04.001. URL https://www.sciencedirect.com/science/article/pii/S1364661324000809.

[57] Carter Wendelken, Denis Nakhabenko, Sarah E Donohue, Cameron S Carter, and Silvia A Bunge. "brain is to thought as stomach is to??": investigating the role of rostrolateral prefrontal cortex in relational reasoning. *Journal of cognitive neuroscience*, 20(4):682–693, 2008.

[58] James C.R. Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E.J. Behrens. The tolman-eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5): 1249–1263.e23, 2020. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2020.10.024. URL https://www.sciencedirect.com/science/article/pii/S009286742031388X.

[59] James CR Whittington, Joseph Warren, and Timothy EJ Behrens. Relating transformers to models and neural representations of the hippocampal formation. *arXiv preprint arXiv:2112.04035*, 2021.

[60] Shuhan Wu, Huaiyu Wan, Wei Chen, Yuting Wu, Junfeng Shen, and Youfang Lin. Towards enhancing relational rules for knowledge graph link prediction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10082–10097, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.676. URL https://aclanthology.org/2023.findings-emnlp.676/.

[61] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.

[62] Chuxu Zhang, Lu Yu, Mandana Saebi, Meng Jiang, and Nitesh Chawla. Few-shot multi-hop relation reasoning over knowledge bases. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 580–585, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.51. URL https://aclanthology.org/2020.findings-emnlp.51/.

[63] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in neural information processing systems*, 34:29476–29490, 2021.

A Details for Encoding and Learning

A.1 Learning and Encoding of KG Training Data

This subsection details how NSR stores the training graph into its connectivity and extracts pairwise relational structure. Compositional multi-hop rules are deferred to §A.2.

Network initialization. All plastic weights are initialized to zero. The only fixed connections are the bidirectional couplings between each entity neuron e_i and its associated L_Z neurons z_i^k (Eq. (1)).

Encoding and pairwise learning. Each training triple (h_i, r_k, t_j) and its inverse (t_j, r_k^{-1}, h_i) are encoded by setting the corresponding L_Z intra-layer and L_Z–L_R weights to one (Eq. (3)). Concurrently, the network records which relations co-occur on the same (h, t) pair; these co-activations drive the Oja-like updates of W_{RR} (Eq. (4)) that capture equivalence, symmetry, and inverse structure.

Algorithm 1 Learning and encoding of KG training data

Input. Training triples $\mathcal{T}_{\text{train}}$; N entities; M relations.
Parameters. Hebbian rate δ_{RR}, decay μ_{RR}.

initialize all weight matrices to 0
for $i = 1, \ldots, N$ **do**
  for $k = 1, \ldots, 2M$ **do**
    fix $W_{E \leftrightarrow Z}(e_i, z_i^k) = 1$
  end for
end for
for each $(h_i, r_k, t_j) \in \mathcal{T}_{\text{train}}$ **do**
  set $W_{ZZ}(z_i^k, z_j^{k+M}) = W_{ZZ}(z_j^{k+M}, z_i^k) = 1$
  set $W_{Z \leftrightarrow R}(z_i^k, r_k) = W_{Z \leftrightarrow R}(z_j^{k+M}, r_{k+M}) = 1$
  record r_k for pair (h_i, t_j)
end for
for each entity pair (h, t) with ≥ 2 recorded relations **do**
  for each distinct pair (r_i, r_j) in that set **do**
    $\Delta w_{ij}^{RR} \leftarrow \delta_{RR}(r_i r_j) - \mu_{RR}(r_i^2 + r_j^2) w_{ij}^{RR}$
    $w_{ij}^{RR} \leftarrow w_{ij}^{RR} + \Delta w_{ij}^{RR}$
  end for
end for

A.2 Compositional Rules Learning

The Relation Composition layer L_C contains sequence-selective neurons $\{c_{ij}\}$, each tuned to an ordered pair of extended relations $(\tilde{r}_i, \tilde{r}_j)$ with $\tilde{r} \in \{0, \ldots, 2M-1\}$ (forward $\tilde{r}_k \equiv r_k$ for $k < M$, inverse $\tilde{r}_{k+M} \equiv r_k^{-1}$). These neurons detect relational chains through asymmetric-delay coincidence detection.

Detection and encoding of relation composition. When the conditions in (5) are satisfied, assign L_C neurons c_p, c_q and c_k to encode relations r_p, r_q and r_k respectively. In addition, assign neuron $c_{\tilde{p}q}$ to be directly coupled with c_q and indirectly coupled with c_p through an intermediate neuron c_s. The activation threshold of $c_{\tilde{p}q}$ through the sigmoidal nonlinearity renders it responsive only when c_p and c_q are activated in a sequential manner. This enables $c_{\tilde{p}q}$ to serve as a sequence detection neuron in a way that is functionally analogous to direction-selective motion detection models proposed for the visual cortex.

Hebbian learning of compositional rules. When a sequence neuron c_{pq} fires upon detecting the ordered pair (r_p, r_q), and the target relation neuron r_k is concurrently activated (either by a direct triple or by its inverse through the bidirectional encoding of Eq. (3)), the coincident activity strengthens the synapse between c_{pq} and r_k via the same Oja-like rule used in L_R:

$$\Delta w_{k-pq}^{CC} = \delta_{CC}\, c_k\, c_{pq} - \mu_{CC}\,(c_k^2 + c_{pq}^2)\, w_{k-pq}^{CC}. \tag{9}$$

where the decay term effectively controls for the baseline occurrence rate of relations and chains.

The composition-detection and Hebbian mechanisms described above operate locally on individual triple chains. To discover all compositional rules supported by the data, NSR enumerates every two-hop path in $\mathcal{T}$ and applies the closed-loop check of Eq. (5) together with the weight update of Eq. (7). Algorithm 2 implements this global traversal.

Algorithm 2 Compositional-rule learning

Notation. $\mathcal{T}$: training triples (h, r, t) with $r \in \{0, \ldots, M-1\}$; extended relations $\tilde{r} \in \{0, \ldots, 2M-1\}$ encode forward ($\tilde{r}_k \equiv r_k$) and inverse ($\tilde{r}_{k+M} \equiv r_k^{-1}$) links via Eq. (3). c_{pq}: sequence neuron for (r_p, r_q). c_k: neuron in L_C representing relation r_k. δ_{CC}, μ_{CC}: Hebbian rate and decay for L_C synapses.

1: **for** each $(h_i, r_p, x_p) \in \mathcal{T}$ **do**
2: **for** each $(x_p, r_q, x_q) \in \mathcal{T}$ with $r_q \neq r_p^{-1}$ **do**
3: **for** each $(x_q, r_k, h_i) \in \mathcal{T}$ **do**
4: $c_{pq} \leftarrow$ sequence neuron for (r_p, r_q)
5: $c_k \leftarrow$ relation neuron for r_k
6: $\Delta w^{CC}_{k-pq} \leftarrow \delta_{CC}\, c_k\, c_{pq} - \mu_{CC}\,(c_k^2 + c_{pq}^2)\, w^{CC}_{k-pq}$
7: $w^{CC}_{k-pq} \leftarrow w^{CC}_{k-pq} + \Delta w^{CC}_{k-pq}$
8: **end for**
9: **end for**
10: **end for**

A.2.1 Offline learning on static graphs

On a static training graph, the per-experience updates of Algorithm 2 admit an exact closed-form evaluation, so the same connectivity W^{CC} can be computed in a single pass over the encoded weights instead of simulating the network dynamics. Two observations underlie this offline variant.

(i) **Forward episodes reduce to the encoded connectivity.** During composition learning the relation-to-relation couplings W_{RR} are disabled, so an episode initiated at head entity h under relation $\tilde{r}_i$ returns exactly the encoded neighbors of h under $\tilde{r}_i$: the dynamics of Eq. (2) reduce to readout through the fixed connectivity of Eq. (3). Writing $A_{\tilde{r}} \in \{0,1\}^{N \times N}$ for the adjacency matrix of the encoded triples under $\tilde{r}$, the two-hop experiences of a chain $(\tilde{r}_i, \tilde{r}_j)$ are therefore enumerated by the sparse product $S_{ij} = A_{\tilde{r}_i} A_{\tilde{r}_j}$, where $S_{ij}[h, t]$ counts the intermediate entities x with $(h, \tilde{r}_i, x)$ and $(x, \tilde{r}_j, t)$ both encoded.

(ii) **Per-experience updates accumulate in closed form.** For an experience (h, t) of the chain, the return readout has the closed form $y_k(h, t) = A_{r_k}[h, t]$: the neuron c_k is active exactly when (h, r_k, t) is encoded. With a running-mean learning rate $\eta_{ij} = 1/n_{ij}$, the update of lines 6–7 of Algorithm 2 is a running average over experiences, so after all n_{ij} experiences of the chain the weight equals the empirical mean

$$w^{CC}_{k-ij} \;=\; \frac{\langle A_{r_k},\, S_{ij} \rangle_F}{\sum_{h \neq t} S_{ij}[h, t]}, \qquad S_{ij} = A_{\tilde{r}_i} A_{\tilde{r}_j}, \tag{10}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product and the diagonal of S_{ij} is excluded ($t \neq h$). A constant learning rate δ_{CC} yields the standard exponentially weighted estimate of the same quantity when triples arrive sequentially.

Algorithm 3 evaluates Eq. (10) directly. It computes the same weights as the online procedure of Algorithm 2; no discrete rule set is constructed, stored, or re-grounded at query time. The evidence n_{ij} of each chain is returned together with the weights, and the thresholds τ_s and τ_c act only as readout gates during reasoning, in the same way as `T_thresh` gates relation readout in §3.4: a composition neuron c_{ij} participates in reasoning only if $n_{ij} \geq \tau_s$, and a consequent relation r_k is read out from c_{ij} only if $w^{CC}_{k-ij} \geq \tau_c$. The full dynamical implementation and this offline variant attain comparable performance on all benchmarks (§4.2).

Algorithm 3 Offline compositional-rule learning

Notation. $A_{\tilde{r}} \in \{0,1\}^{N \times N}$: adjacency matrix of encoded triples under extended relation $\tilde{r}$ (Eq. (3)). n_{ij}: number of experiences (evidence) of chain $(\tilde{r}_i, \tilde{r}_j)$. **Hyperparameters.** τ_s: minimum evidence for c_{ij} to participate in reasoning (query-time gate). τ_c: minimum weight for reading out a consequent relation (query-time gate).

$W^{CC} \leftarrow 0$
for each ordered pair $(\tilde{r}_i, \tilde{r}_j)$ with $\tilde{r}_j \neq \tilde{r}_i^{-1}$ **do**
  $S_{ij} \leftarrow A_{\tilde{r}_i} A_{\tilde{r}_j}$; set diagonal of S_{ij} to zero *// two-hop experiences*
  $n_{ij} \leftarrow \sum_{h,t} S_{ij}[h,t]$
  if $n_{ij} = 0$ **then continue**
  for $k = 1, \ldots, M$ **do**
    $w^{CC}_{k-ij} \leftarrow \langle A_{r_k}, S_{ij} \rangle_F \,/\, n_{ij}$ *// same weights as Alg. 2*
  end for
end for
return $W^{CC}, \{n_{ij}\}$

B Details for the Reasoning Phase

B.1 Reasoning phase (algorithmic form)

Algorithm 4 restates the five-step procedure of §3.4 in pseudocode. Layer updates follow the dynamics defined in Eq. (2); the final aggregation follows Eq. (8).

B.2 Hyperparameter settings

Table 3 lists the hyperparameters of the simplified variant used in §3. They are grouped by the processing stage to which they belong.

Reasoning threshold. `T_thresh` is the activation threshold of Step 1 (§3.4): a relation neuron r_e enters the equivalent-relation set $\mathcal{R}_{\text{eq}}$ only if its activation satisfies $x_e \geq$ `T_thresh`. Given that the Countries dataset contains only two relations, reasoning does not depend on relation similarity; hence, `T_thresh` is set to infinity.

Learning rate. The simplified implementation collapses the Hebbian rates δ_{RR} (Eq. (4)) and δ_{CC} (Eq. (7)) to a single scalar δ_θ, because the simplified variant performs one-pass counting instead of iterative Oja updates. The value reported in Table 3 therefore serves as the effective learning rate for both W_{RR} and W_{CC}.

Compositional-rule thresholds. τ_s and τ_c are the query-time readout gates of Algorithm 3: during reasoning, a composition neuron c_{ij} participates only if its evidence n_{ij} is at least τ_s, and a consequent relation is read out from c_{ij} only if its weight w^{CC}_{k-ij} is at least τ_c. All values were obtained with `Optuna` [1]; the search ranges were `T_thresh` $\in [0.05, 1.0]$, $\delta_\theta \in [0.1, 1.0]$, $\tau_s \in [1, 50]$, and $\tau_c \in [0.05, 0.5]$. NSR is not highly sensitive to these settings.

Table 3: Hyperparameters for the main KG benchmarks

Dataset	`T_thresh`	τ_s	τ_c	δ_θ
Nations	0.22	23	0.20	0.52
Kinship	0.90	27	0.17	0.20
Countries S3	inf	1	1.00	1.00
Kinship1990_EXTENDED	0.46	23	0.13	0.34

Algorithm 4 Reasoning phase for query $(h_i, r_k, ?)$ with max/sum aggregation.

Require: Query head h_i, query relation r_k; trained network weights and layer dynamics in Eq. (2); activation threshold `T_thresh`; gate function $G = \mathrm{TopK}$; aggregation mode $a \in \{\max, \mathrm{sum}\}$, selected on the validation set.

Ensure: Candidate-tail ranking scores $\mathrm{Score}_a(t)$.

Initialize path collections $\mathcal{P}_R \leftarrow \emptyset$ and $\mathcal{P}_C \leftarrow \emptyset$.
{**Step 1:** Retrieve equivalent relations from L_R}
Initialize all layer activations $X_E, X_Z, X_R, X_C \leftarrow \mathbf{0}$.
Clamp query relation: $X_R[r_k] \leftarrow 1$.
Update L_R once using Eq. (2).
$\mathcal{R}_{\mathrm{eq}} \leftarrow \{r_e \mid X_R[r_e] \geq \texttt{T_thresh}\}$.
Store $S_R(r_e) \leftarrow X_R[r_e]$ for each $r_e \in \mathcal{R}_{\mathrm{eq}}$.
{**Step 2:** Relational reasoning via L_Z}
Reset $X_E, X_Z \leftarrow \mathbf{0}$; retain only $\mathcal{R}_{\mathrm{eq}}$ in L_R.
Activate the head entity: $X_E[e_i] \leftarrow 1$.
Iterate L_Z and read out L_E using Eq. (2).
Collect the grounded one-hop paths $p = (h_i, r_e, t)$ realized by these transitions into $\mathcal{P}_R$.
for each $p = (h_i, r_e, t) \in \mathcal{P}_R$ **do**
    Assign path-support score $s(p) \leftarrow S_R(r_e)$.
end for
{**Step 3:** Retrieve relation compositions from L_C}
Re-initialize $X_E, X_Z, X_R, X_C \leftarrow \mathbf{0}$.
Clamp query relation: $X_R[r_k] \leftarrow 1$.
Propagate activity to L_C and update its dynamics using Eq. (2).
$\mathcal{C}_{\mathrm{eq}} \leftarrow G(X_C) = \mathrm{TopK}(X_C)$.
Store $S_C(c_e) \leftarrow X_C[c_e]$ for each $c_e \in \mathcal{C}_{\mathrm{eq}}$.
{**Step 4:** Multi-hop reasoning with each composition}
for each $c_e \in \mathcal{C}_{\mathrm{eq}}$ **do**
    Reset $X_E, X_Z, X_R, X_C \leftarrow \mathbf{0}$.
    Activate c_e in L_C and initialize L_E at e_i.
    Decode c_e into the ordered relation chain $(r_{p_1}, r_{p_2}, \ldots, r_{p_\ell})$.
    for $j = 1, \ldots, \ell$ **do**
        Activate r_{p_j} in L_R.
        Iterate L_Z and read out L_E using Eq. (2), retaining distinct grounded transition traces.
        Use the reached entities as the starting states for the next hop, if any.
    end for
    Collect the completed grounded paths $p = (h_i, r_{p_1}, x_1, \ldots, r_{p_\ell}, t)$ into $\mathcal{P}_{c_e}$.
    for each $p \in \mathcal{P}_{c_e}$ **do**
        Assign path-support score $s(p) \leftarrow S_C(c_e)$.
    end for
    $\mathcal{P}_C \leftarrow \mathcal{P}_C \cup \mathcal{P}_{c_e}$.
end for
{**Step 5:** Max/sum aggregation of path-support scores}
$\mathcal{P} \leftarrow \mathcal{P}_R \cup \mathcal{P}_C$.
$\tilde{\mathcal{T}} \leftarrow \{t \mid \exists p \in \mathcal{P}$ with terminal entity $t\}$.
for each $t \in \tilde{\mathcal{T}}$ **do**
    $\mathcal{P}(t) \leftarrow \{p \in \mathcal{P} \mid p$ has terminal entity $t\}$.
    if $a = \max$ **then**
        $\mathrm{Score}_a(t) \leftarrow \max_{p \in \mathcal{P}(t)} s(p)$.
    else
        $\mathrm{Score}_a(t) \leftarrow \sum_{p \in \mathcal{P}(t)} s(p)$.
    end if
end for
return Score_a.

C Datasets and Detailed Experiments

C.1 Datasets

We evaluate NSR on knowledge-graph datasets of varying scale and relational complexity. Nations, Kinship, and Countries S3 are standard benchmarks in the KG-embedding literature. Kinship1990_EXTENDED is an extended derivative of the classical Kinship domain that we introduce to stress-test compositional relational reasoning; its construction is detailed in § C.2. WN18RR, FB15k-237, and YAGO3-10 are standard large-scale benchmarks. Table 4 summarizes the number of entities, relation types, and triples (edges) in each split.

Table 4: Statistics of the datasets used in our experiments. All counts in the last five columns refer to the number of edges (triples).

Dataset	Entities	Relations	Total	Train	Val	Test
Nations	14	55	1,992	1,592	199	201
Kinship	104	26	10,686	8,544	1,068	1,074
Kinship1990_EXTENDED	480	14	2,240	1,568	224	448
Countries_S3	271	2	1,033	985	24	24
WN18RR	40,943	11	93,003	86,835	3,034	3,134
FB15k-237	14,541	237	310,116	272,115	17,535	20,466
YAGO3-10	123,182	37	1,089,040	1,079,040	5,000	5,000

C.2 How to Construct Kinship1990_EXTENDED

Original data and its limitations. The original Kinship dataset [34] contains 112 triples over 24 persons from two isomorphic families, with 12 binary relations: `wife`, `husband`, `mother`, `father`, `daughter`, `son`, `sister`, `brother`, `aunt`, `uncle`, `niece`, and `nephew`. Three features limit its utility for testing compositional reasoning: (i) *No symmetric relations*—every relation is directional, so link-prediction can be solved by learning inverses alone; (ii) *Flat relation labels*—genealogical concepts such as `aunt` or `uncle` are provided as atomic symbols rather than explicit compositions of primitives (e.g., $\texttt{mother} \circ \texttt{sister}$), so models are not required to discover relational grammar; (iii) *Small scale*—112 triples provide insufficient coverage to stress-test generalization.

Reconstructing the base graph. Because the two families have identical structure, the complete genealogical tree is fully determined by the original triples. We first reconstruct this tree, obtaining all parent–child and spousal links.

Injecting symmetric relations. We add `sibling` as an explicitly symmetric relation. For every pair of distinct children sharing at least one parent in the reconstructed tree, we insert $\texttt{sibling}(X, Y)$ and $\texttt{sibling}(Y, X)$. Unlike the original gender-specific `brother` and `sister`, `sibling` forces the model to respect an equivalence constraint that cannot be reduced to a directional inverse.

Extracting compositional chain relations. We define five new relation types by enumerating multi-hop genealogical paths in the reconstructed tree and abstracting them into explicit labels:

- `grandmother_chain`: the mother of one's father or mother;
- `grandfather_chain`: the father of one's father or mother;
- `maternal_aunt_chain`: the sister of one's mother;
- `paternal_uncle_chain`: the brother of one's father;
- `cousin_chain`: the child of one's parent's sibling.

For example, $\texttt{cousin_chain}(h, t)$ is generated whenever there exist intermediates x, y such that x is a parent of h, y is a sibling of x, and t is a child of y. These chain relations are not redundant with the original flat labels; they make compositional substructure explicit and therefore require the model to reuse primitive relations rather than memorize atomic mappings.

Statistics and splits. The extended graph contains 480 entities, 14 relation types, and 2,240 triples. We follow the protocol: 70% train, 10% validation, 20% test. Because the underlying family tree is small and structurally deterministic, all base-relation edges necessary for genealogical consistency appear in the training split; held-out test triples are drawn predominantly from the extended relation set so that accurate prediction benefits from compositional reuse of the trained primitives.

C.3 Detailed Experiments

Embedding-Based baseline implementations. All neural baselines are trained with PyKEEN v1.10 [2] under identical data splits. Table 5 lists the architecture and optimization settings shared across embedding models. The embedding dimension is set to 100 for small-scale datasets (e.g., Nations and Kinship) and to 200 for large-scale datasets (e.g., WN18RR, FB15k-237 and YAGO3-10). DistMult, ComplEx, RESCAL, and ConvE use the LCWA training loop; TransE and RotatE use sLCWA. ConvE requires inverse triples (create_inverse=True) per PyKEEN's implementation and uses 32 output channels with dropout rates $\{0.2, 0.2, 0.3\}$. The optimizer is Adam with learning rate 10^{-3} for all models except RotatE (5×10^{-4}). Training runs for at most 150 epochs with batch size 32.

Table 5: Neural baseline hyperparameters and training configuration.

Model	Dim	Loop	Inverse	LR	Epochs	Special
TransE	100	sLCWA	No	1e−3	150	scoring_fct_norm=1
DistMult	100	LCWA	No	1e−3	150	—
ComplEx	100	LCWA	No	1e−3	150	—
RotatE	100	sLCWA	No	5e−4	150	—
ConvE	100,200	LCWA	Yes	1e−3	150	out_ch=32, dropouts $\{0.2, 0.2, 0.3\}$
RESCAL	100	LCWA	No	1e−3	150	—

Rule Learning baseline implementations. RNNLogic operates *without pretrained knowledge-graph embeddings*. It first mines relational paths up to length 3 using the C++ miner (16 threads), then learns rule weights via a lightweight predictor trained for 10 iterations (Adam, learning rate 5×10^{-3}, weight decay 0, hidden dimension 32, batch size 16). Label smoothing (0.1) is applied during training, and expectation-based ranking (`expectation=True`) is used at evaluation. Unlike neural baselines, no entity or relation embeddings are supplied or learned.

Reproducibility. We used five random seeds: [42, 43, 44, 45, 46], and the performance of these baseline models and NSR is reported in § 4.2. PyKEEN handles its own internal seeding for negative sampling and parameter initialization. NSR and all baselines are evaluated on identical train/validation/test splits for each dataset.

Table 6: Symbolic baseline hyperparameters and training configuration.

Model	Rule Len	Hidden Dim	LR	Iters	Special
RNNLogic (Miner)	3	—	—	—	16 threads
RNNLogic (Predictor)	—	32	5e−3	10	batch=16, smoothing=0.1, expectation=True, **without embedding**

C.4 Additional Benchmark Results

We report three complementary evaluations deferred from the main text: Countries S3, which probes multi-hop chain generalization; Kinship1990_EXTENDED, our controlled compositional stress test (§C.2); and WN18RR, a standard large-scale benchmark. We then compare NSR against eight symbolic, neuro-symbolic, and neural reasoning baselines across all five benchmarks, and report training and inference costs.

Countries S3. Test queries ask for a country's continent, but training triples only link neighboring countries and upward `locatedIn` edges; no country–continent fact is observed directly, so answering

requires composing `neighborOf` and `locatedIn` chains. Table 7 shows that NSR solves the task perfectly, while embedding baselines collapse (best MRR 0.199) because static vector similarity cannot recover unobserved multi-step paths, and RNNLogic reaches only 0.341 because externally mined templates miss the compositional regularity.

Table 7: Performance comparison on Countries S3. Best results in each column are highlighted in bold. Hits@k are reported as percentage values in $[0, 100]$; standard deviations are in units of 10^{-2} (MRR) and percentage points (Hits@k).

Model	MRR ↑	Hits@1 ↑	Hits@3 ↑	**Train Time (s)**
ConvE	0.1987±2.47	11.25±2.50	19.17±4.04	20.3±2.7
DistMult	0.1761±1.42	7.50±1.02	20.42±3.06	17.4±1.0
RotatE	0.1184±2.42	4.17±1.86	10.00±4.45	19.4±2.9
RESCAL	0.1906±1.00	10.42±2.64	18.33±2.04	14.7±2.1
ComplEx	0.0290±2.68	0.83±1.67	2.08±4.17	18.4±2.2
TransE	0.1182±0.61	0.00±0.00	14.17±2.76	18.9±1.7
RNNLogic	0.3409±3.29	0.00±0.00	62.50±12.36	5.1±0.0
NSR	**1.0000**±0.00	**100.00**±0.00	**100.00**±0.00	**0.0012**±0.0004

Kinship1990_EXTENDED. On the composition-focused extension, test triples are drawn predominantly from the five composition-defined chain relations (§C.2). NSR attains the best MRR (0.953) and Hits@1 (94.3), surpassing all baselines including DistMult, which is strong on MRR (0.941) but relies on memorized symmetric structure; the inversion relative to classical Kinship, where ConvE leads, is discussed in §4.2.

Table 8: Performance comparison on Kinship1990_EXTENDED. Conventions as in Table 7.

Model	MRR ↑	Hits@1 ↑	Hits@3 ↑	**Train Time (s)**
ConvE	0.8265±0.91	74.93±1.47	87.83±0.72	60.5±7.5
DistMult	0.9409±0.15	90.62±0.30	**97.75**±0.25	31.9±4.0
RotatE	0.5988±4.47	51.94±5.27	64.40±4.20	26.0±1.1
RESCAL	0.0144±0.28	0.22±0.12	0.80±0.31	30.3±5.3
ComplEx	0.0220±0.34	0.42±0.23	1.41±0.50	31.4±1.6
TransE	0.2249±0.58	2.19±0.69	33.42±1.58	25.3±2.4
RNNLogic	0.8107±1.82	80.45±1.92	81.38±1.78	9.3±1.4
NSR	**0.9533**±0.30	**94.33**±0.48	96.47±0.09	**0.0451**±0.0002

WN18RR. Table 9 reports full filtered metrics on WN18RR, together with the symbolic and neuro-symbolic baselines (MRR; see Table 10 for the remaining datasets). NSR attains the best Hits@1 (0.463) among all evaluated methods and is competitive with the strongest embedding models on MRR, while the GNN reasoner NBFNet remains stronger; we view NSR as competitive rather than uniformly superior on this benchmark.

Comparison with symbolic and neuro-symbolic reasoners. Table 10 compares NSR against eight reasoning baselines spanning symbolic rule mining (AnyBURL, AMIE), path ranking (PRA/PathRank), differentiable rule learning (NTP, NeuralLP, NCRL), RL-based path search (MINERVA), and GNN reasoning (NBFNet) on all five benchmarks. NSR outperforms the evaluated differentiable rule learners on every dataset they complete, performs comparably to strong symbolic rule miners overall, and achieves the best result on YAGO3-10, while NBFNet remains stronger on WN18RR and FB15k-237 and AMIE is strongest on Nations. Because the three large benchmarks are heterogeneous and not designed around NSR's compositional structure, these results indicate that NSR's performance is not confined to small, composition-aligned benchmarks.

Training and inference cost. Table 11 reports end-to-end training time and full-test-set inference time. NSR trains in seconds to minutes across all benchmarks—orders of magnitude below iterative neural training on the large datasets—while inference cost remains modest; measurements on baselines span different hardware and stopping criteria, so we report them as indicative rather than as strict multiplicative speedups.

Table 9: Filtered link prediction on WN18RR. Symbolic and neuro-symbolic baselines report MRR only; "—" indicates unreported or not scalable. Best result in each column in bold.

Method	MRR	Hits@1	Hits@3	Hits@10
TransE	.226	—	—	.501
DistMult	.430	.390	.440	.490
ConvE	.430	.400	.440	.520
ComplEx	.440	.410	.460	.510
RotatE	.476	.428	.492	.571
BoxE	.451	.400	.472	.541
ModE	.472	.427	.486	.564
HAKE	.497	.452	**.516**	.582
SectorE	.475	.421	.478	**.586**
AnyBURL	.5658	—	—	—
AMIE	.4157	—	—	—
PRA / PathRank	.0556	—	—	—
NTP	—	—	—	—
NeuralLP	.4677	—	—	—
NCRL	.4070	—	—	—
MINERVA	.4890	—	—	—
NBFNet	**.5976**	—	—	—
NSR	.472	**.463**	.483	.484

Table 10: Link prediction results (MRR) on all KG completion benchmarks. Best results are in **bold**; asterisks (*) denote values taken from prior work or official repositories, and "—" indicates results not reported or the method does not scale to the dataset.

Method	**Nations**	**Kinship**	**WN18RR**	**YAGO3-10**	**FB15k-237**
AnyBURL	0.7994	0.6768	0.5658	0.5589	0.332*
AMIE	**0.8559**	0.6767	0.4157	0.5473	0.2170
PRA / PathRank	0.5933	0.6296	0.0556	0.4678	0.0972
NTP	0.6223	0.612*	—	—	—
NeuralLP	0.6841	0.6072	0.4677	—	0.3166
NCRL	0.4571	0.6050	0.4070	0.380*	0.300*
MINERVA	0.5865	0.6253	0.4890	—	0.2734
NBFNet	0.7479	**0.7445**	**0.5976**	0.4946	**0.5114**
NSR	0.8142	0.6515	0.4720	**0.5893**	0.3649

Table 11: Training time / full-test-set inference time (seconds unless noted). **quoted from the official AnyBURL-23-1 release; other local runs use the same machine. NBFNet entries use official training profiles, so cross-hardware comparisons are indicative. OOM: out of memory.

Method	**Nations**	**Kinship**	**WN18RR**	**YAGO3-10**	**FB15k-237**
AnyBURL	64 s / 16 s	63 s / 36 s	603 s / 11 s	1000 s / 424 s	1000 s** / —
AMIE	2246 s / <0.1 s	2 s / 0.3 s	2 s / <0.1 s	83 s / 0.6 s	8 s / 1.2 s
PRA / PathRank	34 s / 0.5 s	86 s / 0.3 s	22 s / 156 s	239 s / 354 s	229 s / 841 s
NTP	1.9 h / 8 s	—	—	—	—
NeuralLP	40 s / 8 s	26 s / 4 s	1.1 h / 129 s	— (OOM)	11.9 h / 533 s
NCRL	134 s / 43 s	99 s / 7 s	244 s / 24 s	—	—
MINERVA	3160 s / 8 s	1.2 h / 13 s	3416 s / 44 s	—	3.3 h / 333 s
NBFNet	29 s / 0.1 s	77 s / 0.3 s	5.6 h / 12 s	2.3 h / 422 s	10.4 h / 46 s
NSR	3 s / 0.4 s	11 s / 1.2 s	3.3 s / 32 s	1142 s / 107 s	2019 s / 55 s

C.5 Ablation Study

We ablate the four learnable or structural components of NSR to isolate their individual contributions: inverse-relation encoding (Eq. (3)), relation-equivalence retrieval (Step 1 of §3.4), compositional inference (Steps 3–4 of §3.4), and Hebbian learning (Eqs. (4) and (7)). All ablations are conducted with the full dynamical model on Nations, Kinship, and Kinship1990_EXTENDED, under the filtered link-prediction protocol of §4.1. Each variant is evaluated on five paired seeds; we report MRR (mean $\pm$ population standard deviation across seeds) together with ΔMRR relative to the paired full model. Ties between equally scored entities are broken by seed-dependent uniform noise of magnitude 10^{-6}.

Table 12: Component-wise ablations of NSR (MRR, mean $\pm$ std over five seeds). Values in parentheses denote ΔMRR relative to the paired full model. Best results in each column are highlighted in bold.

Variant	Nations	Kinship	Kinship1990_EXTENDED
Full model	**0.81**$\pm$0.03	**0.65**$\pm$0.01	**0.95**$\pm$0.00
w/o inverse encoding	0.66$\pm$0.03 (-0.15)	0.05$\pm$0.00 (-0.60)	0.10$\pm$0.00 (-0.85)
w/o relation-equivalence retrieval	0.61$\pm$0.01 (-0.20)	0.42$\pm$0.00 (-0.24)	0.82$\pm$0.00 (-0.12)
w/o compositional inference	0.79$\pm$0.03 (-0.03)	0.48$\pm$0.01 (-0.17)	0.46$\pm$0.00 (-0.48)
w/o Hebbian learning	0.36$\pm$0.01 (-0.45)	0.05$\pm$0.00 (-0.60)	0.02$\pm$0.00 (-0.93)

The variants are defined as follows. *W/o inverse encoding*: the model is retrained from scratch with the inverse half of the encoding in Eq. (3) disabled—inverse triples (t, r^{-1}, h) are never encoded and the inverse relation neurons $\{r_{i+M}\}$ of L_R are unused; both learning phases are otherwise unchanged. *W/o relation-equivalence retrieval*: training is identical to the full model and the same trained network is reused; only the reasoning phase is modified, with the W_{RR} propagation of Step 1 in §3.4 disabled at query time, so a query is answered using only the query relation itself and its associated compositions. *W/o compositional inference*: again the same trained network as the full model is reused, and the L_C pathway (Steps 3–4 of §3.4) is disabled at query time, so candidate tails come exclusively from equivalent-relation paths $\tilde{\mathcal{T}}^{(1)}$. *W/o Hebbian learning*: the encoding phase (Eq. (3)) proceeds as in the full model, but both Hebbian learning phases are skipped, so W_{RR} and W^{CC} remain at their zero initialization; the reasoning pipeline is run unchanged.

Findings. The degradation pattern is functionally specific rather than uniform, showing that each component serves a distinct role. Inverse encoding is decisive on the kinship benchmarks (-0.60 and -0.85 MRR): directed kinship relations such as `son_of` or `grandmother_chain` can only be traversed through their inverses, whereas Nations (-0.15) usually offers an equivalent forward relation as a detour. Relation-equivalence retrieval contributes consistently across all datasets (-0.12 to -0.24), confirming that the learned W_{RR} couplings provide a reliable first source of candidate relations at query time. Compositional inference matters most on Kinship1990_EXTENDED (-0.48), whose test queries require unseen multi-hop compositions by construction, and least on Nations (-0.03), where most test queries are solvable by single-hop equivalence—mirroring the intended difference between the two benchmarks. Finally, disabling Hebbian learning causes the largest overall degradation (-0.45, -0.60, -0.93), showing that the learned associative connectivity is essential to performance rather than a merely decorative biological motif. The small residual MRR of this variant is expected: with W_{RR} and W^{CC} at zero, no equivalent relation or composition can be retrieved, entity scores degenerate, and ranks are effectively determined by tie-breaking noise, whose expected reciprocal rank is nonzero on small entity sets.

We stress the scope of this evidence: these ablations validate the *computational* role of each implemented component. They are not evidence for or against the biological realism of the corresponding mechanisms, which we present as computational inspiration.

Aggregation rule. We also ablate the max aggregation of Eq. (8) against sum aggregation. The two rules exhibit a genuine trade-off: max aggregation can be dominated by a single strong but spurious path, whereas sum aggregation pools convergent evidence but may overcount correlated paths. For example, on the Kinship query (`person80`, `term16`, ?), the target `person25` has no direct evidence; under max aggregation a competitor (`person32`, score 0.662) slightly outscores the target (0.641), while under sum aggregation the target accumulates support from 132 grounded paths (total 46.23, versus 1.55 from 3 paths for the competitor) and becomes the unique top-ranked answer. We

therefore select the aggregation mode on validation data and treat NSR's outputs as ranking scores rather than calibrated probabilities.

D Additional Analyses and Reasoning Traces

D.1 Symmetry Reasoning Trace

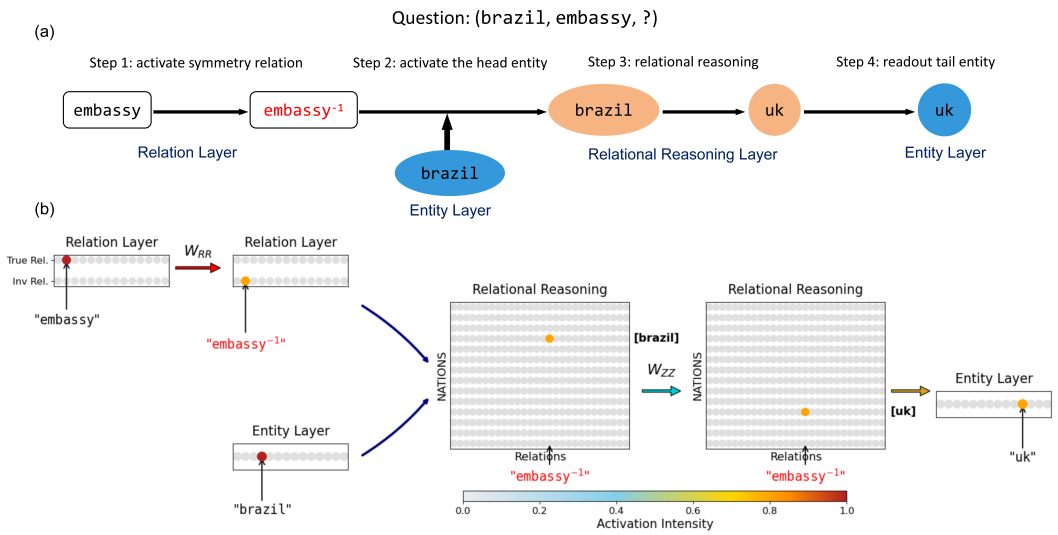


Figure 5: **Symmetry-reasoning trace (symmetry branch only).** In Path A, symmetry and inverse evaluations are performed in parallel. The complete trace therefore contains two branches: a symmetry branch that retains the original relation neuron, and an inverse branch that activates the inverse-relation neuron. For visual clarity, only the symmetry branch is shown here; the inverse branch follows the same gated-diffusion and binding structure with relation inversion.

Figure 5 illustrates the activation trajectory for symmetry reasoning on the query `(Brazil, embassy, ?)`. During the associative phase, the query relation `embassy` diffuses through W_{RR}, and the complementary gate promotes both symmetric and inverse counterparts in parallel. The full inference thus comprises two structurally isomorphic branches: one retaining the original relation `embassy` (symmetry) and one activating $\texttt{embassy}^{-1}$ (inverse). Because the two branches differ only in whether the relation neuron is inverted, we display only the symmetry branch for clarity; the inverse branch can be obtained by following the same pipeline with the W_{RR} dynamics.

D.2 Latent Relational Structure of Countries

The Countries dataset is built from only two primitive relations, `locatedIn` and `neighborOf`, distributed over three entity strata: countries, regions, and continents. Crucially, `neighborOf` edges are observed *only* between country-level entities; the dataset contains no direct adjacency between regions or between continents, even though such adjacencies are clearly entailed by the underlying geography. Recovering these missing higher-order neighborhoods therefore provides a natural probe of NSR's compositional generalization, requiring the model to abstract the same adjacency relation across the entity hierarchy rather than within a fixed layer.

Two chained rules suffice to lift adjacency upward across strata:

$$\texttt{nb}(c_a, c_b) \wedge \texttt{loc}(c_a, R_a) \wedge \texttt{loc}(c_b, R_b) \Rightarrow \texttt{nb}(R_a, R_b), \tag{11}$$

$$\texttt{nb}(R_a, R_b) \wedge \texttt{loc}(R_a, K_a) \wedge \texttt{loc}(R_b, K_b) \Rightarrow \texttt{nb}(K_a, K_b), \tag{12}$$

where $\texttt{nb} \equiv \texttt{neighborOf}$ and $\texttt{loc} \equiv \texttt{locatedIn}$; $c_\bullet$, $R_\bullet$, and $K_\bullet$ index countries, regions, and continents respectively.

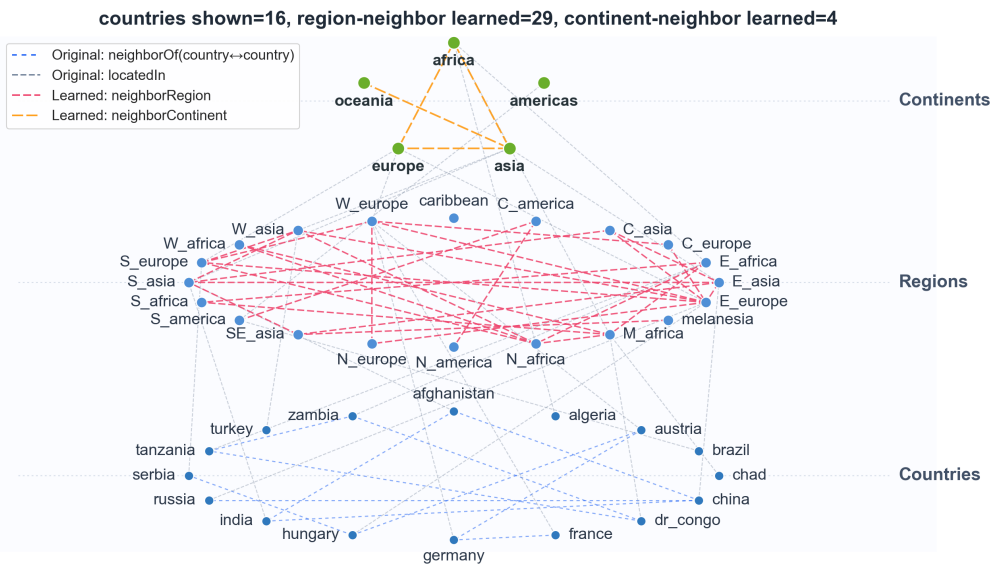


Figure 6: **Discovery of latent `neighborOf` on Countries.** Blue and gray dashed edges denote the original `neighborOf` and `locatedIn` facts; red and orange dashed edges denote the region- and continent-level adjacencies induced by NSR. Region prefixes abbreviate cardinal directions: N (northern), S (southern), E (eastern), W (western), C (central), M (middle), SE (southeastern). Only the 16 most active countries are shown; all regions and continents are displayed in full.

Figure 6 renders the dataset as a three-tier layered graph. Without ever observing a single direct edge of either type during training, NSR recovers 29 region-neighbor and 4 continent-neighbor edges, and the induced adjacencies are geographically coherent: the continent layer assembles the expected Africa–Europe–Asia triangle together with an Asia–Oceania link, while the region layer reproduces dense intra-Europe, intra-Africa, and intra-Asia neighborhoods (e.g., `western_europe` adjoining `northern_europe` and `southern_europe`).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state four main contributions, each of which is substantiated in the corresponding sections of the paper: (1) a formulation of network structure as relational structure, presented in Section 3.1 (Model Architecture); (2) a parallel reasoning mechanism with path-support scores, introduced in Section 3 and detailed in Appendix A.2; (3) competitive performance with high computational efficiency, demonstrated empirically in Section 4 (Experiments); and (4) interpretability arising from the model's dynamics, illustrated through traceable reasoning steps in Section 4.3 and further analyzed in Appendix D.1. The scope of these claims is restricted to the experimental settings described in Section 4

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the Conclusion (Section 5). Specifically, we acknowledge that: (1) NSR has so far been evaluated only on knowledge graphs with strict triplet structure, and it remains open whether its mechanisms generalize to richer formalisms such as temporal, hyper-relational, or event-based knowledge bases; and (2) the method relies on precisely structured relational input, so its applicability to settings where relational associations must be extracted directly from natural text or non-textual episodic experience has yet to be demonstrated. We frame these limitations as concrete directions for future work, moving from a dedicated relational reasoner toward a more general model of structured intelligence.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results such as theorems, lemmas, or propositions requiring proof. All mathematical content is presented as constructive model and algorithm definitions. The contributions of NSR are architectural and empirical rather than theoretical, and are validated through the experiments in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the information necessary to reproduce the main experimental results. The NSR architecture is fully specified in § 3, including the layer organization (§ 3.1), the network dynamics (Eq. (2)), the encoding rules (Eq. (3)), the Hebbian learning procedure (Eqs. (4)–(7)), and the inference pipeline (§ 3.4). Additional implementation details are given in § A.1 and § A.2 (relational associativity and compositional rule extraction) and in § B (reasoning phase pseudocode and the full hyperparameter table, Table 3). Standard datasets are taken from the publicly available PyKEEN library under the standard filtered link-prediction protocol, and the construction of Kinship1990_EXTENDED is described step by step in § C.2. Baseline configurations (architectures, optimizers, learning rates, training loops, early-stopping criteria) and evaluation details are documented in § C.3 (Table 5). All experiments are run with five independent random seeds [42, 43, 44, 45, 46] and identical train/validation/test splits across methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The source code and instructions for reproducing the main experiments are publicly available at our GitHub repository. The paper additionally documents the model architecture, learning procedures, evaluation protocol, and dataset construction in Sections 3–4 and Appendices A–C.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All training and test details are specified in the paper. The datasets and the filtered link-prediction protocol are described in § 4.1, with the construction of Kinship1990_EXTENDED detailed in § C.2. NSR's hyperparameters for each benchmark are listed in § B.2, together with the Optuna search ranges used to obtain them and a brief sensitivity analysis. Baseline architectures and optimization settings (embedding dimension, training loop, inverse-triple usage, learning rate, batch size, epochs, and model-specific options) are documented in § C.3. The early-stopping criterion (PyKEEN's built-in early stopper with validation MRR, patience 10, relative delta 0.002), the evaluation procedure (PyKEEN's RankBasedEvaluator with filtering), and the use of five independent random seeds (42, 43, 44, 45, 46) are also reported in § C.3. Final results are reported as mean $\pm$ standard deviation across these five runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report variability across five runs for the additional experiments and ablation studies where applicable (e.g., Tables 7, 8, and 12). The main benchmark tables report point estimates, with experimental protocols detailed in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: All experiments were conducted on a single workstation equipped with an Intel Core Ultra 9 285K CPU (24 cores), 192 GB of system RAM, and a single NVIDIA RTX Pro6000 GPU (96 GB VRAM) running Linux and PyTorch. For NSR, the most time-consuming step is hyperparameter tuning via `Optuna` on the dataset.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that this research conforms with it in every respect. The work uses only publicly available, standard knowledge-graph benchmarks (Nations, Kinship, Countries) loaded through the PyKEEN library, together with Kinship1990_EXTENDED, which is constructed from the publicly released Kinship dataset and contains no personal, sensitive, or identifying information. The study does not involve human subjects, crowdsourcing, or private data, and the proposed model targets symbolic knowledge-graph reasoning rather than any application with foreseeable dual-use or surveillance risk.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive societal impacts of our work, particularly regarding AI transparency, reliability, and safety. A key contribution of the Neural Structural Reasoner (NSR) is its intrinsic interpretability: unlike conventional "black-box" embeddings that produce opaque scores, NSR's inference is driven by discrete, human-readable network dynamics. This ensures that every prediction is accompanied by a fully traceable reasoning path, allowing users to audit the exact sequence of logical steps and pinpoint precisely where an error may have occurred. We thoroughly analyze and demonstrate these traceable reasoning steps in Section 4.3, with further step-by-step visualizations provided in Appendix D.1. By demonstrating that high reasoning performance and computational efficiency can be achieved without sacrificing transparency, our work contributes to the development of safer, more accountable, and easily verifiable AI systems. We do not foresee direct negative societal impacts or dual-use risks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks. NSR is a small symbolic reasoning model over public knowledge-graph benchmarks; it does not generate text, images, or any other content, does not identify individuals, and does not make autonomous decisions. The datasets used (Nations, Kinship, Countries, and Kinship1990_EXTENDED) are standard, publicly available academic benchmarks containing only structured relational triples with no scraped or potentially unsafe content. Kinship1990_EXTENDED is constructed deterministically from the publicly released Kinship dataset and contains no personal or sensitive information.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets used in this work are properly credited and used in accordance with their licenses. The PyKEEN library v1.10 (MIT License) is used for dataset loading, baseline implementations, and evaluation; Optuna (Apache License 2.0) is used for hyperparameter search. All baseline models are cited to their original publications: TransE, DistMult, ComplEx , RotatE, RESCAL, and ConvE. The benchmark datasets (Nations, Kinship, Countries) are standard knowledge-graph benchmarks distributed through PyKEEN; the original Kinship dataset is credited to Kemp et al. Our re-engineered Kinship1990_EXTENDED is constructed deterministically from the publicly released Kinship dataset and follows its terms of use; the construction procedure is fully documented in § C.2.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces two new assets, both of which are documented within the paper. (1) Kinship1990_EXTENDED is a re-engineered extension of the original Kinship dataset. Its motivation, construction procedure (genealogical-tree reconstruction, injection of symmetric relations, and extraction of five new compositional chain relations), full statistics (480 entities, 14 relation types, 2,240 triples), and 70/10/20 train/validation/test split are documented in § C.2. The dataset is derived deterministically from the publicly released Kinship dataset and inherits its terms of use; it contains no personal or sensitive information. (2) The NSR model is fully specified in § 3 (architecture, dynamics, and learning rules).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. Neuroscience findings cited in Section 2 are drawn from previously published literature.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable. All experiments are conducted on existing publicly available knowledge-graph benchmarks, with no human participants and no data collection from human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as an important, original, or non-standard component. NSR is a bio-plausible network architecture trained with Hebbian-style local learning rules; all baselines are classical knowledge-graph embedding and convolutional models (TransE, DistMult, ComplEx, RotatE, RESCAL, ConvE), and no language model is used for data construction, evaluation, or any part of the proposed pipeline.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.