NEURAL-SYMBOLIC REPRESENTATION LEARNING BY STRUCTURAL ABSTRACTION FROM NOISY DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

How can neural perception be interfaced with symbolic reasoning when primitive symbols and their structure are not given in advance? Palaeographers infer how oracle-bone characters are composed by analysing their shapes and possible strokes, which are noisy demonstrations composed by a sequence of raw perceptual observations without any symbolic labels or predefined structures. In this case, the learner must abstract both the primitive symbols and infer how they compose, unlike traditional neural-symbolic (NeSy) learning tasks which usually assume a manually specified symbols of primitives and relations as the interface. We study this representation-learning problem and introduce structural abstraction, an algorithm for learning the interface from such demonstrations. On few-shot handwritten oracle-bone characters, it recovers expert-annotated radical substructures with 72.28% macro-averaged recall, compared with 9.58–13.09% for feature-based motif-discovery baselines. The learned representations extrapolate to unseen ancient and modern Chinese characters and discovers novel radicals at the same time. Our results show that perceptually grounded symbols and their compositional structure can be learned from demonstrations.

1 INTRODUCTION

Neural-symbolic (NeSy) AI aims to combine neural models that learn from perceptual data with symbolic representations that support explicit reasoning (De Raedt et al., 2025). The central challenge is *the interface learning*: perceptual states must acquire symbolic identities and relations that a reasoner can use, while learned representations must capture concepts across levels of abstraction (Mitchell, 2021; Muttenthaler et al., 2025). The same issue arises in Large Language Model (LLM) or reinforcement learning (RL) agents for complex long-horizon tasks that requires sound symbolic verification: the verifier needs an explicit vocabulary and a mapping from raw perceptual input to symbols and formulae in a first-order theory. Yet most NeSy approaches assume that the symbolic vocabulary and relations are specified in advance, leaving open how to learn this interface when the primitive symbols and their structure are unknown (Wang et al., 2025).

For example, the interface problem becomes concrete in palaeographic analysis and decipherment of oracle-bone characters, which were used in 14th˜11th centry BC in China. Consider the two characters  and . An experienced palaeographer can guess and trace the characters as strokes; for example, tracing yields the candidate stroke sequence ⟨ , , , , , , , , , ⟩. A natural local assumption of the palaeographers tells them that the first six red strokes which form the upper parts of the two characters could be a meaningful radical, helping them parse into a set with two elements { , }. Analysing a new character provides further evidence that is also a meaningful component which should be remembered, since it appears in the analysed .

In this analytical learning process, however, the whole-character images and stroke sequences provide only noisy perceptual evidence: *they do not label which strokes share a symbolic identity or specify how those identities compose*. As in a step-by-step demonstration to a child learning to write, the learner must abstract the reusable symbols visually, and their structure from demonstrations; the result of analytical learning is a representation that is both *perceptural and structural*.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

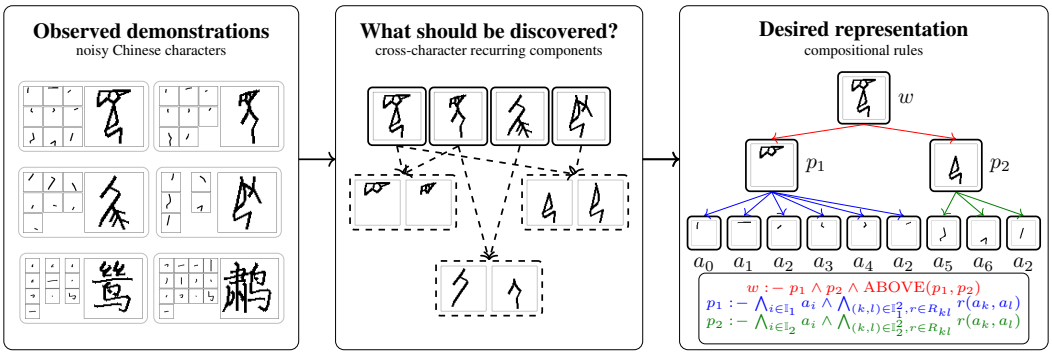

Figure 1: An example of neural-symbolic representation learning from ancient oracle-bone characters. Given noisy stroke traces, the goal is to discover meaningful symbolic components and represent them as grounded primitives, reusable parts, and compositional rules.

Previous work addresses important parts of this problem, but generally does not learn the neural-symbolic interface itself from perceptual demonstrations. Neural representation learning has shown that useful representations can be learned directly from raw data, as in variational autoencoders (Kingma & Welling, 2014); subsequent approaches have learned factors, entities, and relations using disentangled representations (Higgins et al., 2017), capsule networks (Sabour et al., 2017), graph-based representations (Battaglia et al., 2018), and object-centric learning (Locatello et al., 2020; Didolkar et al., 2025). However, these methods generally do not induce an explicit symbolic vocabulary together with rules for composing its elements.

Conversely, inductive logic programming learns symbolic representations from examples, and predicate invention can discover new concepts and rules (Muggleton, 1995; Muggleton et al., 2015; Cropper et al., 2022); such methods, however, typically rely on a predefined symbolic language and therefore do not derive the required symbols and structure directly from perceptual inputs. Neural-symbolic approaches connect neural perception with symbolic representations and reasoning: Neural-Symbolic VQA (Yi et al., 2018) and the neural-symbolic Concept Learner (Mao et al., 2019) support compositional reasoning over visual inputs, while DeepProbLog integrates neural predicates with probabilistic logic programming (Manhaeve et al., 2018). Abductive learning further couples perceptual learning with symbolic reasoning, using inference to identify and correct missing or erroneous symbolic facts (Dai et al., 2019; Dai & Muggleton, 2021). Yet these approaches typically assume that the symbolic vocabulary and relations are already specified: they learn to ground or reason with a given interface, rather than to discover the primitive symbols and their compositional structure jointly.

In this paper, we propose a neural-symbolic representations learning task towards interfacing the two sides through structure abstraction. As illustrated in Fig. 1, we view such a representation as a set, in which each discovered object is defined by a set contains its sub-objects and abstracted relations over them. The set of symbols together with the learned composition rules form a first-order language (FOL) defines the abstract structure (Enderton, 1972). Besides of the symbolic, set-theory-like definition, the element in the set is also accompanied with a neural interpretation, supplying the FOL theory with semantics grounded to the perceptual world (Marker, 2002). Each object has both a neural representation—an embedding grounded in perceptual observations, and a symbolic representation—a term built from primitive symbols and relations.

We design this interface from the perspective of model theory (Marker, 2002): a first-order theory specifies its vocabulary and formation rules, while neural modules learn a set-theoretic model grounded in perceptual observations. We introduce structural abstraction to learn this representation by combining variational Bayesian learning of neural predicates with abductive reasoning over candidate structures and grammar induction. On few-shot oracle-bone character learning and parsing experiments, it substantially outperforms perceptual feature-based motif-discovery baselines; its learned representations also generalise to unseen ancient and modern Chinese characters and discover novel radicals.

The remainder of the paper is organised as follows. Section 2 formalises the NeSy representation-learning problem. Section 3 presents the structural abstraction algorithm. Section 4 evaluates radical discovery on oracle-bone characters, tests generalisation to modern characters and unseen radicals. Section 5 discusses the findings and limitations. Section 6 concludes.

2 NEURAL-SYMBOLIC REPRESENTATION LEARNING

To formalise this problem, we distinguish three levels of objects: atomic observations, inferred parts, and whole observations. Let $\ell \in \{\mathsf{atomic}, \mathsf{part}, \mathsf{whole}\}$ index these levels and let $\mathcal{O}_\ell$ be the corresponding domain of objects. Let R contain the relations between objects provided as background knowledge. According to model theory, the semantics of a first-order theory is given by a model: a unary predicate denotes a subset of an object domain, and a relation denotes a set of tuples over the relevant domains.

In this paper, $\mathcal{S}_\theta$ denotes the semantic interpretation grounded by neural modules with parameters θ. For a learned unary predicate P at level ℓ, its extension $P^{\mathcal{S}_\theta} \subseteq \mathcal{O}_\ell$ is the set of objects to which it applies. The network estimates membership through a score $s_{\theta,P}(o) \in [0,1]$ for an object $o \in \mathcal{O}_\ell$; supplied relations retain their given interpretations. A compositional relation connects component objects to the object they form, which can participate in further relations. We therefore view the NeSy interface as a learned first-order theory together with its semantics: a set-theoretic model of objects, predicates, and relations. Neural modules learn to ground this semantics in perceptual observations, while formation rules specify how grounded objects compose.

Specifically, the input is a set of noisy raw demonstrations $\mathcal{D}_o = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^N$. Here N is the number of demonstrations, y_i is the whole-level observation in demonstration i, and $\boldsymbol{x}_i = \langle x_{i1}, \ldots, x_{iM_i} \rangle$ is its ordered sequence of M_i atomic-level observations. The target is a first-order theory $\mathcal{L} = (\Sigma, \Gamma)$: Σ is its predicate vocabulary and Γ is its set of formation-rule formulae. First, Σ contains neural predicates grounded in the whole, part, and atomic observations. Some relation predicates, such as spatial relations, are supplied as background knowledge.

Secondly, Γ stands for the clauses to be learned, describing how objects at these levels compose. Its formulae are learned from the demonstrations, subject to the meta-rules $\mathcal{M}$ provided as background knowledge. Reusable part-level observations are not provided. For each demonstration, $\mathcal{U}_i$ denotes the set of inferred part occurrences. Each occurrence $u = (s,t) \in \mathcal{U}_i$ is indexed by its start and end positions in $\boldsymbol{x}_i$ and grounded in the contiguous observation sequence $\boldsymbol{x}_i[s:t] = \langle x_{is}, \ldots, x_{it} \rangle$. Learning the theory therefore also requires inferring which observed segments constitute reusable parts.

Example. Consider the oracle-bone character example above. Here, y_i denotes the image of the whole character , and $\boldsymbol{x}_i$ denotes its atomic observations $\langle \, ' \, , — , \ldots \rangle$, one for each stroke. The target is to learn radicals such as . The learner must infer the part-level predicate grounded in the strokes forming the radical; this contiguous stroke segment is the observation associated with an inferred part occurrence $u \in \mathcal{U}_i$. The learner must also infer formulae in Γ that describe how the part is formed from atomic-level neural predicates and their relations.

We now state the learning problem mathematically. Let $\Sigma^\ell_{\text{neural}}$ be the learned unary predicates at level ℓ, and let $\Sigma_{\text{neural}} = \bigcup_{\ell \in \{\mathsf{atomic},\mathsf{part},\mathsf{whole}\}} \Sigma^\ell_{\text{neural}}$. In demonstration i, let w_i be the predicate assigned to y_i, a_{ij} the predicate assigned to x_{ij}, and p_{iu} the predicate assigned to the observed segment of inferred part u. Write X^w_i, X^a_{ij}, and X^p_{iu} for variables that bind to the corresponding whole, atomic, and part objects. Let $\boldsymbol{a}_i$ be the formula expressing the atomic predicates and their relations, and let $\{p_{iu}(X^p_{iu})\}_{u \in \mathcal{U}_i}$ denote the set of inferred part predicates. The symbolic formulae for demonstration i are $\Phi_i = \{w_i(X^w_i), \boldsymbol{a}_i, \{p_{iu}(X^p_{iu})\}_{u \in \mathcal{U}_i}\}$. The assignment η_i maps each variable in Φ_i to its observed object.

We learn the theory $\mathcal{L}$, jointly training the neural grounding parameters θ, inducing a compositional grammar Γ, and discovering the latent parts $\{\mathcal{U}_i\}$. Let $h(\Sigma)$ denote the current hypothesis about predicate assignments, and $p_\theta(o \mid h(\Sigma))$ the likelihood of an observation o under that hypothesis. The hypothesis space $\mathcal{H}$ contains theories whose rule formulae can be generated from $\mathcal{M}$. We write $\models$ for satisfaction of formulae in the semantic model $\mathcal{S}_\theta$. The objective rewards the likelihood of

whole, atomic, and inferred part observations while penalising theory size:

$$\max_{\theta,\{\mathcal{U}_i\}_{i=1}^{N},\mathcal{L}} \quad \sum_{i=1}^{N}\Big[\log p_\theta(y_i \mid h(\Sigma)) + \sum_{j=1}^{M_i}\log p_\theta(x_{ij} \mid h(\Sigma)) + \sum_{u=(s,t)\in\mathcal{U}_i}\log p_\theta(\boldsymbol{x}_i[s:t] \mid h(\Sigma))\Big] - \log\big(|\Sigma_{\text{neural}}| + |\Gamma|\big) \tag{1a}$$

$$\text{s.t.} \quad \mathcal{L} = (\Sigma,\Gamma) \in \mathcal{H}, \qquad \mathcal{H} = \{(\Sigma,\Gamma) \mid \mathcal{M} \vdash \Gamma\}; \tag{1b}$$

$$\mathcal{U}_i \subseteq \{(s,t) \mid 1 \le s \le t \le M_i\} \quad (i = 1,\dots,N), \qquad \bigcup_{i=1}^{N}\mathcal{U}_i \neq \emptyset; \tag{1c}$$

$$\mathcal{S}_\theta \models \Gamma, \qquad \mathcal{S}_\theta, \eta_i \models \Phi_i \quad (i = 1,\dots,N). \tag{1d}$$

The segment indices in $\mathcal{U}_i$ keep occurrences at different positions distinct even when their observed content is identical. The complexity penalty counts learned neural predicates and formation rules; supplied relations are excluded.

The constraints specify the setting requirements. The first defines the hypothesis space: $\mathcal{M}$ restricts the forms of Γ, while the predicate vocabulary is learned from demonstrations. The second requires every inferred part to be grounded in an observed segment and excludes a solution with no inferred parts. The third requires the semantic interpretation to satisfy the learned rules and ground each demonstration's symbolic formulae in its observations. When different observations receive the same neural predicate, they share a symbolic identity; Γ specifies how these identities compose. The learning problem is therefore to discover grounded symbols, latent reusable parts, and compositional formulae together. Structural abstraction, described in Section 3, is one algorithm for solving this problem.

3 The Structural Abstraction Algorithm

Fig 2 illustrates the idea of our proposed Structural Abstraction algorithm, which seeks the theory $\mathcal{L}$, neural grounding $\mathcal{S}_\theta$, and latent part occurrences $\{\mathcal{U}_i\}$ defined in Section 2.

It first estimates which observations belong to the same predicate extension. It then finds recurring segments that can become part-level objects and induces formation rules for their composition. These rules revise the perceptual assignments, so the semantic grounding and the theory are learned together. This interaction follows abductive learning, which revises perceptual predictions in light of symbolic explanations (Dai et al., 2019; Dai & Muggleton, 2021). At level ℓ, the assignment function $f_u^\ell(o)$ selects a neural predicate for observation o; together the three functions form f_u. The relation function $f_b(x_{ik}, x_{il})$ returns the supplied relations between atomic observations, and the abstraction program $f(d_i) = (w_i(X_i^w), \boldsymbol{a}_i)$ constructs the whole and atomic formulae from one raw demonstration $d_i = (y_i, \boldsymbol{x}_i)$.

3.1 Primitive Symbol Discovery

The first task is to assign predicate identities to observations without symbol labels. At any level $\ell \in \{\mathsf{atomic}, \mathsf{part}, \mathsf{whole}\}$, let x be an observation, z its continuous embedding, and $\mathrm{c} \in \{0,1\}^{K_\ell}$ a one-hot assignment to one of the $K_\ell = |\Sigma_{\text{neural}}^\ell|$ candidate neural predicates. The encoder $q_\theta(\mathrm{z} \mid \mathrm{x})$ estimates the embedding, the decoder $p_\theta(\mathrm{x} \mid \mathrm{z})$ reconstructs the observation, and $p_\theta(\mathrm{z} \mid \mathrm{c})$ describes the embedding distribution of the assigned predicate. The current symbolic hypothesis supplies the prior $p(\mathrm{c} \mid h(\Sigma))$. Together these terms give the generative model

$$p_\theta(\mathrm{x},\mathrm{z},\mathrm{c} \mid h(\Sigma)) = p_\theta(\mathrm{x} \mid \mathrm{z})p_\theta(\mathrm{z} \mid \mathrm{c})p(\mathrm{c} \mid h(\Sigma)). \tag{2}$$

The resulting categorical posterior $q(\mathrm{c} \mid \mathrm{x})$ estimates predicate membership from perceptual evidence and the symbolic prior. Writing KL for Kullback-Leibler divergence, its update is

$$q(\mathrm{c} \mid \mathrm{x}) \propto p(\mathrm{c} \mid h(\Sigma)) \exp\big[-\mathrm{KL}\big(q_\theta(\mathrm{z} \mid \mathrm{x}) \| p_\theta(\mathrm{z} \mid \mathrm{c})\big)\big]. \tag{3}$$

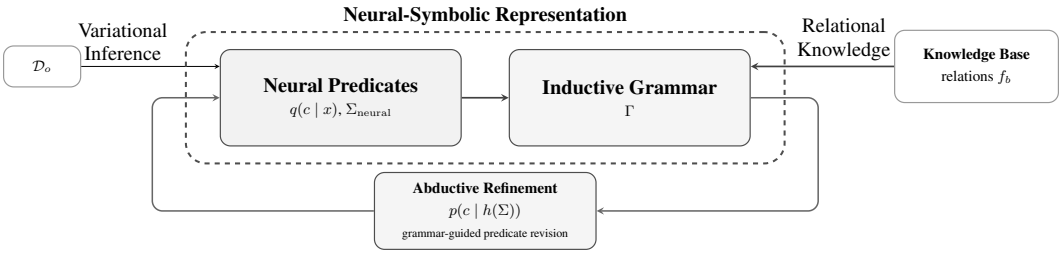

Figure 2: **Neural-symbolic representation learning.** The neural-symbolic representation combines neural predicates with an induced grammar. A variational Bayesian model learns predicate assignments $q(c \mid x)$ from observations, while the predicates and knowledge-base relations induce Γ. The symbolic prior $p(c \mid h(\Sigma))$ is constructed through abduction, which guides predicate refinement and forms an iterative learning loop.

The posterior probabilities provide neural estimates of the predicate membership scores $s_{\theta,P}(\mathrm{x})$ in Section 2. We train the model by maximising the evidence lower bound (ELBO), $\mathcal{J}_{\mathrm{VBM}}(\mathrm{x})$, using variational inference (Corduneanu & Bishop, 2001):

$$\begin{aligned}\mathcal{J}_{\mathrm{VBM}}(\mathrm{x}) = \mathbb{E}_{q_\theta(\mathrm{z}|\mathrm{x})}[\log p_\theta(\mathrm{x} \mid \mathrm{z})] \\ + \log \sum_{\mathrm{c}} p(\mathrm{c} \mid h(\Sigma)) \exp\big[-\mathrm{KL}\big(q_\theta(\mathrm{z} \mid \mathrm{x}) \| p_\theta(\mathrm{z} \mid \mathrm{c})\big)\big] .\end{aligned} \tag{4}$$

The expectation measures reconstruction fit; the divergence compares an observation's embedding with each predicate distribution. The model begins with more candidate predicates than it needs and merges distributions whose divergence falls below the level-specific threshold $\tau^{\ell}_{\mathrm{abs}}$. Initially, this decision is perceptual: a dot and a right-falling stroke can be assigned the same predicate when their appearances overlap. Formation rules learned later can revise such assignments through the prior. The distributional assumptions and ELBO derivation are given in Appendix A.

3.2 Meaningful Subsequence Discovery

The next task is to identify part-level objects within long, noisy atomic sequences. For demonstration i, let $a_{im} = f^a_u(x_{im})$ be the predicate assigned to atomic observation x_{im}, and let X^a_{im} be its object variable. The supplied relation program gives $R^i_{kl} = f_b(x_{ik}, x_{il}) \subseteq \mathcal{R}$ for atomic positions k and l; each $r \in R^i_{kl}$ is a relation predicate. The atomic formula returned by the abstraction program is

$$\boldsymbol{a}_i = \bigwedge_{m=1}^{M_i} a_{im}(X^a_{im}) \ \wedge \bigwedge_{\substack{1 \le k < l \le M_i \\ r \in R^i_{kl}}} r(X^a_{ik}, X^a_{il}). \tag{5}$$

A candidate part is an interval $\boldsymbol{x}_i[s : t]$ of the observed sequence. This restricts a sequence of length M_i to at most $M_i(M_i+1)/2$ candidate windows, compared with $2^{M_i} - 1$ arbitrary subsets. The search first compares their predicate assignments, then their relations; an alignment may permute positions within a window. It retains maximal recurring windows as the occurrences in $\mathcal{U}_i$. Appendix C describes the search.

Without loss of generality, consider two windows of equal length m, with position sets I and J, and let $\phi : I \to J$ be a bijection aligning them. Write $\boldsymbol{a}_i[I]$ for the subformula induced by the first window and $\boldsymbol{a}_j[J]$ for the second. The indicator $\mathbb{I}(A)$ is one if A is true and zero otherwise. For relation sets A, B, define $\mathrm{RelSim}(A, B) = 2|A \cap B|/(|A| + |B|)$ when either set is non-empty, and 1 when both are empty. For $m > 1$, structural consistency is

$$\begin{aligned}\mathrm{SC}(\boldsymbol{a}_i[I], \boldsymbol{a}_j[J], \phi) = \frac{1}{2m} \sum_{k \in I} \Bigg[\mathbb{I}(a_{ik} = a_{j\phi(k)}) \\ + \frac{1}{m-1} \sum_{\substack{l \in I \\ l \neq k}} \mathrm{RelSim}(R^i_{kl}, R^j_{\phi(k)\phi(l)}) \Bigg].\end{aligned} \tag{6}$$

For a one-element window, the score is its predicate agreement. Windows with scores above the structural threshold τ_{sc} are accepted as common substructures. Because initial predicate assignments may be wrong, later rule-based revisions can change which windows are accepted.

3.3 Grammar Induction

Each accepted window $u = (s, t) \in \mathcal{U}_i$ supplies a part-level observation $\boldsymbol{x}_i[s : t]$. The definitive clauses of their predicates $p_{iu} = f_u^p(\boldsymbol{x}_i[s : t])$ are learned from u, and the embeddings of the abstracted parts are computed from the covered examples, serving as the neural-side representation of p_{iu}. Applying the structural consistency defined in equation 6 across different hierarchical levels yields hierarchical structural consistency, which groups demonstrations whose whole, part, and atomic formulae have compatible predicate assignments and relations. A meta-interpretive learning (MIL) algorithm is used for seeking formulae in the hypothesis space derived from $\mathcal{M}$ to explain the shared compositions of the sub-sequences. Whole-level rules connect whole predicates to part or atomic predicates; part-level rules connect part predicates to atomic predicates and the relations among their objects as their definitions. These learned formulae form Γ, while the learned neural predicates extend Σ as its model grounded in raw data. In order to make the induction feasible, a set of general-purposed meta-rules are used for restricting the hypothesis space, which is further introduced in the appendix (Muggleton et al., 2015).

3.4 Representation Refinement by Abductive Learning

The induced rules test whether predicate assignments across demonstrations can share a compositional explanation, as required by the model constraint in equation 1d. When an assignment conflicts with an otherwise shared rule, abductive reasoning proposes a consistent assignment and updates the symbolic prior $p(\mathrm{c} \mid h(\Sigma))$. The neural modules are then trained again under the ELBO in equation 4. Their revised membership estimates change both the atomic formulae and the candidate part windows. Repeating the four stages lets structural evidence improve perceptual clustering and lets improved clustering reveal further structure. The process stops when the assignments and learned theory stabilise. Let g denote the grounding map from a symbolic part occurrence back to its observed segment $\boldsymbol{x}_i[s : t]$.

Importantly, the structural consistency threshold does not require the predicate assignments at the whole, part, and atomic levels to be identical across all demonstrations in a group. Instead, abduction resolves predicate inconsistencies across hierarchical levels by inferring the predicate assignments that best support the shared compositional rules. This process effectively provides pseudo-labels for the observations at each level, represented as distributions $p(\mathrm{c} \mid h(\Sigma))$, which are incorporated into the training of the neural predicates. As a result, the demonstrations within a group are progressively aligned in their neural-symbolic representations, while retaining flexibility in their initial predicate assignments.This is precisely why structural abstraction enables handwritten oracle bone characters with substantial visual variance to share the same semantic meaning.

Algorithm 1 Structural Abstraction

Require: Raw demonstrations $\mathcal{D}_o$, merge thresholds τ_{abs}^{ℓ}, and structural threshold τ_{sc}
  Initialise whole- and atomic-level predicate distributions
  repeat
    Assign predicates to observations using the current neural grounding $\mathcal{S}_\theta$
    Infer maximal recurring windows and their part occurrences $\{\mathcal{U}_i\}$
    Learn part predicates and induce formation rules Γ
    Revise inconsistent assignments and update neural modules
  until assignments and $\mathcal{L} = (\Sigma, \Gamma)$ stabilise
Ensure: Theory $\mathcal{L}$, semantic grounding $\mathcal{S}_\theta$, and part grounding map g

4 Experiments

The experiments examine whether structural abstraction learns grounded components that recur across noisy demonstrations, whether the resulting structure extends to unseen characters, and

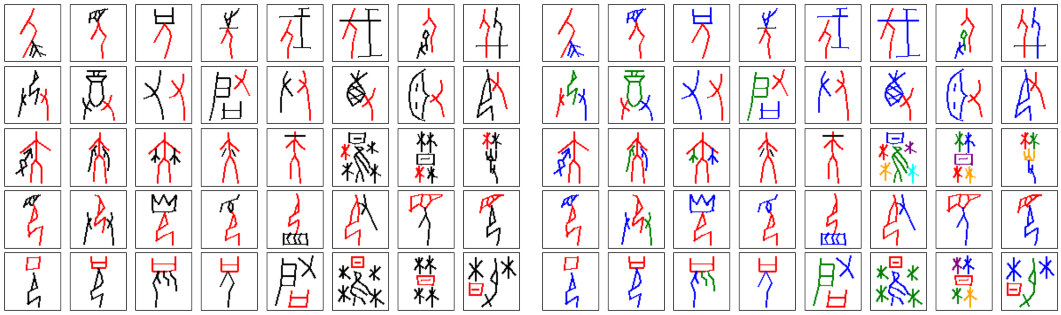

Figure 3: Oracle-bone radical discovery. Radicals discovered by our method are shown on the left, while the corresponding expert annotations are shown on the right.

whether it helps identify components absent from the training set. We begin with handwritten oracle-bone characters, the motivating case, and then examine modern Chinese characters and novel radicals. Digit sequences provide a controlled setting in which the generating grammar is known.

4.1 Experimental Setting

We evaluate part discovery by Substructure Discovery Recall (SDR): for each annotated substructure category, we measure the proportion of its instances recovered among the candidate parts and then average these recalls across categories. Expert radical annotations are used only for this evaluation; they are not supplied during predicate learning, part discovery, or grammar induction. Consequently, SDR measures recovery of the annotated categories, not the quality of every candidate part produced by an unsupervised method.

We compare structural abstraction with LoCoMotif, an unsupervised method for finding recurring motifs in multivariate time series (Wesenbeeck et al., 2024), applied to three learned feature spaces. An autoencoder learns features through reconstruction (Berahmand et al., 2024), SimCLR through contrastive learning (Chen et al., 2020), and the variational Bayesian model (VBM) provides the perceptual features used by our method. For evaluation, motif groups that LoCoMotif separates but that correspond to the same annotated substructure are merged using ground-truth information. Structural abstraction does not use this merging step. The VBM comparison tests what is recovered from the learned perceptual features when recurring parts are sought by motif discovery rather than by the structural reasoning procedure.

4.2 Radical Discovery in Oracle-Bone Characters

The oracle-bone data contain handwritten characters with sequentially recorded strokes. There are 80 character categories, with 12 training and four test samples per category, and eight expert-annotated radical categories. The small number of demonstrations, variation in handwriting, and imbalance among strokes and radicals make recurring parts difficult to identify from visual similarity alone. The learner receives the complete character and its stroke sequence, but no radical boundaries or labels.

Figure 3 places discovered radical instances beside the corresponding expert annotations. The comparison shows that the inferred parts can be grounded in strokes of individual characters and can recur across different samples; it also makes visible where the discovered parts differ from the annotations.

Table 1 reports macro-averaged recall. Structural abstraction recovers 72.28% of the annotated oracle-bone radical instances, compared with 9.75%, 9.58%, and 13.09% for the autoencoder, SimCLR, and VBM feature spaces paired with LoCoMotif. The comparison with VBM is particularly informative: perceptual features from the same model give much lower recovery when recurring parts are selected by feature-space motifs. This result is consistent with the role of predicate assignments and structural relations in identifying reusable components, although the comparison does not isolate the effect of each algorithmic step.

Table 1: Macro-averaged substructure discovery recall across datasets.

Dataset	AE+LoCoMotif	SimCLR+LoCoMotif	VBM+LoCoMotif	Ours
Oracle-bone characters	9.75%	9.58%	13.09%	72.28%
Modern Chinese characters	54.24%	54.58%	57.20%	91.54%
Controlled digit sequences	27.41%	17.71%	29.89%	94.02%

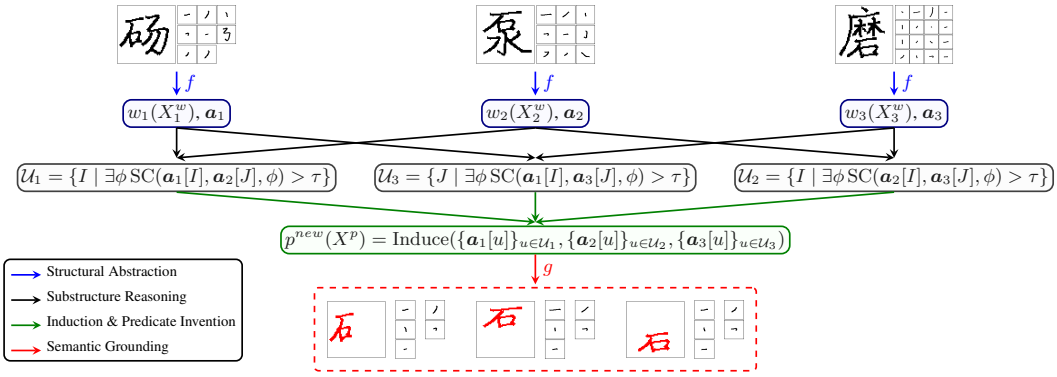

Figure 4: Discovery of a new radical in unseen modern Chinese characters. The diagram shows atomic-level descriptions, shared structure, induction of a part predicate and rule, and grounding in observed strokes.

4.3 Structural Discovery in Modern Chinese Characters

We next examine whether the same procedure recovers parts beyond the oracle-bone setting. The modern-character data comprise 1,000 one-shot training samples and 130 test samples, all of which are out of distribution relative to training; the evaluation covers 13 radical categories with imbalanced radical and stroke frequencies. The learner again receives characters and their atomic observations without radical annotation.

As shown in Table 1, structural abstraction reaches 91.54% macro-averaged SDR, compared with 54.24%–57.20% for the three feature-based baselines. Thus, the recovery of annotated parts is not confined to the handwritten oracle-bone examples. This result concerns parts within the annotated categories of the modern-character test set; discovery of radicals absent from training is examined separately below.

4.4 Discovery of Novel Radicals in Unseen Characters

A learned vocabulary must also be useful when an observation contains structure it cannot yet describe. We therefore apply the abstraction function learned from the one-shot modern-character training set to an additional set of 100 distinct characters, comprising ten samples from each of ten radical categories. Neither these characters nor their radicals occur in that training set. Existing atomic predicates and relations provide a partial description of each character; substructure reasoning then searches for recurring parts across the new observations, and grammar induction assigns a predicate to the shared structure. Figure 4 illustrates this sequence from observations to a grounded new part.

The radical-wise SDRs in Table 2 range from 40% to 100%: four categories are recovered completely, whereas two reach 40%. The variation shows that discovering a new radical does not mean every instance is recovered. Taken together with the illustration, these results support the more specific claim that structure learned from previous characters can assist the discovery and grounding of parts outside the training vocabulary.

Table 2: Substructure discovery recall for radical categories absent from the modern-character training set.

Radical	土	山	日	水	火/灬	王	石	艹	纟	言
SDR	100%	90%	60%	60%	40%	100%	100%	100%	40%	80%

4.5 Controlled Verification with Digit Sequences

Finally, we use digit sequences constructed from MNIST (LeCun et al., 1998) to check structure recovery against a known generating grammar. Each sequence has length one to ten and is formed by selecting and ordering the subsequences 12, 345, and 6789, with zeros inserted at random positions. The data contain 11,516 training and 1,543 test sequences.

Structural abstraction obtains 94.02% macro-averaged SDR, compared with 17.71%–29.89% for the feature-based baselines in Table 1. On the test sequences, neural-symbolic parsing fully explains 84.64%, partially explains 13.67%, and leaves 1.69% unparsed. After matching atomic symbols to their ground-truth digit labels, all returned parsing programs are consistent with the annotated sequence structure. A parsing example and a comparison of the learned and generating grammars are shown in Appendix Figures 7 and 6. These measurements provide a controlled check that the induced symbols and rules describe the intended recurring subsequences.

5 Discussion and Limitations

The experiments show that a neural-symbolic interface can be learned from perceptual demonstrations even when its primitive predicate vocabulary and formation rules are not specified in advance. From the aspect of model theory, this representation serves as the interface and have two linked parts. Neural modules ground predicates in observed objects, while the learned first-order theory defines the compositions explicitly. Their co-existence matters because perceptual similarity alone cannot determine which noisy observations should share a symbol, and shared structure can revise those assignments.

Oracle-bone characters provide a concrete test of this setting. On the few-shot handwritten data, structural abstraction recovers substantially more expert-annotated radical instances than the feature-based motif baselines. It also finds radicals in modern characters and identifies substructures in unseen characters, including radicals absent from the training set. The controlled digit sequences confirm that the method can recover recurring structure when the generating grammar is known. These results establish the usefulness of the learned representation for structural discovery and parsing; they do not by themselves establish the historical or linguistic correctness of every discovered radical. Expert annotations are used for evaluation, rather than to supervise discovery.

Several assumptions bound the setting studied here. The set-theory-like whole, part, and atomic levels; the meta-rules; and a handful of spacial relation predicates are supplied. Part discovery searches for contiguous sub-sequences in the noisy demonstrations, although their elements may be aligned in different orders. Merge and structural thresholds control the size of the learned theory, and neural grounding remains limited by the coverage of the observations and the chosen models. These choices reduce the search space, but they also restrict which structures can be found. The reported radical recall depends on the available expert annotations, which do not exhaust all potentially meaningful structures.

6 Conclusion

In this paper, we formulated neural-symbolic representation learning from noisy perceptual demonstrations as learning a first-order theory and its model that grounds its set-theoretic semantics in observed objects using neural nets. The algorithm of structural abstraction is proposed, coupling neural predicate discovery, recurring part discovery, grammar induction, and abductive refinement of the learned representation. On handwritten oracle-bone characters, it recovers expert-annotated radicals more effectively than the tested feature-based baselines; experiments on modern characters, unseen radicals, and controlled digit sequences further demonstrate the representation supports reusable compositional structure and extrapolates to unseen data with unknown structures. The setting tries to establish the *neural-symbolic interfacing* as a new learning problem, in which its primitive vocabulary and formation rules are the learning target of a open first-order language, whose semantics should be grounded by the learned neural nets in this joint neural-symbolic learning task.

REFERENCES

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

Kamal Berahmand, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li, and Yue Xu. Autoencoders and their applications in machine learning: a survey. *Artificial Intelligence Review*, 57(2): 28, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1597–1607, 2020.

Adrian Corduneanu and Christopher M. Bishop. Variational bayesian model selection for mixture distributions. In *Proceedings of the Eighth International Conference on Artificial Intelligence and Statistics*, pp. 27–34, 2001.

Andrew Cropper, Sebastijan Dumancic, Richard Evans, and Stephen H. Muggleton. Inductive logic programming at 30. *Machine Learning*, 111(1):147–172, 2022.

Wang-Zhou Dai and Stephen H. Muggleton. Abductive knowledge induction from raw data. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 1845–1851, 2021.

Wang-Zhou Dai, Qiu-Ling Xu, Yang Yu, and Zhi-Hua Zhou. Bridging machine learning and logical reasoning by abductive learning. In *Advances in Neural Information Processing Systems 32*, pp. 2811–2822, 2019.

Luc De Raedt, Fredrik Heintz, Kristian Kersting, and Giuseppe Marra. A learning and reasoning manifesto. In *Challenges and Algorithms for Knowledge Discovery from Data*, volume 16067, pp. 97–108, 2025.

Aniket Rajiv Didolkar, Andrii Zadaianchuk, Anirudh Goyal, Michael Curtis Mozer, Yoshua Bengio, Georg Martius, and Maximilian Seitzer. On the transfer of object-centric representation learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Herbert B. Enderton. *A mathematical introduction to logic*. Academic Press, 1972.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems 33*, 2020.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems 31*, pp. 3753–3763, 2018.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*, 2019.

David Marker. *Model theory: an introduction*. Springer, 2002.

Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

Stephen H. Muggleton. Inverse entailment and progol. *New generation computing*, 13(3–4):245–286, 1995.

Stephen H. Muggleton, Dianhuan Lin, and Alireza Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited. *Machine Learning*, 100(1):49–73, 2015.

Lukas Muttenthaler, Klaus Greff, Frieda Born, Bernhard Spitzer, Simon Kornblith, Michael C. Mozer, Klaus-Robert Müller, Thomas Unterthiner, and Andrew K. Lampinen. Aligning machine and human visual representations across abstraction levels. *Nature*, 647(8089):349–355, 2025.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30*, pp. 3856–3866, 2017.

Zilong Wang, Jiongda Wang, Xiaoyong Chen, Meng Wang, Ming Ma, Zhipeng Wang, Zhenyu Zhou, Tianming Yang, and Wang-Zhou Dai. From end-to-end to step-by-step: Learning to abstract via abductive reinforcement learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 6515–6523, 2025.

Daan Van Wesenbeeck, Aras Yurtman, Wannes Meert, and Hendrik Blockeel. Locomotif: discovering time-warped motifs in time series. *Data Mining and Knowledge Discovery*, 38(4):2276–2305, 2024.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems 31*, pp. 1039–1050, 2018.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and polishing of the manuscript.

We have not used generative AI tools for research ideation, data collection, data analysis, experimental design, code generation, or the generation of figures, tables, or other research artifacts, and the rest of the required disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools for language editing and polishing as a recommended disclosure. We have reviewed all AI-assisted work and manually verified the revised text to ensure that the intended meaning, technical content, and claims were accurately preserved. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise any specific ethical concerns.

REPRODUCIBILITY STATEMENT

We strive to ensure the reproducibility of our work. Detailed descriptions of the proposed method and experimental settings are provided in the main paper and supplementary materials. Upon publication, we will release the source code and datasets used in this work to facilitate reproduction and further research.

A DERIVATION OF THE ELBO

For the observation model in equation 2, let $q_\theta(\mathrm{z} \mid \mathrm{x})$ be the embedding posterior and $q(\mathrm{c} \mid \mathrm{x})$ the categorical posterior. Jensen's inequality gives

$$
\begin{aligned}
\log p(\mathrm{x} \mid h) &= \log \int \sum_c p(\mathrm{x}, \mathrm{z}, \mathrm{c} \mid h)\, \mathrm{dz} \\
&= \log \int \sum_c q(\mathrm{c} \mid \mathrm{x}) q(\mathrm{z} \mid \mathrm{x}) \frac{p(\mathrm{x}, \mathrm{z}, \mathrm{c} \mid h)}{q(\mathrm{c} \mid \mathrm{x}) q(\mathrm{z} \mid \mathrm{x})}\, \mathrm{dz} \\
&= \log \int \sum_c q(\mathrm{c} \mid \mathrm{x}) q(\mathrm{z} \mid \mathrm{x}) \frac{p(\mathrm{x} \mid \mathrm{z}) p(\mathrm{z} \mid \mathrm{c}) p(\mathrm{c} \mid h)}{q(\mathrm{c} \mid \mathrm{x}) q(\mathrm{z} \mid \mathrm{x})}\, \mathrm{dz} \\
&\geq \int q(\mathrm{z} \mid \mathrm{x}) \log p(\mathrm{x} \mid \mathrm{z})\, \mathrm{dz} + \int \sum_c q(\mathrm{c} \mid \mathrm{x}) q(\mathrm{z} \mid \mathrm{x}) \log \frac{p(\mathrm{z} \mid \mathrm{c}) p(\mathrm{c} \mid h)}{q(\mathrm{c} \mid \mathrm{x}) q(\mathrm{z} \mid \mathrm{x})}\, \mathrm{dz} \\
&= \int q(\mathrm{z} \mid \mathrm{x}) \log p(\mathrm{x} \mid \mathrm{z})\, \mathrm{dz} + \sum_c q(\mathrm{c} \mid \mathrm{x}) \Big(\int q(\mathrm{z} \mid \mathrm{x}) \log \frac{p(\mathrm{z} \mid \mathrm{c})}{q(\mathrm{z} \mid \mathrm{x})}\, \mathrm{dz} + \log \frac{p(\mathrm{c} \mid h)}{q(\mathrm{c} \mid \mathrm{x})} \Big) \\
&= \mathcal{J}_{\mathrm{VBM}}(\mathrm{x}).
\end{aligned}
$$

Solving the variational optimization problem yields

$$
q(\mathrm{c} \mid \mathrm{x}) \propto p(\mathrm{c} \mid h) \exp(-\mathrm{KL}(q(\mathrm{z} \mid \mathrm{x}) \parallel p(\mathrm{z} \mid \mathrm{c}))).
$$

Thus, the variational lower bound is given by

$$
\mathcal{J}_{\mathrm{VBM}}(\mathrm{x}) = \mathbb{E}_{q(\mathrm{z}|\mathrm{x})} \left[\log p(\mathrm{x} \mid \mathrm{z}) \right] + \log \sum_{\mathrm{c}} p(\mathrm{c} \mid h(\Sigma)) \exp\big[-\mathrm{KL}\big(q_\theta(\mathrm{z} \mid \mathrm{x}) \| p_\theta(\mathrm{z} \mid \mathrm{c}) \big) \big]
$$

Maximising the bound over $q(\mathrm{c} \mid \mathrm{x})$ gives the categorical update in equation 3. Substituting that update yields $\mathcal{J}_{\mathrm{VBM}}(\mathrm{x})$ in equation 4. As a side note, to obtain an analytical expression for the evidence lower bound, we assume that all continuous random variables follow multivariate Gaussian distributions with isotropic covariance matrices and independent dimensions.

B META-RULES

The meta-rules used to restrict rule induction are formulated below using BNF-style notation:

$$
\begin{aligned}
\langle \text{start} \rangle &::= \langle \text{whole-symbol} \rangle \\
\langle \text{whole-symbol} \rangle &::= \langle \text{atomic-compound} \rangle \mid \langle \text{part-compound} \rangle \\
\langle \text{part-symbol} \rangle &::= \langle \text{atomic-compound} \rangle \\
\langle \text{relation} \rangle &::= \langle \text{any-relation-symbol} \rangle \\
\langle \text{part-compound} \rangle &::= \langle \text{part-formula} \rangle \mid \langle \text{part-compound} \rangle \ \wedge \ \langle \text{part-formula} \rangle \\
\langle \text{atomic-compound} \rangle &::= \langle \text{atomic-formula} \rangle \mid \langle \text{atomic-compound} \rangle \ \wedge \ \langle \text{atomic-formula} \rangle \\
\langle \text{part-formula} \rangle &::= \langle \text{relation} \rangle \text{ `(' } \langle \text{part-symbol} \rangle \text{ `,' } \langle \text{part-symbol} \rangle \text{ `)'} \\
\langle \text{atomic-formula} \rangle &::= \langle \text{relation} \rangle \text{ `(' } \langle \text{atomic-symbol} \rangle \text{ `,' } \langle \text{atomic-symbol} \rangle \text{ `)'}.
\end{aligned}
$$

C ALGORITHM FOR SUBSTRUCTURE REASONING

Substructure reasoning essentially concerns the isomorphism between two formulas. In practice, during substructure reasoning, selecting only the largest subformula is sufficient for learning a good grammar. To enable efficient reasoning, we make the assumption that the shared substructure between two atomic-level observation sequences corresponds to a contiguous subsequence of observations, while the order within the extracted subsequence can be permuted. Based on this assumption, the reasoning process is implemented as a meta-program. Substructure reasoning first identifies candidate pairs of subformulas with consistent unary predicates, and then evaluates the consistency of their binary predicates. When all predicates satisfy the consistency criterion, the common subformula is inferred from the two formulas.

D EXPERIMENTAL SETTING

D.1 CHINESE CHARACTER

The neural network architectures used to parameterize the probability distributions in the Chinese Character experiments are presented in Table 3.

For oracle bone inscriptions, 315 whole symbols and 200 atomic symbols are initialized. The cluster merging thresholds are set to 2, 3, and 4 for the whole-, part-, and atomic-level neural modules, respectively. The threshold for hierarchical structural consistency is set to 0.9. The background knowledge contains spatial relations, such as positional relations.

For modern Chinese characters, 200 atomic symbols are initialized. The cluster merging thresholds are set to 3 and 1.5 for the part- and atomic-level neural modules, respectively. The background knowledge contains spatial relations.

D.2 DIGIT SEQUENCES

The neural network architectures used to parameterize the probability distributions in the digit sequence experiments are presented in Table 4 and Table 5. Table 4 presents the architecture of the atomic neural modules, while Table 5 presents the architectures of the part and whole modules.

The numbers of initialized whole symbols and atomic symbols are 96 and 24, respectively. The cluster merge thresholds are set to 0.75, 10, and 3 for the whole-, part-, and atomic-level neural modules, respectively. The threshold for hierarchical structural consistency is set to 0.9. The background knowledge contains only the ordering relation.

Table 3: Encoder and decoder architectures used in the Chinese character experiments.

Network	Layer	Parameters
	Input	64×64
Encoder	Conv2d + InstanceNorm2d + ReLU	$c = 32$, $k = 4$, $s = 2$, $p = 1$
Encoder	Conv2d + InstanceNorm2d + ReLU	$c = 64$, $k = 4$, $s = 2$, $p = 1$
Encoder	Conv2d + InstanceNorm2d + ReLU	$c = 128$, $k = 4$, $s = 2$, $p = 1$
Encoder	Conv2d + InstanceNorm2d + ReLU	$c = 256$, $k = 4$, $s = 2$, $p = 1$
Encoder	Flatten + Linear	$256 \times 4 \times 4 \to 16$
Decoder	Linear + Unflatten + ReLU	$16 \to 256 \times 4 \times 4$
Decoder	Upsample	$s = 2$
Decoder	Conv2d + InstanceNorm2d + ReLU	$c = 128$, $k = 3$, $p = 1$
Decoder	Upsample	$s = 2$
Decoder	Conv2d + InstanceNorm2d + ReLU	$c = 64$, $k = 3$, $p = 1$
Decoder	Upsample	$s = 2$
Decoder	Conv2d + InstanceNorm2d + ReLU	$c = 32$, $k = 3$, $p = 1$
Decoder	Upsample	$s = 2$
Decoder	Conv2d	$c = 1$, $k = 3$, $p = 1$
	Output	64×64

Table 4: Encoder and decoder architectures for the atomic module used in the digit sequence experiments.

	Layer	Parameters
	Input	28×28
Encoder	Conv2d + ReLU	$c = 32$, $k = 7$, $s = 4$
Encoder	Conv2d + ReLU	$c = 64$, $k = 3$, $s = 2$
Encoder	Conv2d + ReLU	$c = 128$, $k = 3$, $s = 2$
Encoder	Flatten + Linear	$128 \times 4 \times 4 \to 8$
Decoder	Linear + Unflatten + ReLU	$8 \to 128 \times 4 \times 4$
Decoder	Upsample	$s = 2$
Decoder	Conv2d + InstanceNorm2d + ReLU	$c = 64$, $k = 3$
Decoder	Upsample	$s = 2$
Decoder	Conv2d + InstanceNorm2d + ReLU	$c = 32$, $k = 3$
Decoder	Upsample	$s = 2$
Decoder	Conv2d	$c = 1$, $k = 3$
	Output	28×28

E Additional Experimental Results

E.1 Chinese Character

Figure 5 presents additional qualitative results on oracle-bone and modern Chinese characters. These results demonstrate the effectiveness of the proposed method in discovering substructures (radicals) and its robustness to variations in radical scale, position, and deformation.

E.2 Digit Sequences

The algorithm results are shown in Fig.6. For ease of comparison, whole symbols are omitted from the learned grammar. By selecting appropriate thresholds, the learned formal language is broadly consistent with the ground truth. As the method is unsupervised, it discovers some additional subsequences. However, the number of latent observations corresponding to each part symbol indicates that the method has captured the core patterns in the data.

Table 5: Encoder and decoder architectures for part and whole module used in the digit sequence experiments.

	Layer	Parameters
	Input	280×28
Encoder	Conv2d + ReLU	$c = 32$, $k = 16 \times 8$, $s = 14 \times 4$
Encoder	Conv2d + ReLU	$c = 64$, $k = 5 \times 3$, $s = 3 \times 2$
Encoder	Conv2d + ReLU	$c = 128$, $k = 5 \times 2$, $s = 1$
Encoder	Flatten + Linear	$128 \times 10 \times 4 \rightarrow 8$
Decoder	Linear + Unflatten + ReLU	$8 \rightarrow 128 \times 10 \times 4$
Decoder	Upsample	$s = 4 \times 2$
Decoder	Conv2d + InstanceNorm2d + ReLU	$c = 64$, $k = 4$
Decoder	Upsample	$s = 4 \times 2$
Decoder	Conv2d + InstanceNorm2d + ReLU	$c = 32$, $k = 4$
Decoder	Upsample	$s = 4 \times 2$
Decoder	Conv2d	$c = 1$, $k = 4$
	Output	280×28



Figure 5: Examples of discovered radicals. Red parts indicate radicals discovered from characters. Left: oracle bone inscriptions; right: modern Chinese characters.

F NEURAL-SYMBOLIC PARSING

The most direct way to evaluate the learned neural predicates and formal languages is to measure their effectiveness in explaining and analyzing test samples. Therefore, we propose neural-symbolic parsing, a method for sample analysis comprising syntactic parsing and semantic verification. Figure 7 illustrates how a digit sequence is parsed. Syntactic parsing analyzes the structure of each sample and decomposes it into the constituent substructures. Semantic verification checks whether the logical formulas of the same entity at two different levels are semantically equivalent. For example, if both $p^2(\boldsymbol{x}_1)$ and $a^6a^4(x_4, x_5)$ are true, the logical rule $p^2(X^p) :- a^6(X^a)a^4(X^a)$ is considered to explain the structure of the sample.

Most importantly, neural-symbolic parsing does not simply output a True or False value. Instead, it produces a set of formation rules (i.e., logical program), thereby categorizing samples into three types. The first type consists of in-distribution data, for which the logical program fully explains the sample structure. The second type consists of interpretable out-of-distribution data, for which the logical program partially explains the sample structure. The third type consists of uninterpretable out-of-distribution data, for which no logical program is returned. The significance of neural-symbolic parsing lies in enabling machines to identify what they have learned and what remains unexplained, opening new possibilities for continual learning in neural-symbolic AI.

Neural-symbolic parsing divides the test set into fully parsed, partially parsed, and unparsed samples, accounting for 84.64%, 13.67%, and 1.69%, respectively. For parsed samples, we evaluate whether the generated formation rules are consistent with the ground-truth annotations. Specifically, we match atomic symbols to their ground-truth labels to assign semantics to the corresponding part and whole symbols, and then compare the generated and ground-truth structures. The parsing accuracy is 100%, indicating that neural-symbolic parsing produces correct formation rules for all

$$\begin{aligned}&\mathsf{Seq} \to \mathsf{Elem} \mid \mathsf{Elem}\ \mathsf{Seq},\\ &\mathsf{Elem} \to P \mid A,\\ &P \to 12 \mid 345 \mid 6789,\\ &A \to 0 \mid 1 \mid 2 \mid \cdots \mid 9.\end{aligned}$$

$$\begin{aligned}&\mathsf{Seq} \to \mathsf{Elem} \mid \mathsf{Elem}\ \mathsf{Seq},\\ &\mathsf{Elem} \to p^0 \mid p^1 \mid p^2 \mid p^3 \mid p^4 \mid p^5 \mid p^6 \mid A,\\ &p^0 \to a^1a^8a^3a^9, p^1 \to a^7a^2a^5,\\ &p^2 \to a^6a^4, p^3 \to a^4a^0, p^4 \to a^7a^2,\\ &p^5 \to a^3a^9, p^6 \to a^7a^5,\\ &A \to a^0 \mid a^1 \mid \cdots \mid a^9.\end{aligned}$$

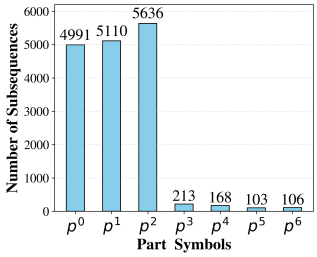

Figure 6: Results on digit sequences. Left: ground-truth grammar; middle: abstracted grammar; right: number of grounded subsequences for part symbols.

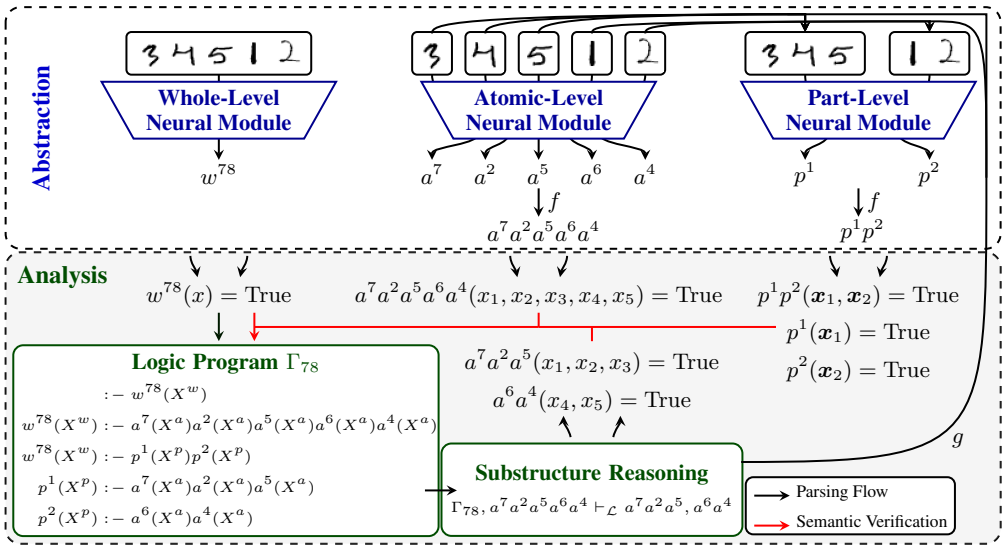

Figure 7: Neural-Symbolic Parsing. In syntactic parsing, the production rules of the logic program Γ_{78} are first efficiently searched using the whole-level symbol w^{78}. All valid subformulas are then recursively derived from atomic-level formulas according to Γ_{78}. In semantic verification, the truth values of the formulas at the three levels are first determined from the abstraction program. The logically equivalent production rules are then identified, and the corresponding valid grammar is output. In this example, the output is Γ_{78}.

successfully parsed test samples. Consequently, we only need to examine why some test samples are partially parsed or unparsed. The main reason is that neural modules at different levels do not generalize to large-scale variations in the samples, leading to semantic verification failure. In short, the generalization capability of the neural modules determines whether a sample can be successfully parsed.