NeSy 2026

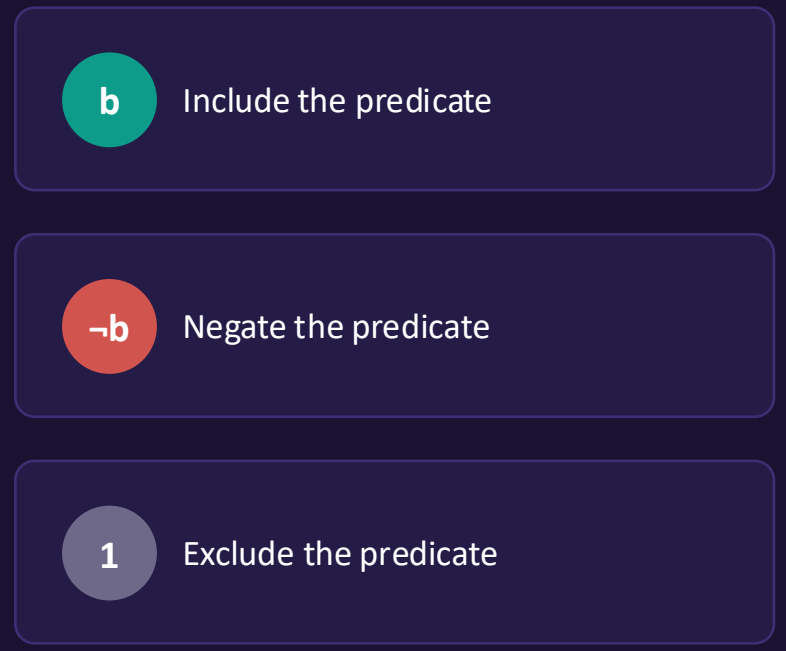
ANDRE

An Attention-based Neuro-symbolic Differentiable Rule Extractor for Inductive Logic Programming

Learning first-order rules from noisy, probabilistic data — without rule templates.

Iman Sharifi · Peng Wei · Saber Fallah

George Washington University, Washington DC, USA · University of Surrey, Guildford, UK



A continuous, template-free rule space

Where differentiable ILP breaks down

ILP learns interpretable rules — but current neuro-symbolic solvers pay for it in flexibility, gradients, and robustness.

T

Rule templates required

Template- and beam-search-based learners (∂ILP, αILP) need hand-crafted candidate rules or domain priors, limiting what can be discovered.

∧

Product-based operators

Approximating conjunction as x·y drives values toward 0 as rules grow, causing vanishing gradients and heavy bias away from the true min.

?

Noise & uncertainty

Real predicate valuations are probabilistic, not crisp Booleans. Under moderate label noise, existing methods lose the correct symbolic structure entirely.

ANDRE removes all three: no templates · attention-based logical operators · accurate rule recovery under noise

What ANDRE contributes

Rule induction reframed as optimisation over a continuous, fully differentiable rule space.

1 A continuous rule space

Inclusion, negation and exclusion of every body predicate are relaxed into a learnable probability distribution — symbolic programs become gradient-trainable.

2 Attention-based logic

Softmin / softmax attention gives a differentiable, high-fidelity approximation of min–max conjunction and disjunction over probabilistic valuations.

3 Semantic + syntactic losses

Accuracy, rule diversity, head-variable inclusion, auxiliary-variable connectivity and digitisation — with scheduled loss weights for stable convergence.

4 Validated at scale

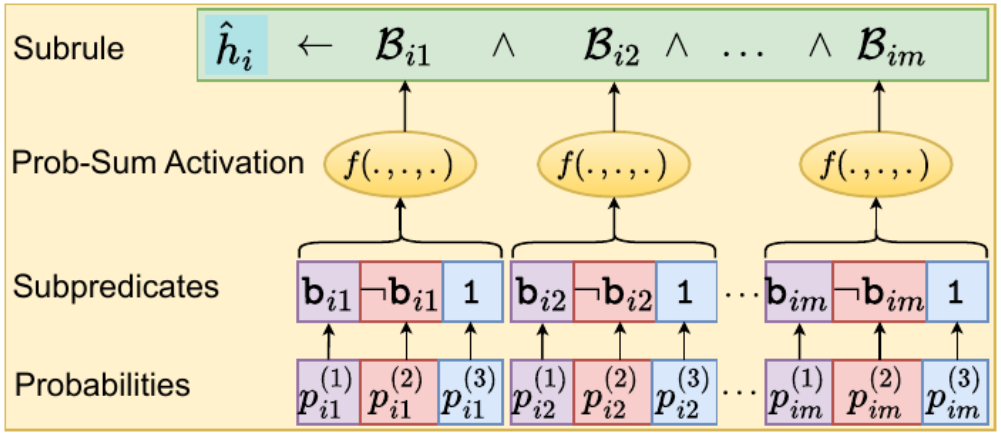
Classical ILP benchmarks, large knowledge bases, and probabilistic / noisy synthetic data — competitive or superior accuracy with far better rule recovery.

A continuous rule space

Fixed-size tensors for a variable-size symbolic problem — using the logical identity 1 ∧ b = b.

- Each candidate body predicate B_{ij} takes one of three symbolic forms: b_j, $\neg b_j$, or 1 — include, negate, or exclude.
- Exclusion is free: adding 1 to a conjunction changes nothing, exactly like zero-padding a neural input.
- A trainable tensor $P \in [0,1]^{n \times m \times 3}$ holds the probability of each choice, with $\Sigma_k p_{ij}^k = 1$.
- Element-wise product B ⊙ P weights every subpredicate; a prob-sum activation then selects the most confident one.

```
f(x,y,z) = x + y + z − xy − xz − yz + xyz    (prob-sum)
```



Subrule construction inside the rule space

Why it matters. No rule template is needed: the model searches the whole space of subrules by gradient descent and never has to be told which predicates are relevant.

Attention as a logical operator

One softmax layer can approximate Boolean AND/OR — we use its temperature to steer it toward min and max.

∧ Conjunction — softmin attention

$$C_i = \Sigma_j \; softmax_j(-\beta \cdot B_{ij}) \cdot B_{ij}$$

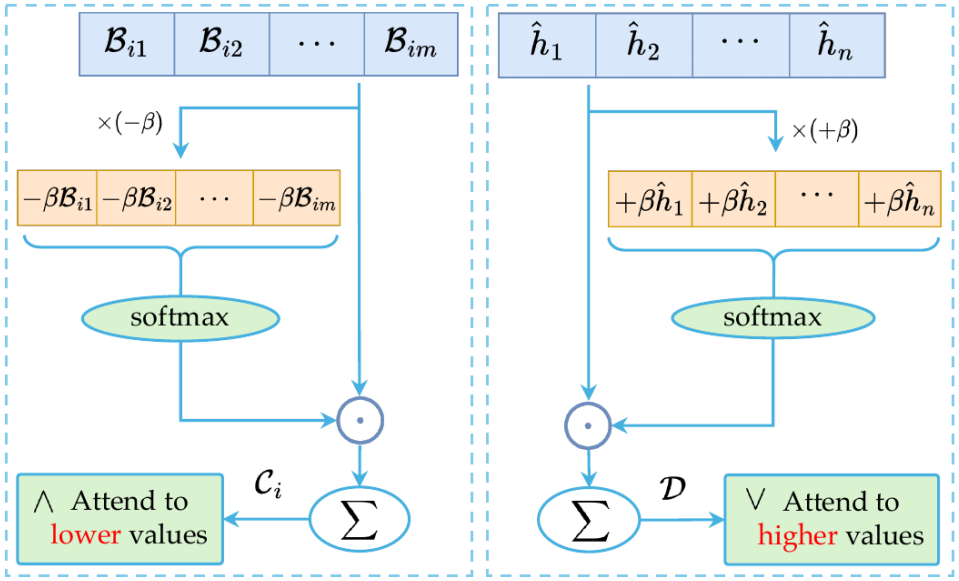
Negative scores put the most weight on the smallest valuation — a smooth, unbiased stand-in for min.

∨ Disjunction — softmax attention

$$D = \Sigma_i \; softmax_i(+\beta \cdot h_i) \cdot h_i$$

Flipping the sign of the score emphasises the largest subrule head — a smooth stand-in for max.

β controls sharpness: larger β ⇒ closer to hard min/max, but less stable. A sigmoid with steepness λ and midpoint γ binarises each output.

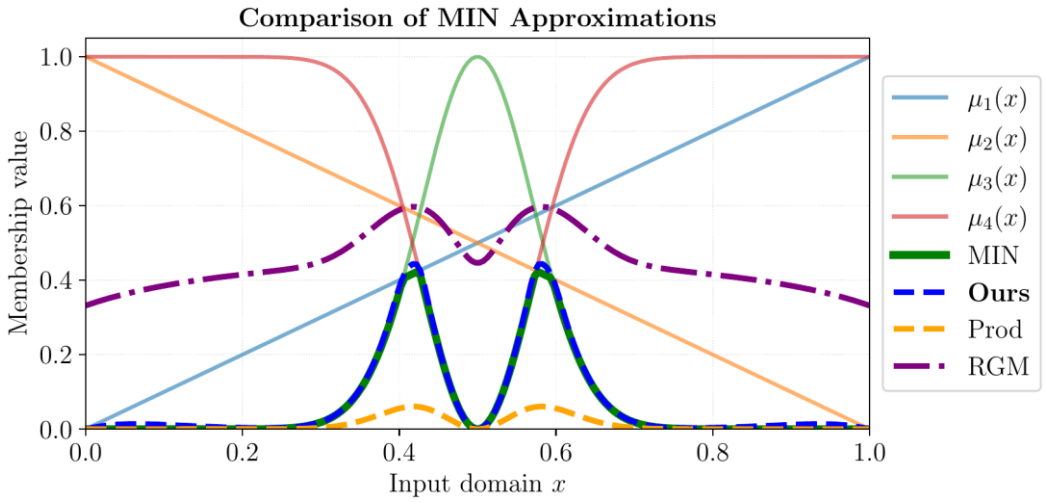


Attend to lower values for ∧ (left) and higher values for ∨ (right).

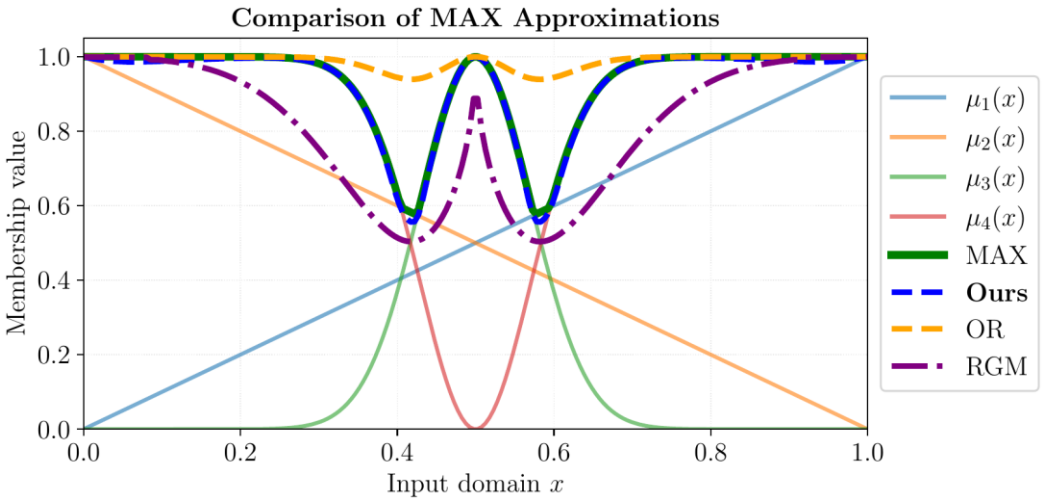
Attention beats product and RGM operators

Four fuzzy membership functions; compare each differentiable operator against the true MIN and MAX.

Conjunction: approximating MIN



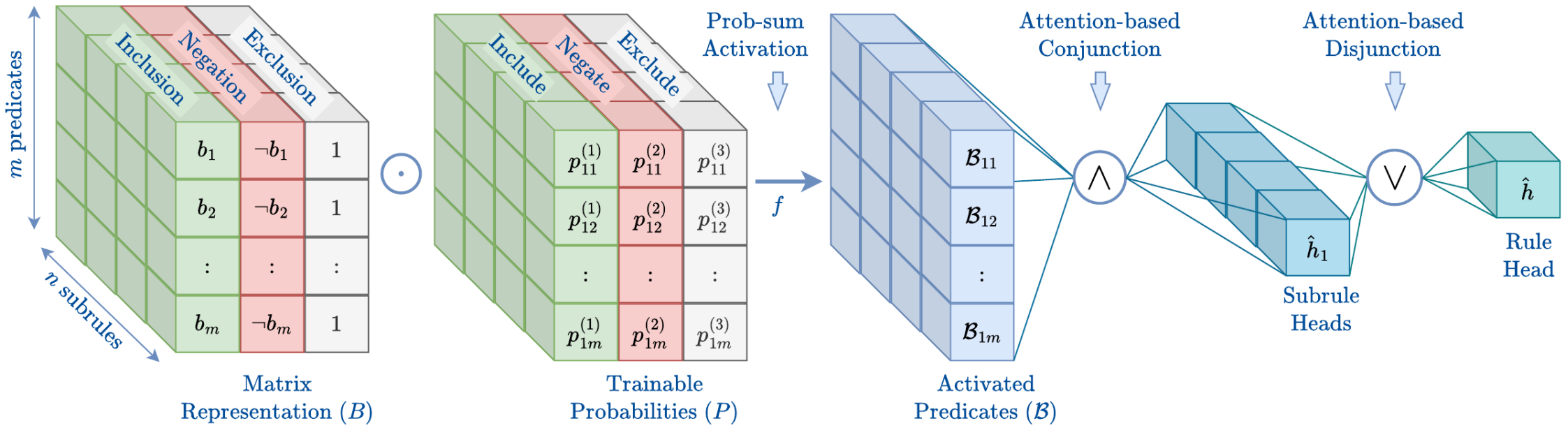
Disjunction: approximating MAX



✓ **Softmin tracks the true minimum almost exactly**, while the product t-norm collapses toward zero and RGM stays systematically biased. We prove there exists a finite β* whose approximation error is strictly smaller than the product t-norm's.

ANDRE architecture

Rule space (left) feeds the logical network (right) — end-to-end differentiable, fully vectorised, GPU-friendly.



⊙ Symbolic forms weighted by learnable probabilities

f Prob-sum activation selects the confident subpredicate

∨ Attention conjunction, then disjunction to the head

Training: accuracy that stays valid logic

Semantic losses fit the data; syntactic losses keep every learned clause a well-formed first-order rule.

Σ Semantic

Binary cross-entropy L_{BCE}
Predict the Boolean label of every example correctly.

Entropy L_E
Push each predicate's distribution to a single confident choice.

Similarity L_S
Penalise cosine similarity between subrules to keep them diverse.

∀ Syntactic

Range-restricted L_R
Every head variable must appear in the body at least once.

Connected L_C
Auxiliary variables must link two predicates, never dangle.

Digitisation L_D
Nudge expected variable counts toward integers (small weight).

$$L = L_{BCE} + \lambda_E \cdot L_E + \lambda_S \cdot L_S + \lambda_R \cdot L_R + \lambda_C \cdot L_C + \lambda_D \cdot L_D$$

λ weights are scheduled over training — structure first, strict constraints later — preventing early over-regularisation.

Classical ILP and large knowledge bases

Ten classical tasks, plus Countries, Nations, UMLS and Kinship.

100%

rule-extraction accuracy on all ten classical ILP tasks — matching DFORL, ahead of ∂ILP and NeuralLP.

Tasks covered

Arithmetic

Predecessor · Odd · Even · LessThan

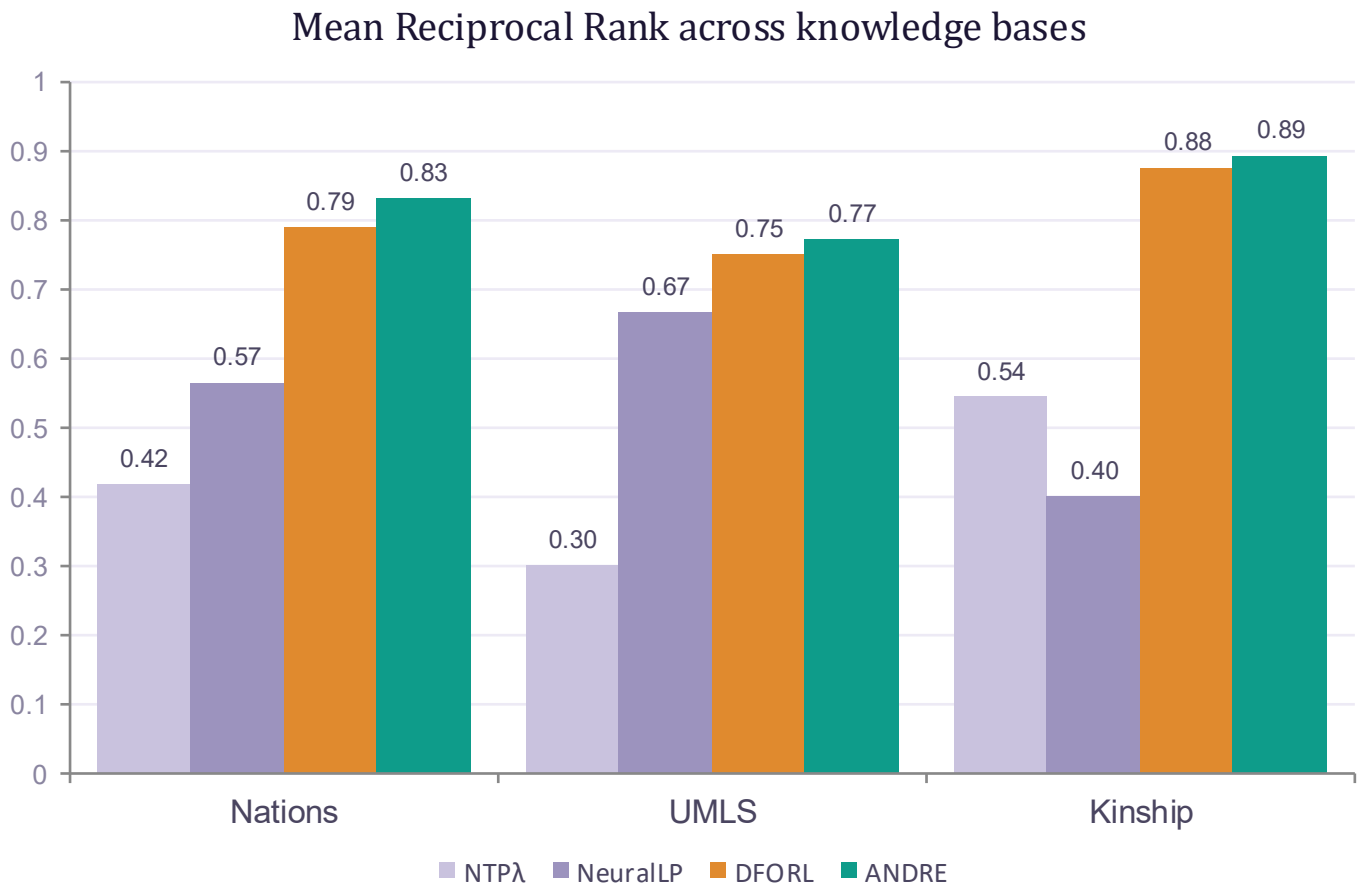
Family

Son · Grandparent · Related · Father

Graph

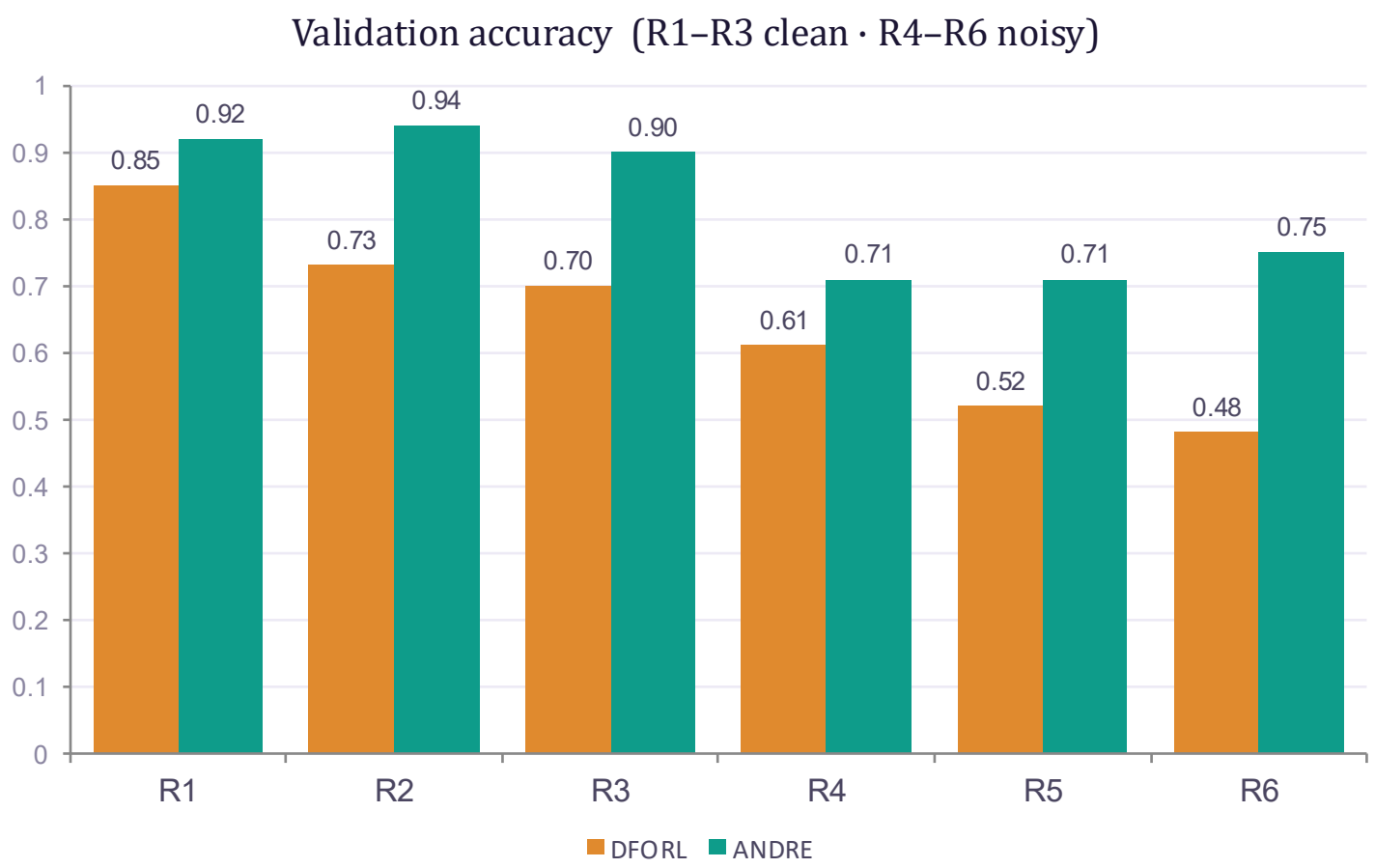
Direct Edge · Connected

All extracted clauses satisfied both syntactic constraints.



Probabilistic and noisy data — the real test

Synthetic rules R_1–R_3 of increasing depth; R_4–R_6 add 5–45% flipped labels.



+21.6%

average accuracy gain over DFORL across all probabilistic datasets

5.92×

smaller accuracy drop from R_1 to R_3 as rule depth grows

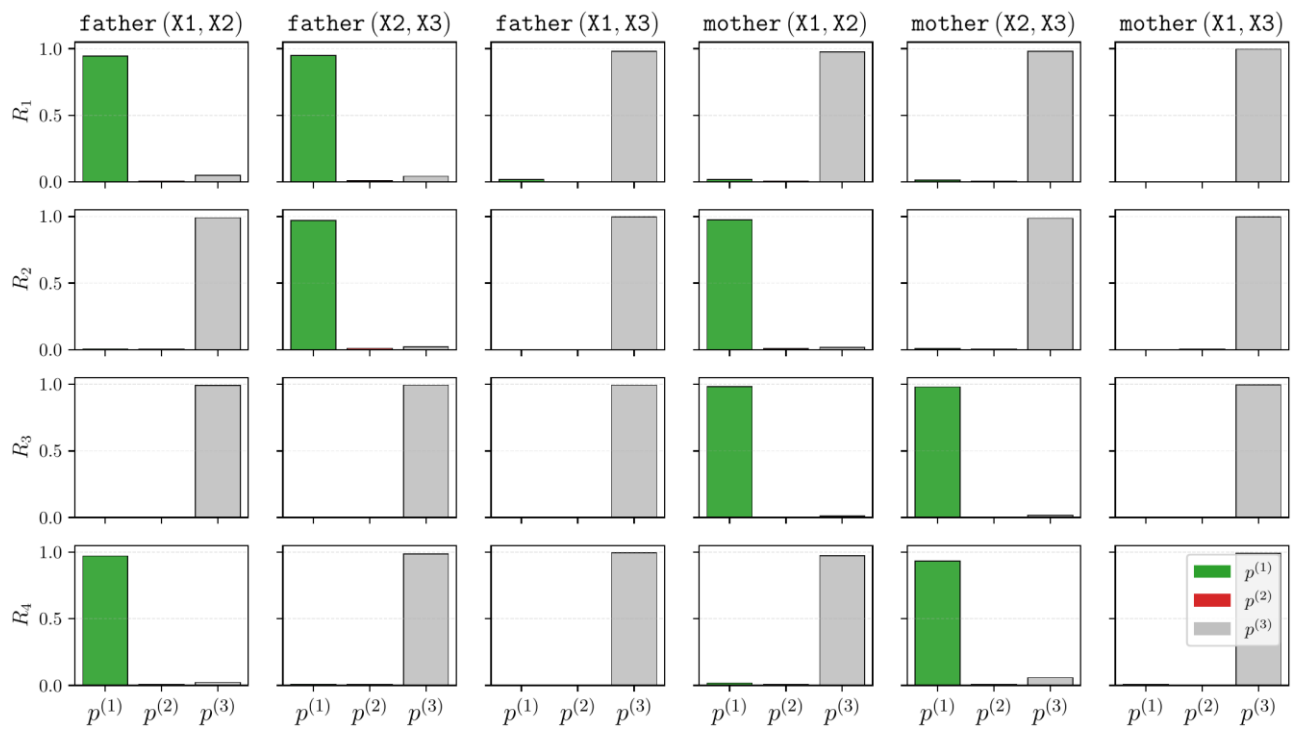
9 / 14

ground-truth rules recovered exactly under noisy supervision

Ablation — remove attention: 0.92 → 0.76

What the model actually learns

Grandparent task: converged subpredicate probabilities and the program read off from them.



Green = include · red = negate · grey = exclude, for each subrule R_1–R_4.

Extracted program

```
grandparent(X1,X3) ←
   father(X1,X2) ∧ father(X2,X3)

grandparent(X1,X3) ←
   father(X1,X2) ∧ mother(X2,X3)

grandparent(X1,X3) ←
   mother(X1,X2) ∧ father(X2,X3)

grandparent(X1,X3) ←
   mother(X1,X2) ∧ mother(X2,X3)
```

Distributions converge to one-hot: the rules are read directly off the weights, with X_2 correctly used as a connecting auxiliary variable.

Takeaways

ANDRE: rule induction as optimisation over a continuous rule space.

Template-free

A continuous rule space lets predicates be included, negated or excluded by gradient descent — no candidate clauses required.

Attention as logic

Softmin / softmax operators approximate min–max far more faithfully than product or RGM operators, and keep gradients alive.

Robust and interpretable

Competitive or better accuracy on ILP benchmarks and knowledge bases; correct rules recovered under up to 25% mislabelled data.

Limitations & next steps

· No predicate invention yet — a stacked, multi-layer ANDRE is the natural fix

· Deeper rules under heavy noise remain hard

· Next: ANDRE as a transparent policy network for RL

Thank you — questions?

i.sharifi@gwu.edu · Iman Sharifi, Peng Wei, Saber Fallah