Improving Neural Logic Machines via Failure Reflection

Zhiming Li *, Yushi Cao*, Yan Zheng, Xu Liu, Bozhi Wu, Tianlin Li,
Xiufeng Xu, Junzhe Jiang,
Yon Shin Teo, Shang-wei Lin, Yang Liu
Nanyang Technological University, Tianjin University, Hong Kong Polytechnic University, Continental Automotive Singapore Pte. Ltd.

Background – Inductive Logic Programming

- A programming paradigm based on first-order logic
- Key Concepts:
  - Predicate
  - Variable
  - Atom
  - Rule

$$\texttt{HasSister}(x) \leftarrow \exists y, z, \texttt{IsDaughter}(z, y) \wedge \texttt{IsMother}(x, z)$$

Background – Inductive Logic Programming

- A programming paradigm based on first-order logic
- Key Concepts:
  - Predicate
  - Variable
  - Atom
  - Rule
  - Premise
  - Positive examples
  - Negative examples

$$\texttt{HasSister}(x) \leftarrow \exists y, z, \texttt{IsDaughter}(z, y) \wedge \texttt{IsMother}(x, z)$$

$$\mathbf{B} = \{\texttt{IsMother}(C, B), \texttt{IsMother}(D, B), \texttt{IsDaughter}(B, C), \texttt{IsDaughter}(B, D), \texttt{IsDaughter}(A, C), \texttt{IsDaughter}(A, D)\}$$

$$\rho = \{\texttt{HasSister}(C), \texttt{HasSister}(D)\}$$

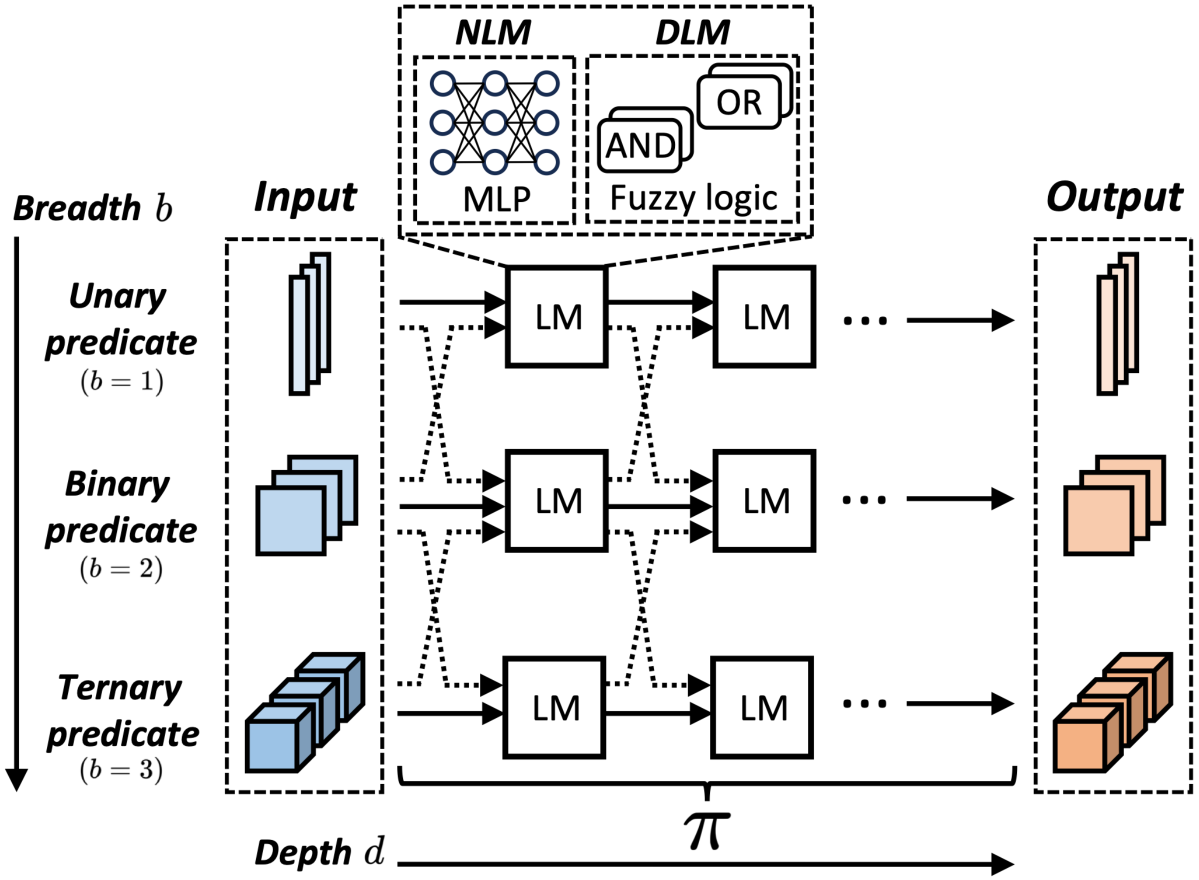
$$\eta = \{\texttt{HasSister}(A), \texttt{HasSister}(B)\}.$$

Background – Neural Logic Machines

- Using neural network to approximate logic operations

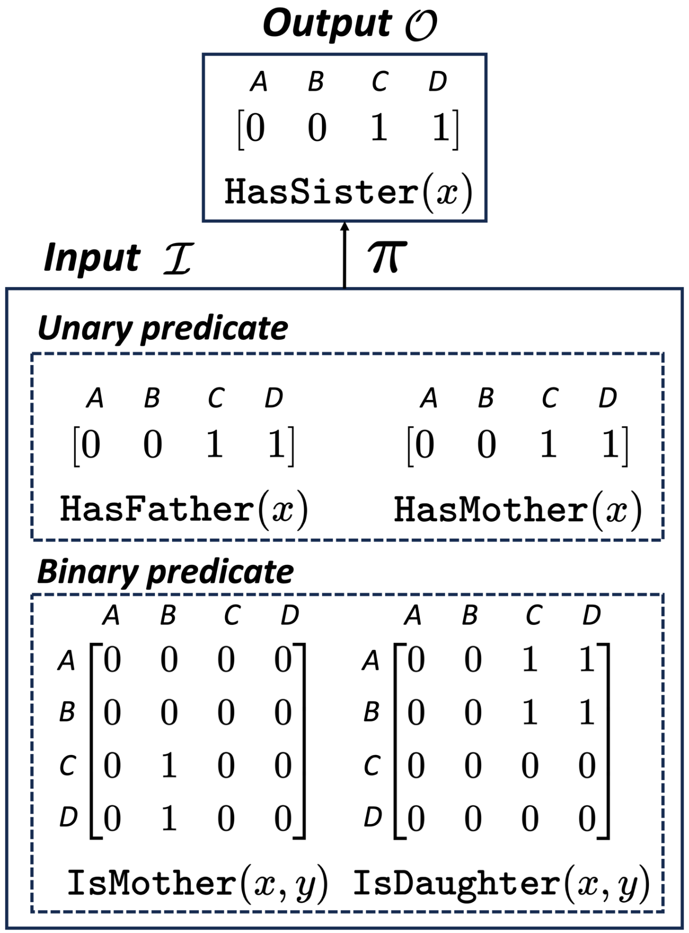
$$\forall x_{b+1}\ q\left(x_1, x_2, \cdots, x_b, x_{b+1}\right) \leftarrow p\left(x_1, x_2, \cdots, x_b\right)$$

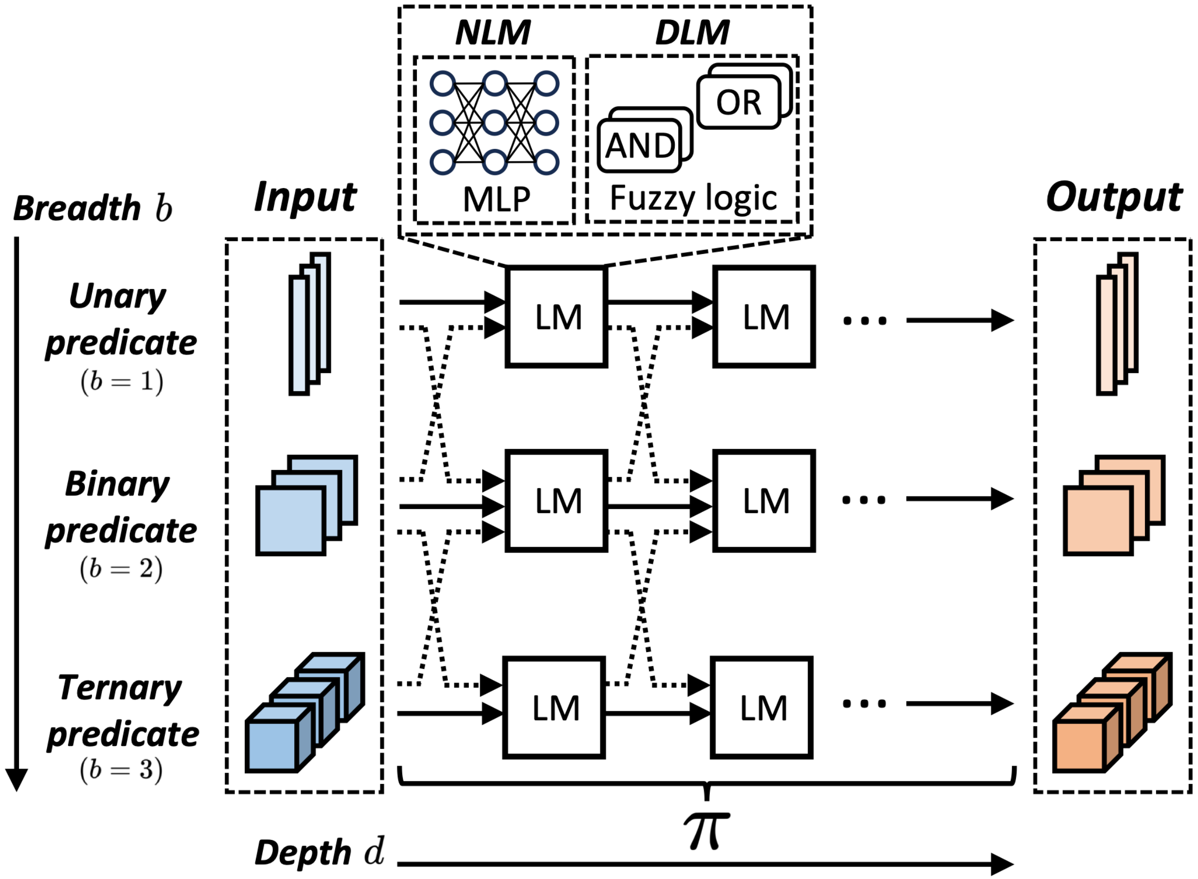
- DLM is a variant of NLM, replaces the MLP with fuzzy logic



Background – Neural Logic Machines

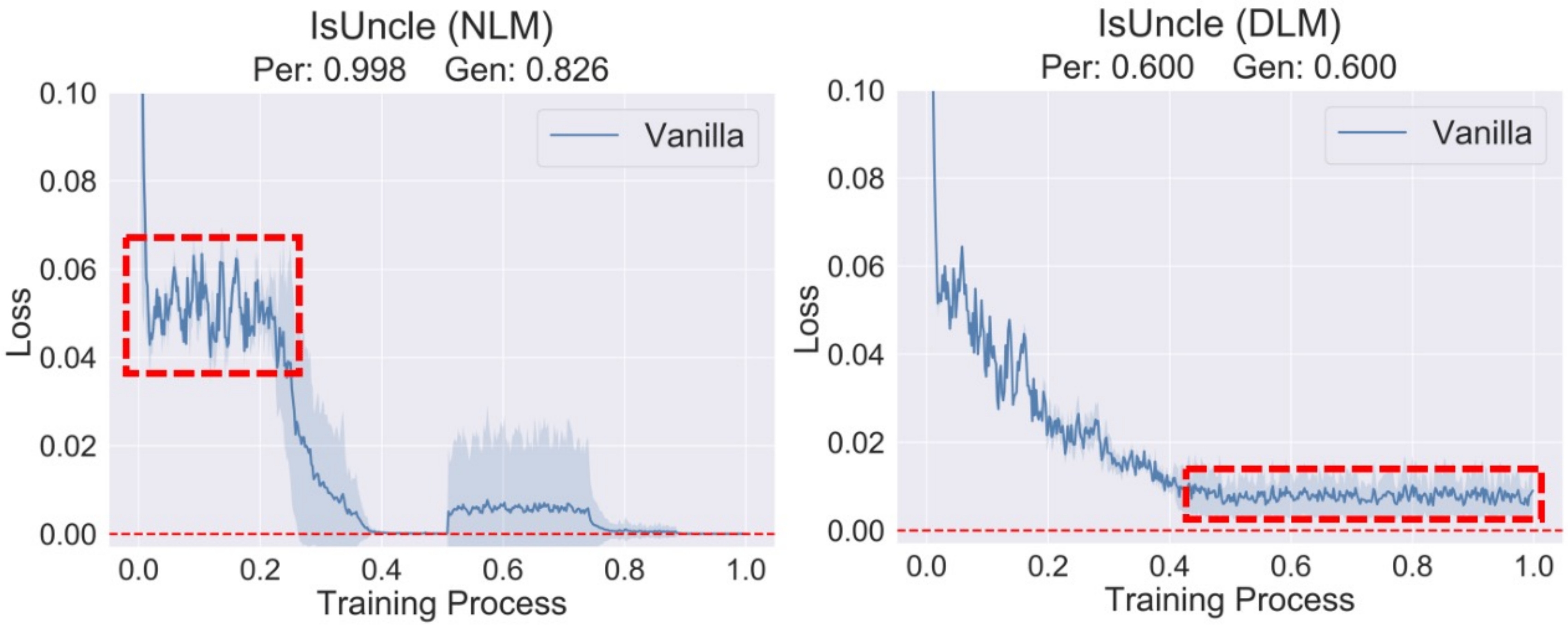
- Taking truth value tensor as input and output



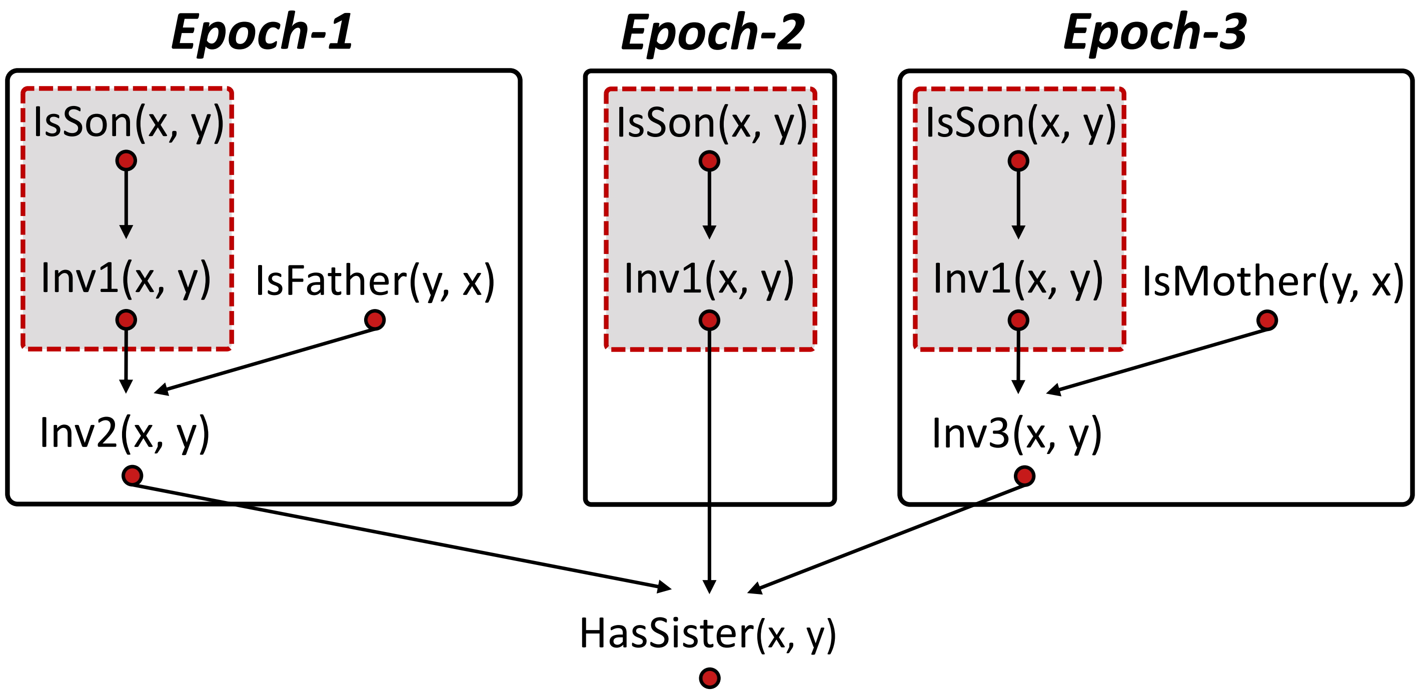


Motivation

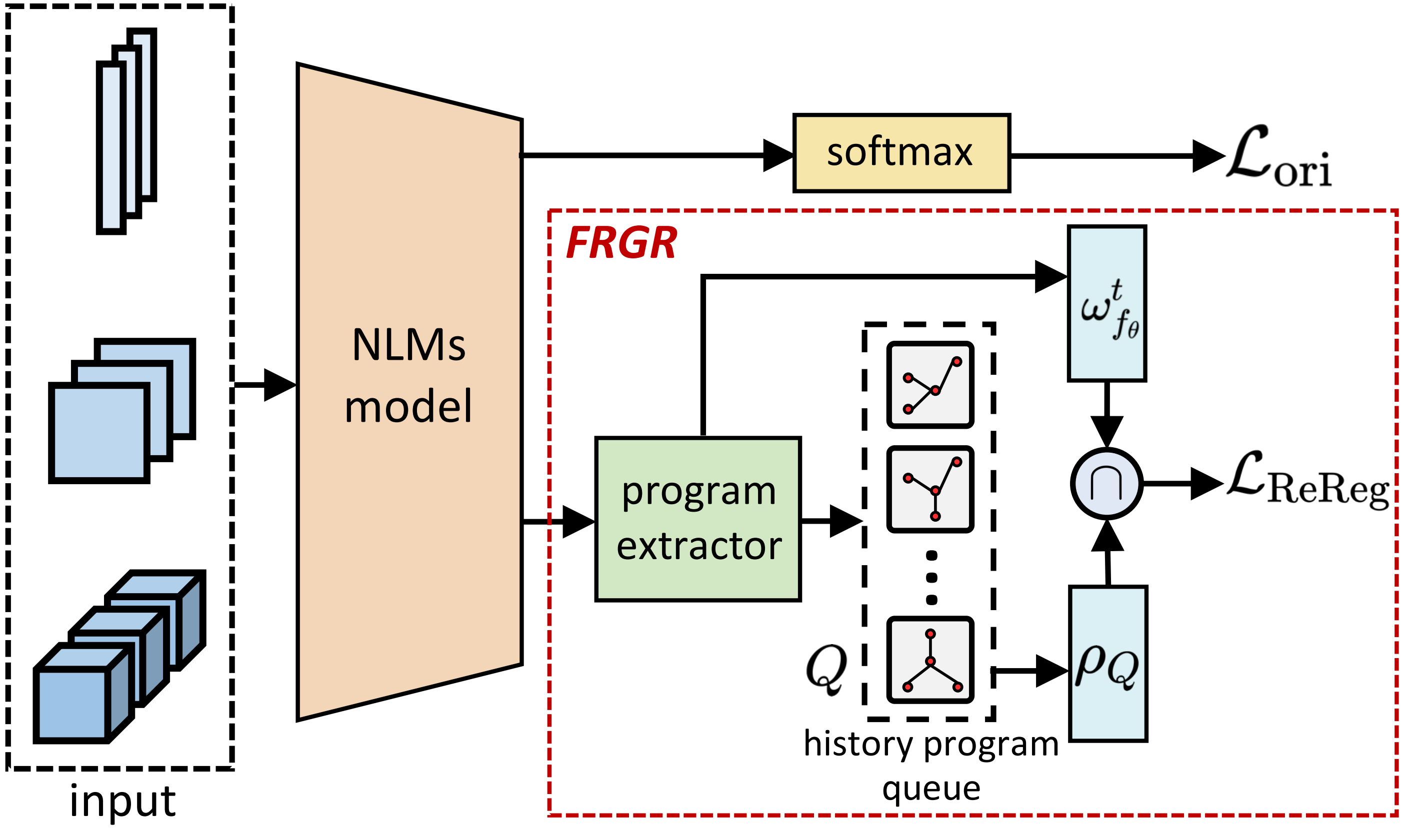
- Training approaches are far from perfect
- NLMs would repeat similar mistakes
- Oscillating for a long while



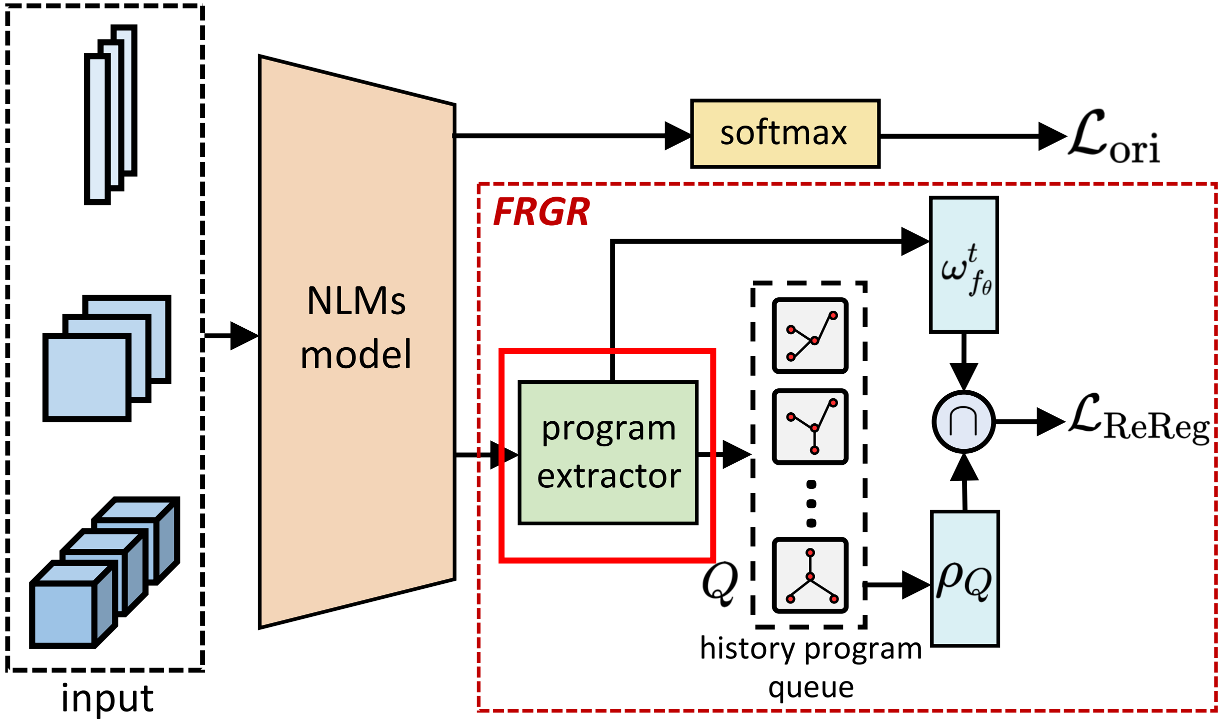
Methodology – An intuitive example

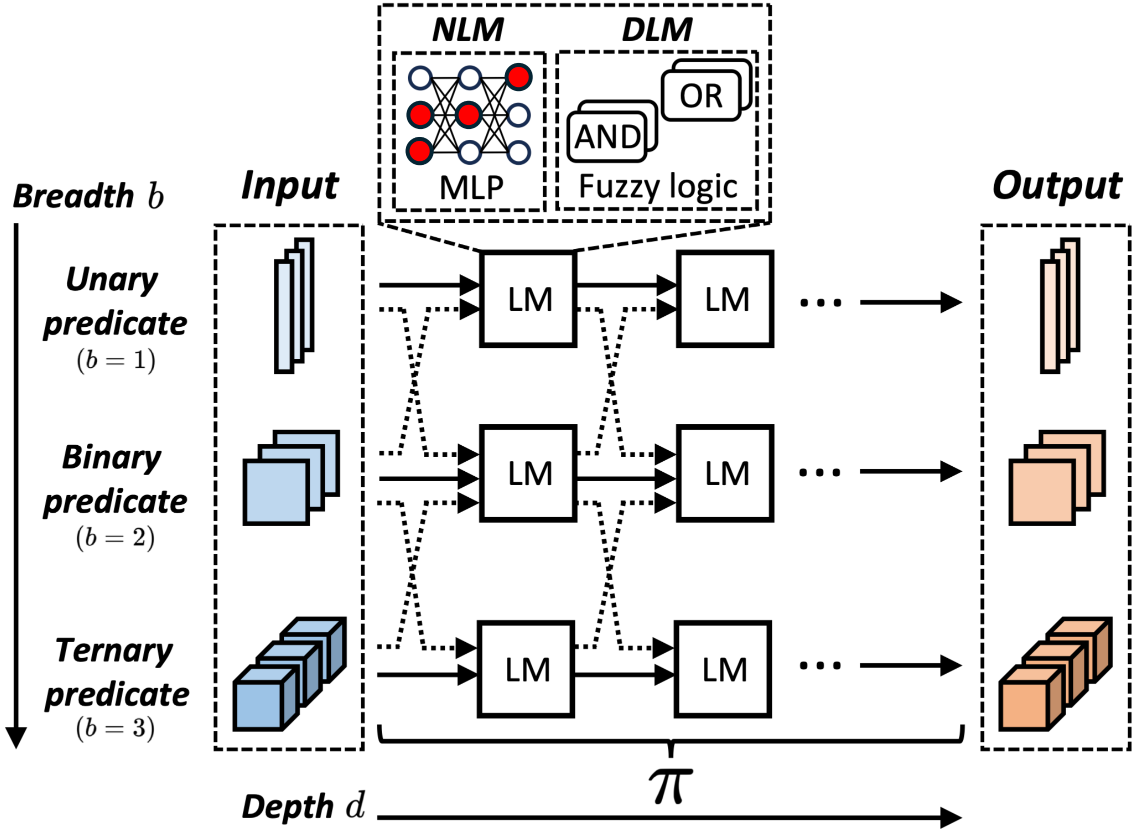


Methodology

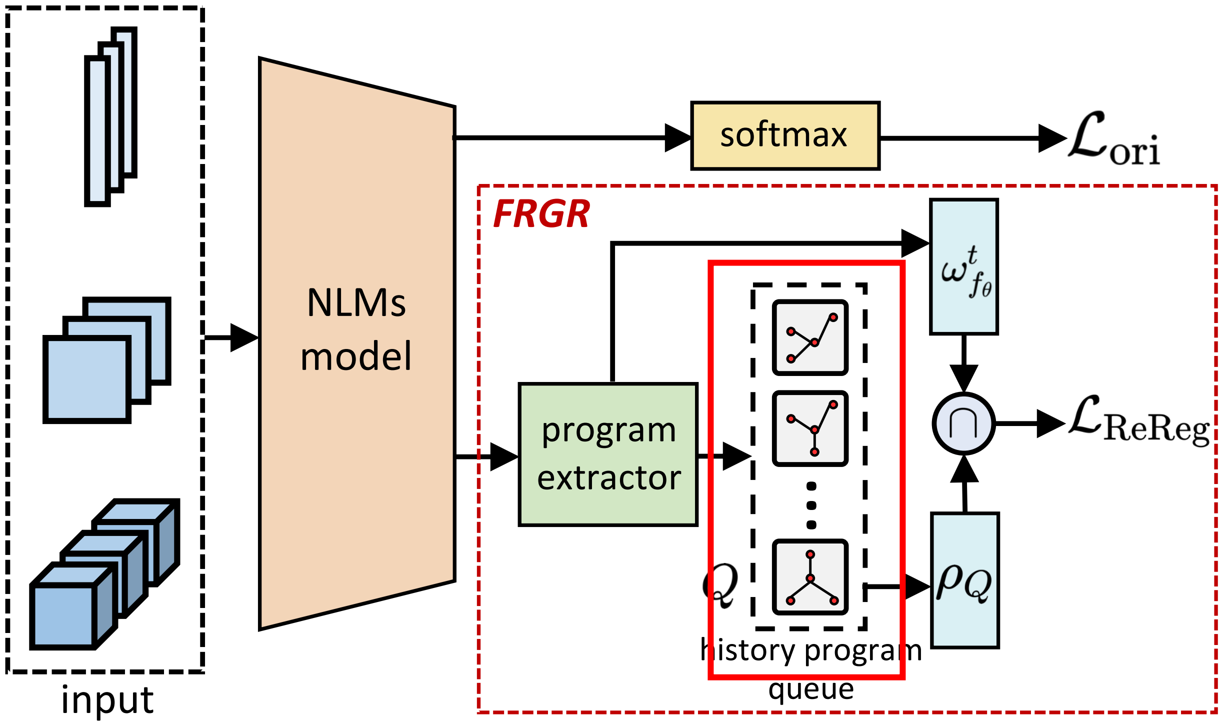


Methodology



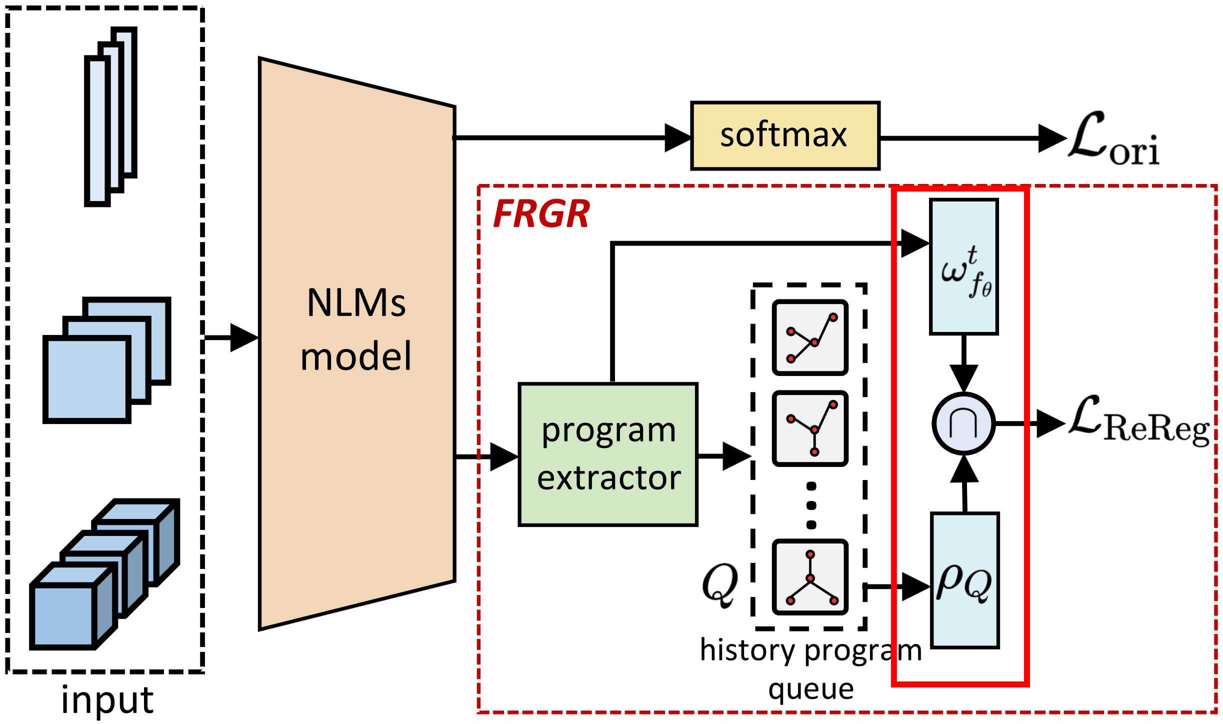


Methodology



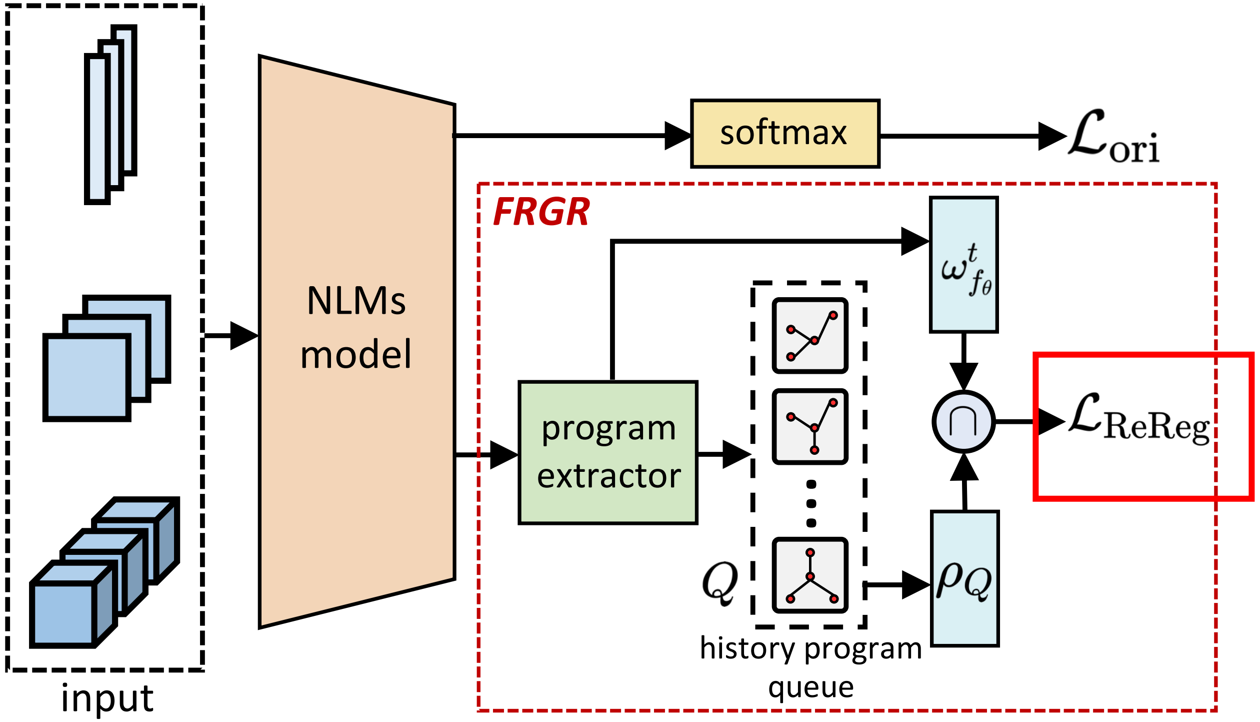
$$Q^{t+1} = \text{enqueue}(Q^t, \omega^t_{f_\theta}), \exists (I, O) \in u : f^t_\theta(I) \neq O$$

Methodology



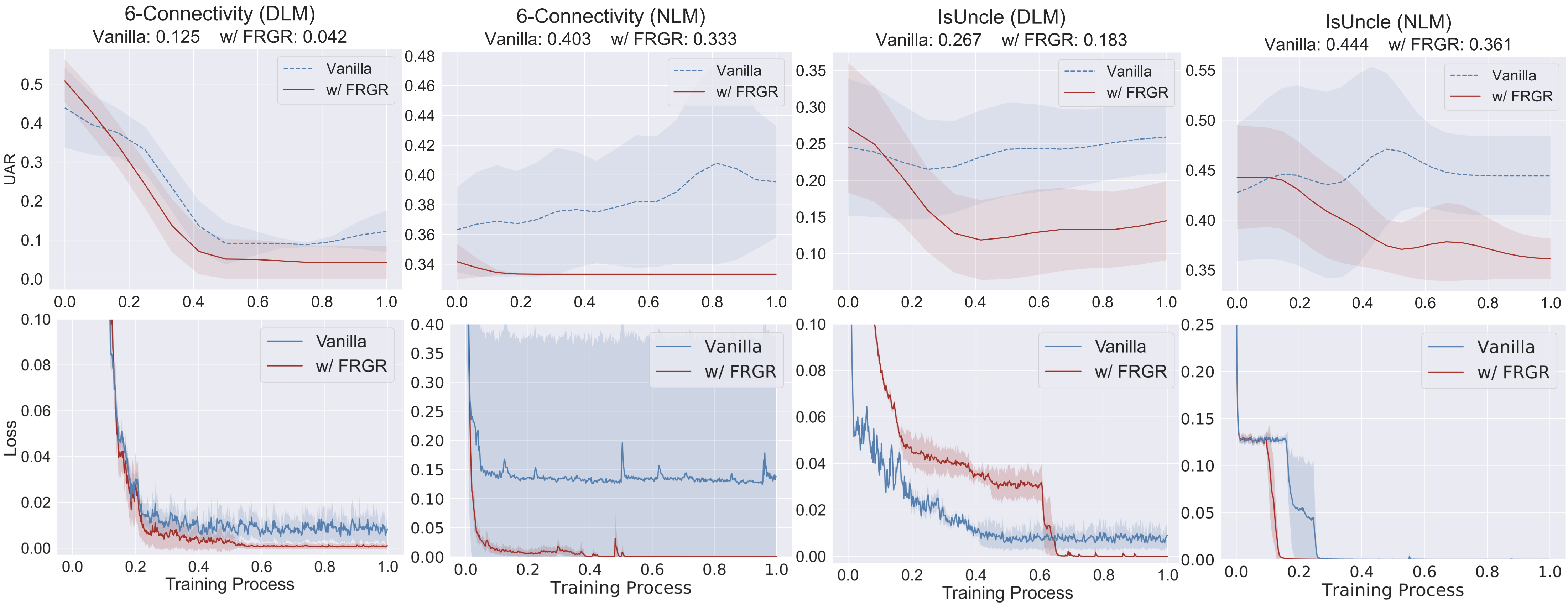
$$\rho_Q = \{J \subseteq S_{f_\theta} \mid |\{\omega^t_{f_\theta} \in Q \mid J \subseteq \omega^t_{f_\theta}\}| > \text{minsup}\}$$

Methodology



$$\mathcal{L}_{\text{ReReg}}(\theta) = \sum_{j=0}^{|\phi|} \|\phi_j\|_1, \phi = \left\{\theta_{(b,d,x,y)} : (b, d, x, y) \in \omega^t_{f_\theta} \cap \rho_Q\right\}$$

Results – Motivation Validation & Repetition Mitigation



Results – Data-Rich Setting

Task	NLM / **NLM w/ FRGR (Ours)**				DLM / **DLM w/ FRGR (Ours)**			
Family Tree	Grad-ratio (%)↑	n=20 (%)↑	n=100 (%)↑	# Epochs↓	Grad-ratio (%)↑	n=20 (%)↑	n=100 (%)↑	# Epochs↓
HasFather	100.00/100.00	100.00/100.00	100.00/100.00	5.90/6.00	100.00/100.00	100.00/100.00	100.00/100.00	22.00/23.6
HasSister	100.00/100.00	100.00/100.00	100.00/100.00	18.09/17.64	100.00/100.00	100.00/100.00	100.00/100.00	68.80/67.20
IsGrandparent	100.00/100.00	100.00/100.00	100.00/100.00	96.20/55.80	100.00/100.00	100.00/100.00	100.00/100.00	50.40/51.20
IsUncle	90.00/100.00	99.76/100.00	82.60/100.00	143.70/78.40	60.00/80.00	60.00/80.00	60.00/80.00	319.20/278.40
IsMGUncle	70.00/100.00	97.16/99.96	10.04/60.44	203.88/175.20	40.00/60.00	48.1/58.20	20.00/40.00	459.20/423.80
Graph Reasoning	Grad-ratio (%)↑	n=10 (%)↑	n=20 (%)↑	# Epochs↓	Grad-ratio (%)↑	n=10 (%)↑	n=20 (%)↑	# Epochs↓
1-OutDegree	100.00/100.00	100.00/100.00	100.00/100.00	14.30/17.00	100.00/100.00	100.00/100.00	100.00/100.00	46.20/50.00
2-OutDegree	90.00/100.00	96.52/100.00	90.80/100.00	77.9/13.40	100.00/100.00	100.00/100.00	100.00/100.00	81.60/73.60
4-Connectivity	100.00/100.00	100.00/100.00	100.00/100.00	16.80/20.50	100.00/100.00	100.00/100.00	100.00/100.00	90.40/87.40
6-Connectivity	60.00/100.00	74.40/100.00	69.20/100.00	278.00/41.60	80.00/80.00	86.90/95.40	53.28/90.10	282.40/230.80
Reinforcement Learning	Grad-ratio (%)↑	n=10 (%)↑	n=50 (%)↑	# Epochs↓	Grad-ratio (%)↑	n=10 (%)↑	n=50 (%)↑	# Epochs↓
Sorting	100.00/100.00	100.00/100.00	100.00/100.00	24.00/22.20	-	-	-	-
Path	50.00/60.00	99.55/100.00	99.95/100.00	311.00/305.20	-	-	-	-
BlocksWorld	40.00/60.00	97.11/96.59	76.89/83.90	390.11/386.67	-	-	-	-

Results – Data-Scarce Setting

Task	NLM / **NLM w/ FRGR (Ours)**				DLM / **DLM w/ FRGR (Ours)**			
Family Tree	Grad-ratio (%)↑	n=20 (%)↑	n=100 (%)↑	# Epochs↓	Grad-ratio (%)↑	n=20 (%)↑	n=100 (%)↑	# Epochs↓
HasFather	100.00/100.00	100.00/100.00	100.00/100.00	5.50/ 5.50	100.00/100.00	100.00/100.00	100.00/100.00	23.20/27.00
HasSister	100.00/100.00	100.00/100.00	100.00/100.00	13.20/13.30	100.00/100.00	100.00/100.00	100.00/100.00	67.20/68.00
IsGrandparent	90.00/100.00	68.70/77.51	63.40/72.26	127.90/54.30	100.00/100.00	100.00/100.00	100.00/100.00	57.14/56.00
IsUncle	100.00/100.00	96.49/98.05	62.53/81.50	134.30/102.80	40.00/80.00	40.20/80.00	40.00/80.00	401.60/362.80
IsMGUncle	70.00/100.00	68.52/94.20	33.00/48.00	356.30/251.60	0.00/0.00	0.00/0.00	0.00/0.00	500.00/500.00
Graph Reasoning	Grad-ratio (%)↑	n=10 (%)↑	n=20 (%)↑	# Epochs↓	Grad-ratio (%)↑	n=10 (%)↑	n=20 (%)↑	# Epochs↓
1-OutDegree	100.00/100.00	100.00/100.00	100.00/100.00	52.90/61.10	100.00/100.00	100.00/100.00	100.00/100.00	47.20/48.00
2-OutDegree	90.00/100.00	99.92/100.00	99.72/100.00	135.70/73.30	100.00/100.00	100.00/100.00	100.00/100.00	92.00/83.62
4-Connectivity	100.00/100.00	100.00/100.00	100.00/100.00	151.60/195.10	100.00/100.00	100.00/100.00	100.00/100.00	82.80/68.00
6-Connectivity	80.00/80.00	63.20/77.20	39.40/70.80	63.75/38.25	20.00/40.00	75.30/86.30	59.80/70.00	424.00/359.20
Reinforcement Learning	Grad-ratio (%)↑	n=10 (%)↑	n=50 (%)↑	# Epochs↓	Grad-ratio (%)↑	n=10 (%)↑	n=50 (%)↑	# Epochs↓
Sorting	100.00/100.00	100.00/100.00	100.00/100.00	28.20/24.60	-	-	-	-
Path	70.00/100.00	98.94/99.88	93.65/99.80	304.60/206.00	-	-	-	-
BlocksWorld	40.00/40.00	84.13/90.13	45.93/52.60	414.00/442.00	-	-	-	-

Conclusion

- We propose a novel regularization framework called FRGR, which improves the optimization of the NLMs models by utilizing the root cause of error repetition.
- Our proposed method first summarizes the root cause of errors from the models' previous behavior with pattern mining techniques.
- FRGR penalizes the model if it repeats similar mistakes in future training iterations.
- Experimental results on multiple reasoning benchmarks demonstrate that FRGR can effectively improve the NLMs' performance, generalization, training efficiency, and data efficiency.