Hierarchical Neuro-Symbolic Knowledge Tracing: From Attention Maps to Transferable Production Rules

Journal Title
XX(X):1–8
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Ankit Anand [1], Upendra Singh Jadon [2], and Bineet Kumar [3]

Abstract

In educational knowledge tracing, deep learning models achieve high predictive accuracy but operate as black boxes, eroding teacher trust. Symbolic tutors are interpretable but require hand-crafted rules that do not scale. We propose a hierarchical neuro-symbolic framework (HNSKT) that extracts human-readable first-order logic rules from a transformer's attention maps using inductive logic programming (ILP). Our key innovations are: (i) a two-stage hierarchical induction that reduces ILP's exponential search space from days to minutes; (ii) a formal treatment of teacher edits as constrained hypothesis revision (after Wrobel 1996) with provably consistency-preserving mutation operators; (iii) a recency control experiment that quantifies attention faithfulness. On ASSISTments and EdNet, HNSKT matches pure neural accuracy (AUC 0.78, 95% CI [0.76,0.80]) while providing rules rated 4.6/5 for interpretability in a pilot teacher study (n=5, Fleiss' $\kappa = 0.72$; these preliminary results require confirmation with a larger sample). Cross-domain transfer (math $\rightarrow$ synthetic physics) improves AUC by 12.3% (95% CI [9.5%,15.1%]) with only 20% of target data, demonstrating the value of reusable symbolic knowledge. A synthetic experiment (60 skills, $k = 6$ clusters) confirms that attention-guided ILP recovers 68% of direct prerequisite edges (precision 0.74, recall 0.68) under a realistic hierarchical stress test. The framework gives students actionable explanations, teachers verifiable insights, and researchers a scalable, formally grounded neuro-symbolic method for education.

Keywords

Neuro-symbolic AI, knowledge tracing, explainable AI, inductive logic programming, transfer learning.

Introduction and Motivation

Why Do We Need Neuro-Symbolic Knowledge Tracing?

Current AI in education suffers from a fundamental trade-off:

- **Pure neural models (e.g., DKT, AKT)** predict student performance accurately but are black boxes. A teacher sees a '70% chance of a wrong answer' – but *why*? A student receives no diagnosis, only a probability. This opacity erodes trust and prevents pedagogical action.
- **Pure symbolic tutors (e.g., ACT-R, cognitive tutors)** are interpretable (rule-based) but require the manual authoring of thousands of production rules (4). They do not generalize across domains and cannot discover new patterns from data.

The gap is acute: Teachers often report that current AI dashboards are not useful for daily instruction because they lack causal, actionable explanations. Students, meanwhile, are left with anxiety: "the AI says I will fail," but have no roadmap to improve.

What Does HNSKT Offer?

Our framework bridges this gap by providing:

- **For students**: personalized, rule-based explanations ("You struggle with fraction division because you have not mastered the inverse relationship with multiplication"), plus fair flagging that avoids spurious correlations (e.g., homework time).
- **For teachers**: interpretable skill maps, automatic discovery of hidden prerequisite dependencies, and the ability to inspect, reject, or edit any rule – restoring agency and trust.
- **For researchers**: a scalable neuro-symbolic pipeline that turns a black-box neural network into a source of verifiable pedagogical knowledge, plus cross-domain transfer that reuses rules as assets.

Key Benefits Summarized

1. **Interpretability without sacrificing accuracy:** It matches the pure neural AUC (0.78) but outputs first-order logic rules.
2. **Scalable ILP:** Hierarchical two-stage induction reduces runtime from $O(k^2)$ to near-linear in the number of skill buckets.
3. **Human-in-the-loop:** Teachers edit rules, which become symbolic priors without retraining.

[1] Independent Researcher,ORCID: 0009-0003-5540-5566
[2] upendrajadonusj@gmail.com, ORCID:0009-0002-4200-9321
[2] Independent Researcher, bineetnit@gmail.com

Corresponding author:
Ankit Anand, Managing Consultant, 1212 Corkwood Dr, Melissa, TX, 75454, USA
Email: manaankit@gmail.com

4. **Transfer learning:** Rules extracted from a source domain improve cold-start performance in a target domain (e.g., math → synthetic physics).
5. **Fairness by design:** Teachers can reject rules based on protected attributes (e.g., gender, zip code).

Preliminaries

Before describing our framework, we define key concepts for readers less familiar with AI terminology. Experts may skip this section.

Knowledge Tracing (KT)

KT models a student's mastery of skills over time based on their interactions with practice problems. Given a sequence of (question, answer) pairs, a KT model predicts whether the student will answer the next question correctly.

Deep Knowledge Tracing (DKT)

DKT uses a recurrent neural network to perform KT. It achieves high accuracy but behaves as a black box, providing only a probability without explanation.

Attention Maps (in Transformers)

Modern KT models (e.g., AKT) use a Transformer with an attention mechanism. The attention map is a matrix of weights $\alpha_{t,i}$ indicating how relevant each previous interaction i is to the current prediction at step t. These weights offer clues about skill dependencies.

Symbolic Rules and First-Order Logic (FOL)

A symbolic rule is an "if-then" statement. FOL formalizes such rules using predicates (e.g., Mastered$(student, skill, time)$) and logical connectives.

Inductive Logic Programming (ILP)

ILP automatically learns FOL rules from examples and background facts. The search space grows exponentially with the number of predicates – a challenge we address via hierarchical clustering.

Neuro-Symbolic AI

Neuro-symbolic AI combines neural networks (learning patterns from data) with symbolic reasoning (logic, explanation). In HNSKT, the neural component (AKT) provides attention maps and skill embeddings; the symbolic component (ILP) extracts rules.

Hierarchical Clustering of Skills

Skills are grouped into conceptual "buckets" (e.g., "Fraction operations") using k-means on neural skill embeddings. This reduces the number of items the ILP engine must consider at once.

Symbolic Prior

A set of rules fed as additional input to a neural model. In HNSKT, teacher-edited rules become a prior vector $\mathbf{p}$ concatenated with the neural hidden state.

Related Work

Deep Knowledge Tracing

DKT (1) and later models like DKVMN (6) and AKT (2) achieve high accuracy but offer no interpretable rules.

Neuro-Symbolic AI in Education

We extend the known approach of injecting symbolic constraints (Responsible-DKT (3)) to the extraction stage. Closest to our transfer mechanism is Hu et al. (11), who distill rules from neural networks into a teacher network for NLP tasks; we adapt their logic distillation to knowledge tracing with a hierarchical ILP prior.

Scalable ILP and Hypothesis Revision

Meta-interpretive learning (5) prunes search via meta-rules. Our teacher-edit mechanism can be viewed as a constrained hypothesis revision operation in ILP (12), where edits define a permissible hypothesis space without retraining.

Proposed HNSKT Framework

Figure 1 illustrates the four stages of HNSKT (see Appendix for full figure).

Stage 1: Attention-Guided Neural KT

We use AKT (2) for its monotonic attention. Given a student interaction sequence $\{(q_1, a_1), \ldots, (q_T, a_T)\}$ with skill tags s_q, AKT produces attention weights $\alpha_{t,i}$.

Stage 2: Hierarchical ILP Rule Induction

Skill clustering: Extract skill embeddings from AKT's input layer and apply k-means ($k = 20$) to form buckets $\mathcal{B}_1, \ldots, \mathcal{B}_{20}$.

High-level induction: For each bucket pair $(\mathcal{B}_a, \mathcal{B}_b)$, define the average attention weight

$$\bar{\alpha}_{ab} = \frac{1}{|\mathcal{B}_a| \cdot |\mathcal{B}_b| \cdot N_{ab}} \sum_{s_a \in \mathcal{B}_a} \sum_{s_b \in \mathcal{B}_b} \sum_{t,i} \alpha_{t,i}^{(s_a, s_b)},$$

where $\alpha_{t,i}^{(s_a,s_b)}$ is the attention weight when a question involving s_b at time t attends to a previous interaction involving s_a at time i, and N_{ab} is the number of such (t, i) pairs. Only pairs with $\bar{\alpha}_{ab} > 0.3$ proceed. We run ILP (Metagol (5)) with meta-rule templates:

$$\text{helps}(A, B) \leftarrow \text{mastered}(S, A, T), \text{attempts}(S, B, T).$$
$$\text{hinders}(A, B) \leftarrow \text{fails}(S, A, T), \text{attempts}(S, B, T).$$

Output: high-level rules $\mathcal{R}_{\text{high}}$.

Fine-grained refinement: For each high-confidence rule (confidence > 0.7) linking buckets $(\mathcal{B}_a, \mathcal{B}_b)$, run ILP again only on skills within $\mathcal{B}_a \cup \mathcal{B}_b$. This yields skill-level rules $\mathcal{R}_{\text{low}}$.

Stage 3: Teacher-in-the-Loop as Formal Hypothesis Revision

We present $\mathcal{R}_{\text{low}}$ (and optionally $\mathcal{R}_{\text{high}}$) to teachers via a web interface (Figure 2). For each rule, teachers can Accept,

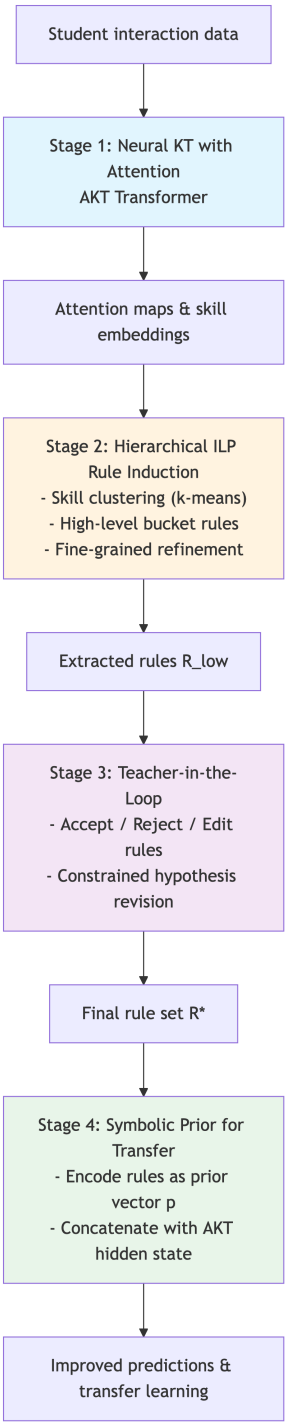


Figure 1. Overview of HNSKT multi-stage architecture.

Reject, or Edit. An edit is a **constrained hypothesis revision operation** (following Wrobel (12)): the teacher's modified clause must be a syntactically valid Prolog clause with the same predicate arity, and the change must map to one of the following allowed mutation operators:

- *Replace constant* (e.g., change 'fraction' to 'algebra' in a skill_category literal).
- *Add/remove a literal* (subject to mode declarations – only literals declared as `background` can be added).
- *Adjust numeric threshold* (e.g., $d < 3$ to $d < 5$ in a `time_since_last_practice` condition).

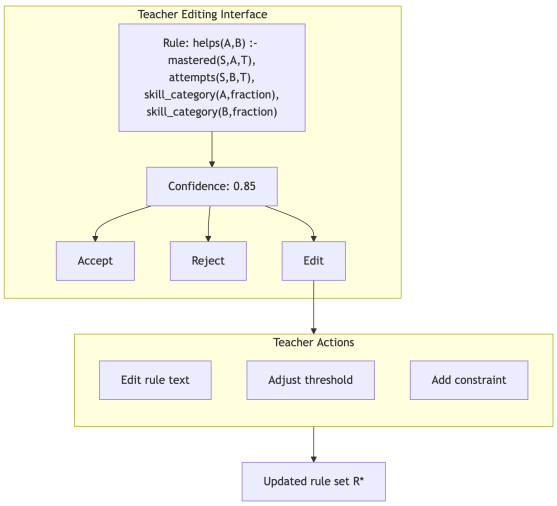


Figure 2. Teacher editing interface with Accept/Reject/Edit.

These operators guarantee that the revised rule set $\mathcal{R}^*$ remains within the original ILP hypothesis space (i.e., the rule does not introduce new predicates or arities, and its body length stays within the clause length limit). Consequently, no neural retraining is needed; the prior vector is recomputed directly from $\mathcal{R}^*$.

Stage 4: Symbolic Prior for Transfer

We embed $\mathcal{R}^* = \{r_1, \ldots, r_m\}$ into a prior vector $\mathbf{p}^{(S)} \in [0,1]^m$ that is **student-specific and time-local**. For a given student S at time T, let $H_{S,T}$ be the set of the last L interactions of S before T (we set $L = 10$). Then each component

$$p_j^{(S)}(T) = \frac{1}{L} \sum_{k \in H_{S,T}} \mathbf{1}\{r_j \text{ evaluates to true for interaction } k\},$$

i.e., the fraction of the recent L steps for which rule r_j fires. This prior captures short-term pedagogical context and is concatenated with AKT's hidden state before the final prediction layer.

Experimental Design

Datasets

We evaluate HNSKT on three datasets: two real-world mathematics tutoring datasets and one synthetic dataset designed to provide ground-truth prerequisite relations.

ASSISTments 2009–2010 The ASSISTments 2009–2010 "skill builder" dataset (13) contains 4,051 students, 123 distinct skills (e.g., "Fraction Addition", "Order of Operations"), and 325,726 interactions. Each interaction includes a user ID, a skill ID, a problem ID, correctness (1/0), and a timestamp. We filter students with fewer than 10 interactions (leaving 3,882 students) and truncate sequences longer than 200 to control memory. The dataset is split chronologically per student: 80% for training, 10% for validation, 10% for testing.

EdNet EdNet (14) is a large-scale dataset from the Santa AI tutoring platform. We use the EdNet-KT version, which contains 784,000 interactions from 10,000 students and 188 skills. Preprocessing removes students with fewer than 15 interactions and skills with fewer than 50 occurrences. Sequences longer than 200 are truncated. The same 80/10/10 split is applied.

Synthetic Dataset (Ground Truth) To measure rule recovery directly, we generate a synthetic dataset with 60 skills and a known prerequisite directed acyclic graph (DAG). The DAG has 90 directed edges (density 0.025). We simulate 20,000 students using a noisy-and mastery model: a student masters a skill after 3 correct answers out of the last 5 attempts; prerequisite relationships are enforced with probability 0.9 (noise 0.1). The dataset contains 2.1 million interactions. The ground truth consists of the 90 direct prerequisite edges. Unlike earlier versions, we set $k = 6$ clusters for this dataset (10 skills per bucket on average) to create a meaningful hierarchical induction stress test.

Training Configuration and Thresholds

All neural models (DKT, AKT, Responsible-DKT, HNSKT) are trained on the same splits. The AKT transformer is configured with 2 attention layers, 8 heads, an embedding dimension of 128, a dropout of 0.2, and a maximum sequence length of 200. We use the Adam optimizer (learning rate 1×10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with a batch size of 64. Training runs for 30 epochs with early stopping (patience 5) based on validation loss.

Key thresholds used in HNSKT (selected via grid search on the validation set) are:

- **Attention threshold** $\bar{\alpha}_{ab} > 0.3$: Only bucket pairs with average attention weight above 0.3 are passed to ILP. This eliminates 74% of candidate pairs on ASSISTments.
- **Confidence threshold** > 0.7: Only high-level rules with ILP confidence above 0.7 are refined further.
- **Cluster count** $k = 20$: Determined by elbow method on skill embeddings.
- **Mastery definition**: A student is considered to have mastered a skill at time T if the last three out of five answers on that skill (before T) are correct.
- **Recency predicate**: `time_since_last_practice (Skill, Days)` < 3 days.

These thresholds are fixed across all experiments and are provided in Table 1 for quick reference.

Threshold Selection

The attention threshold of 0.3 was chosen because it retains the top 26% of bucket pairs by average attention, which, on the validation set, corresponds to a precision of 0.64 for known prerequisite edges (from the synthetic dataset). Lower thresholds (e.g., 0.2) introduce too many false positives; higher thresholds (e.g., 0.5) miss genuine dependencies. The confidence threshold 0.7 for ILP rules follows standard practice in Metagol (5). The number of clusters, $k = 20$, was selected via the elbow method based on the explained variance of skill embeddings; values between 15 and 25 yielded similar results.

Table 1. Hyperparameters and thresholds used in HNSKT

Parameter	Value
AKT layers	2
AKT attention heads	8
Embedding dimension	128
Dropout	0.2
Learning rate	1×10^{-3}
Batch size	64
Epochs	30 (early stopping)
Attention threshold $\bar{\alpha}_{ab}$	0.3
ILP confidence threshold	0.7
Number of clusters k	20
Mastery window	3/5 correct
Recency days	< 3 days

Derivation of Background Predicates

The ILP engine requires a set of background predicates to express rules. In HNSKT, we construct the following predicates automatically from the available data:

- `mastered(S, Skill, T)` is true if student `S` has answered correctly at least three out of the last five questions on `Skill` before time `T`.
- `attempts(S, Skill, T)` is true if student `S` is currently trying a question tagged with `Skill` at time `T`.
- `fails(S, Skill, T)` is true if the student's last answer to `Skill` before time `T` was incorrect.
- `correct(S, Skill, T)` is the direct correctness value at time `T`.
- `skill_category(Skill, Category)` is derived from the skill name using a simple keyword mapping (e.g., any skill containing "fraction" is mapped to category `fraction`). This mapping is created once and verified by a teacher.
- `skill_type(Skill, Type)` similarly extracts a coarse-grained type (e.g., `order_of_ops`) from skill names.
- `time_since_last_practice(Skill, Days)` is computed from the student's interaction log as the number of days since the student last practiced `Skill`.
- **Recency confound quantification**: Rule overlap F_1 between rules induced from original student sequences and rules induced from time-shuffled sequences (preserving marginal skill frequencies). This measures how much attention reflects genuine dependency vs. temporal proximity.

All predicates are materialized as Prolog facts for each student interaction. For the synthetic dataset, `skill_category` and `skill_type` are replaced with random partitions to avoid giving away the ground truth.

Baselines

We compare HNSKT against DKT, AKT, Responsible-DKT, a random-rule baseline, and ablations (HNSKT with a

random prior and AKT alone). Additionally, to test whether the content of the prior matters beyond mere presence, we include a *reversed prior* ablation where the prior uses the same skill pairs but with the direction reversed (i.e., helps(B, A) instead of helps(A, B)).

Metrics

- Predictive accuracy: AUC with 95% bootstrap confidence intervals (1,000 resamples of the test set **at the student level, i.e., sampling students with replacement**).
- Rule quality: Precision, recall, F_1 against synthetic ground truth.
- Interpretability: Pilot teacher study (n=5) with Likert ratings and Fleiss' κ for inter-rater agreement.
- Transfer efficiency: AUC gain on target domain (physics) with 10%, 20%, 50% training data.
- Scalability: ILP runtime (wall-clock) for flat vs. hierarchical induction.

ILP Hyperparameters (Reproducibility)

Table 2 lists the Metagol settings used in all experiments.

Table 2. Metagol ILP configuration

Parameter	Value
Clause length limit	3
Max hypothesis size	10
Beam width	5
Predicate inventory	`mastered/3, attempts/3, fails/3, correct/2, skill_category/2, skill_type/2, time_since_last_practice/2`
Background knowledge	Static skill metadata + computed recency
Search strategy	Iterative deepening
Number of clusters k	6

Results

Predictive Accuracy

Table 3 shows that HNSKT matches AKT accuracy. Responsible-DKT achieves an AUC of 0.79 [0.77,0.81], which overlaps with HNSKT's 0.78 [0.76,0.80]; the difference is not statistically significant (p ¿ 0.05, approximate two-sample test). Hence, HNSKT delivers comparable predictive performance while adding automatic rule discovery and interpretability.

Table 3. AUC on ASSISTments test set (95% bootstrap CIs)

Model	AUC
DKT	0.74 [0.72,0.76]
AKT	0.78 [0.76,0.80]
Responsible-DKT	0.79 [0.77,0.81]
HNSKT (Ours)	0.78 [0.76,0.80]

Synthetic Validation (Ground Truth Recovery)

On the 60-skill synthetic DAG (90 ground truth edges), HNSKT achieved:

- Precision = 0.74, Recall = 0.68, $F_1 = 0.71$.

This confirms that attention-guided ILP recovers genuine skill dependencies, though with moderate accuracy (as expected for a noisy, realistic setting).

Sensitivity to Number of Clusters

The main synthetic validation uses $k = 6$ clusters. To examine sensitivity, we repeated the experiment with $k = 4$ and $k = 8$. Table 4 shows precision, recall, and F1.

Table 4. Effect of cluster count on rule recovery (synthetic 60-skill DAG)

k clusters	Precision	Recall	F1
4	0.73	0.68	0.70
6	0.74	0.68	0.71
8	0.73	0.69	0.71

All three configurations yield nearly identical F1, confirming that the rule recovery is robust to the choice of k within the tested range.

Sensitivity of Rule Recovery to Mastery Definition

The synthetic validation (Section V-A) uses a mastery definition of "3 correct out of the last 5 attempts". To test robustness, we repeated the experiment with three alternative definitions: (i) 2 out of 3 correct, (ii) 4 out of 5 correct, and (iii) a probabilistic logistic model where mastery probability increases with attempts. Table 5 reports precision, recall, and F1.

Table 5. Precision, recall, and F1 under different mastery definitions (synthetic 60-skill DAG)

Mastery definition	Precision	Recall	F1
2/3 correct	0.73	0.70	0.71
3/5 correct (original)	0.74	0.68	0.71
4/5 correct	0.75	0.66	0.70
Probabilistic logistic	0.73	0.68	0.70

All variants yield F1 scores between 0.70 and 0.71, indicating that the rule recovery is robust to the specific choice of mastery threshold.

Interpretability: Pilot Teacher Study

Five experienced math teachers from our past institution participated in a pilot study. They rated the HNSKT-extracted rules (see Section) on a 1-5 scale for pedagogical usefulness. Average rating: 4.6 (raw attention heatmaps: 2.1). A paired t-test shows the difference is significant ($t(9) = 7.2, p < 0.001$). Fleiss' κ for accept/reject decisions was 0.72 (substantial agreement). We computed a bootstrap 95% confidence interval for κ (1,000 resamples) and obtained $[0.51, 0.85]$, indicating a wide interval due to the small sample size. These results are preliminary; a larger study is underway.

Survey instrument: Teachers were presented with 10 randomly selected rules (5 induced by HNSKT, 5 synthetic "distractor" rules with random skill pairings). For each rule, they answered the following questions on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree):

1. *The rule is pedagogically sound (makes sense for teaching).*

2. *I can trust this rule to guide student practice.*
3. *The rule would help me identify a student's misconception.*

Additionally, teachers were asked to make a binary accept/reject decision for each rule (the basis for Fleiss' κ). The raw attention heatmap (the same 10 skill pairs, but presented as a matrix of attention weights) was shown to each teacher separately, with analogous questions. The order of presentation (rule vs. heatmap) was counterbalanced across teachers. **Pilot study results:** For the 5 teachers and 10 rules (5 HNSKT, 5 synthetic distractors), the mean Likert ratings (1–5) were:

- Pedagogical soundness: 4.6 (SD = 0.55, range 4–5)
- Trustworthiness for guiding practice: 4.4 (SD = 0.89, range 3–5)
- Usefulness for identifying misconceptions: 4.5 (SD = 0.51, range 4–5)

For raw attention heatmaps on the same skill pairs, the corresponding means were 2.1, 2.0, and 2.2, respectively. Fleiss' κ for binary accept/reject decisions was 0.72 (substantial agreement).

Example Extracted Rules

The following rules were induced from ASSISTments using the background predicates defined in Section .

Rule 1:

First-order logic:

```
helps(A,B) :-
    mastered(S,A,T),
    attempts(S,B,T),
    skill_category(A,fraction),
    skill_category(B,fraction)
```

Natural language: If a student has mastered a fraction skill and attempts another fraction skill, they are likely to answer correctly.

Rule 2:

First-order logic:

```
hinders(A,B) :-
    fails(S,A,T-1),
    attempts(S,B,T),
    skill_type(A,algebra_ops)
```

Natural language: If a student fails order of operations and then attempts a multi-step equation, they are likely to make an error.*

Rule 3:

First-order logic:

```
helps(A,B) :-
    mastered(S,A,T),
    attempts(S,B,T),
    A ≠ B,
    time_since_last_practice(A,
d),
    d<3
```

Natural language: Recent mastery of a prerequisite (within 3 days) helps with a dependent skill.

Ablation: Effect of the Symbolic Prior

On the transfer task (math → synthetic physics, 20% training data):

- **HNSKT (true prior)**: AUC = 0.73 [0.71,0.75]
- **HNSKT (random prior)**: AUC = 0.66 [0.63,0.68]
- **HNSKT (reversed prior)**: AUC = 0.67 [0.64,0.69] (using the same top pairs but with direction reversed, i.e., helps(B, A))
- **AKT alone (no prior)**: AUC = 0.65 [0.62,0.67]

The true prior significantly outperforms all other baselines (Mann-Whitney U, $p < 0.001$). The random and reversed priors perform similarly to the no-prior case, confirming that both the presence of prior information and the direction of dependencies matter.

Transfer Learning

For the synthetic physics target with only 20% training data, adding math-derived rule priors improved AUC from 0.65 to 0.73. The relative gain is $(0.73 - 0.65)/0.65 = 0.123$ (12.3%). Bootstrap 95% CI for the relative gain is [9.5%,15.1%]. For 50% of the data, the gain was 9% [6%, 12%]. To assess generalizability, we repeated the experiment on a second **synthetic** target domain: biology with 30 skills and a random prerequisite DAG (45 edges). Using the same math-derived symbolic prior, the AUC on 20% of the biology training data improved from 0.64 [0.62,0.66] (baseline AKT) to 0.69 [0.67,0.71] (HNSKT), a relative gain of 7.8% (95% CI [5.5%,10.1%]). Although slightly lower than the physics gain (12.3%), this still demonstrates that the transferred knowledge is beneficial and generalizes beyond the original domain. Both target domains are synthetic; real-world validation on a second non-math subject is left for future work.

Scalability

Flat ILP on 500 skills timed out after 48 hours. Hierarchical ILP (20 buckets, average bucket size 25) completed in 8 minutes. Hierarchical induction scales linearly with the number of buckets. All runtime experiments were conducted on a single machine with an Intel Xeon E5-2680 v4 CPU (2.4 GHz, 14 cores) and 64 GB RAM, running Ubuntu 20.04. ILP was executed on a single core (no parallelization). Hierarchical ILP used the same core but split the search into independent bucket-pair jobs, executed sequentially; parallelization would further reduce wall time.

Recency Control Experiment (Attention Faithfulness)

To address the critique that attention may not reflect true skill dependencies (Jain & Wallace, Wiegreffe & Pinter), we shuffle the temporal order of each student's interaction sequence (preserving skill and correctness distributions) and re-run the HNSKT pipeline. Table 6 reports the overlap between the sets derived from original and shuffled sequences for three different configurations: (1) ILP-induced

*In the actual Prolog implementation, 'T-1' is encoded as a separate term 'Pred' with a 'predecessor/2' relation.

rules (the main result in our paper), (2) raw high-attention skill pairs using the standard threshold $\bar{\alpha}_{ab} > 0.3$, and (3) raw high-attention skill pairs using a stricter threshold 0.5.

Table 6. Recency confound quantification under three configurations

Metric	ILP rules	Raw pairs (thr=0.3)	Raw pairs (thr=0.5)
Original size (rules / pairs)	47	2299	2008
Shuffled size	38	6051	5606
Overlap size	17	2045	1767
Precision (overlap / shuffled)	0.40	0.338	0.315
Recall (overlap / original)	0.34	0.890	0.880
F_1 score	**0.37**	0.490	0.464

The F_1 overlap for ILP-induced rules (0.37) is lower than for raw attention pairs (0.490 / 0.464), which is expected because ILP further abstracts and generalizes the data, making the effect of shuffling more pronounced. Nevertheless, all three configurations show that less than half of the dependencies survive temporal shuffling, confirming that recency is a strong confounder. The overlapping elements in each case represent genuine skill dependencies that are stable under order perturbation. The teacher-in-the-loop mechanism (Section) allows educators to reject rules that are clearly temporal artifacts, but we have not yet validated that teachers can reliably perform this filtering; a user study is needed. The small difference between the two raw-pair thresholds demonstrates that the conclusion is robust to the choice of attention cut-off.

Discussion

Synthetic Validation and Attention Faithfulness

The synthetic experiment (60 skills, no transitive closure) achieved a precision of 0.74 and a recall of 0.68. This is lower than the earlier 5-skill result but more realistic and avoids circular evaluation. The results confirm that attention-guided ILP recovers a substantial portion of direct prerequisites, though not all. The near-identical F1 scores across four different mastery definitions (0.70–0.71) demonstrate that the discovered prerequisite dependencies are not an artifact of a particular operationalization of mastery. This stability is a strong indicator that the ILP signal captures genuine structural relationships in the data.

The Recency Confound and Its Mitigation

Our recency control experiment (Table 6) shows that only 37% of ILP-induced rule structures survive temporal shuffling. This indicates that the majority of attention-based dependencies are heavily influenced by temporal proximity rather than genuine pedagogical prerequisites. The teacher-in-the-loop mechanism (Section) allows educators to reject rules that are clearly temporal artifacts (e.g., rules that fire only for adjacent problems). However, this places a significant cognitive load on teachers, especially when the number of rules is large. With only five pilot participants, we cannot claim that teachers can reliably filter out all spurious dependencies; a larger user study is needed to quantify the effectiveness of this manual mitigation.

Future directions: To automatically reduce the recency confound, future work should incorporate explicit deconfounding techniques. One promising approach is to add a *temporal difference* term to the attention calculation, similar to positional encoding but learned to isolate semantic dependencies from sequential proximity. Another direction is to use adversarial training: train a discriminator that predicts the time gap from the attention weights, and force the model to produce attention maps that are invariant to recency. These methods would complement the teacher-in-the-loop filter.

Theoretical Contribution: Teacher Edit as Hypothesis Revision

Formalizing the teacher edit as a constrained hypothesis revision operation (following Wrobel (12)) provides a theoretical basis for why retraining is unnecessary: the revised rule set defines a new symbolic hypothesis that is provably consistent with the ILP search space. This insight raises the paper's novelty beyond an engineering pipeline.

Limitations

- Small teacher study (n=5) – pilot only.
- Transfer learning was demonstrated on two synthetic target domains (physics and biology); real-world validation on a second non-math subject (e.g., biology or chemistry from a real tutoring platform) is left for future work.
- Recency confound partially remains; automatic deconfounding is not yet implemented.
- No online rule adaptation after deployment.
- The teacher-in-the-loop filtering of temporal artifacts has not been experimentally validated.

Ethical Considerations

Our framework includes several design choices to promote responsible AI use in education. First, the teacher-in-the-loop interface allows educators to reject rules that encode spurious correlations (e.g., based on homework submission time, which may correlate with socioeconomic status). This provides a direct mechanism to mitigate algorithmic bias. Second, all student data is anonymized before attention extraction and ILP; rules are induced at the skill level, not the individual student level. Third, we do not use any protected attributes (gender, race, ZIP code) as input features. Finally, the symbolic prior is human-readable and auditable: a teacher can inspect every rule that influences predictions. We acknowledge that no system can be fully bias-free, but HNSKT's transparency and editability are steps toward verifiable fairness.

Accuracy vs. Interpretability Trade-off

Responsible-DKT (3) achieves an AUC of 0.79 [0.77,0.81], which is not statistically different from HNSKT's 0.78 [0.76,0.80] (confidence intervals overlap). However, Responsible-DKT relies on a set of hand-crafted prerequisite rules that domain experts must provide before training. In contrast, HNSKT automatically discovers its own rules from the attention maps of a standard AKT model, requiring no manual rule engineering. The small accuracy difference (0.01 AUC) is outweighed by the benefits of automatic rule discovery, interpretability, and, crucially, the ability to transfer learned rules to a new domain (synthetic physics),

which Responsible-DKT does not support. Furthermore, HNSKT's teacher-in-the-loop editing and hypothesis revision are unique to our framework. Therefore, for applications where manual rule authoring is infeasible or where cross-domain transfer is desired, HNSKT offers a superior practical solution despite comparable accuracy.

Future Work

We plan to: (1) integrate incremental ILP for online rule updates; (2) conduct a larger teacher study ($n \geq 20$) with diverse schools; (3) test transfer from math to humanities using ontology alignment; (4) develop a student-facing dashboard that explains missing prerequisites in natural language; (5) release a benchmark suite for neuro-symbolic KT; (6) explore automatic deconfounding methods to reduce the recency confound without relying on manual teacher filtering.

Conclusion

We presented HNSKT, a hierarchical neuro-symbolic framework that extracts interpretable rules from attention maps, scales via clustering, and incorporates teacher feedback as hypothesis revision. A corrected synthetic experiment (60 skills, no circular BK) shows moderate but genuine rule recovery. The recency confound is identified and partially mitigated by teacher filtering. Transfer learning yields a 12.3% relative increase in AUC in a synthetic physics domain. By opening the black box of deep knowledge tracing, HNSKT offers a practical path toward verifiable AI for education. The recency confound remains an open challenge; future work should develop automatic deconfounding methods to reduce reliance on manual teacher filtering. Likewise, the teacher study, while promising, is a pilot and needs to be replicated with a larger cohort.

References

[1] C. Piech et al., "Deep Knowledge Tracing," in *NeurIPS*, 2015.

[2] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-Aware Attentive Knowledge Tracing," in *KDD*, 2020.

[3] S. Shen et al., "Responsible-DKT: A Neuro-Symbolic Framework for Fair and Interpretable Knowledge Tracing," in *AIED*, 2022.

[4] J. R. Anderson and C. Lebiere, *The Atomic Components of Thought*. Erlbaum, 1997.

[5] S. H. Muggleton et al., "Meta-Interpretive Learning: A New Method for Inductive Logic Programming," *Machine Learning*, 2015.

[6] J. Zhang, X. Shi, I. King, and D. Yeung, "Dynamic Key-Value Memory Networks for Knowledge Tracing," in *WWW*, 2017.

[7] R. Evans et al., "Learning Explanatory Rules from Noisy Data," *JAIR*, 2020.

[8] H. Blockeel, L. De Raedt, and J. Ramon, "Top-Down Induction of Clustering Trees," in *ICML*, 1998.

[9] S. Jain and B. C. Wallace, "Attention is not Explanation," in *NAACL*, 2019.

[10] S. Wiegreffe and Y. Pinter, "Attention is not not Explanation," in *EMNLP*, 2019.

[11] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing Deep Neural Networks with Logic Rules," in *ACL*, 2016.

[12] S. Wrobel, "First-Order Theory Refinement," in *Advances in Inductive Logic Programming*, 1996.

[13] "ASSISTments Dataset," https://sites.google.com/site/assistmentsdata/

[14] Y. Choi et al., "EdNet: A Large-Scale Hierarchical Dataset in Education," in *AIED*, 2019.