Goal-Aware Neural SAT Solver

Emīls Ozoliņš, Kārlis Freivalds, Andis Draguns, Elīza Gaile, Ronalds Zakovskis, Sergejs Kozlovičs

IJCNN 2022

Boolean Satisfiability (SAT)

NP-complete

Used in Circuit Design, Planning and Scheduling, Model Checking

Often solved by search

Can neural solvers be faster?

$$(x1 \lor \neg x2) \land (\neg x1 \lor x2 \lor x3) \land \neg x1$$

$$x1 = F \ \land \ x2 = F \land X3 = F\,?$$

$$\vdots$$

$$x1 = T \ \land \ x2 = T \land X3 = T\,?$$

Contributions

Query mechanism for neural networks

Unsupervised training for Boolean Satisfiability (SAT)

QuerySAT – the SOTA fully neural SAT solver

12*56 = ?

12*56 > 4? Yes

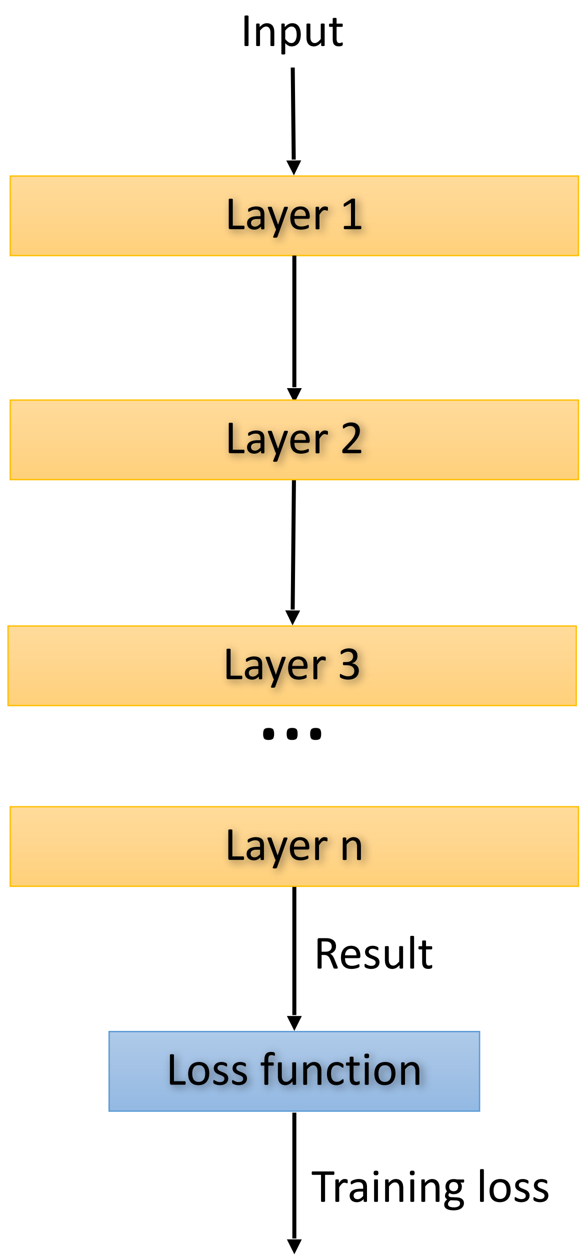
12*56 < 100? No

…

12*56 = 672

Neural Query

Deep Neural Network

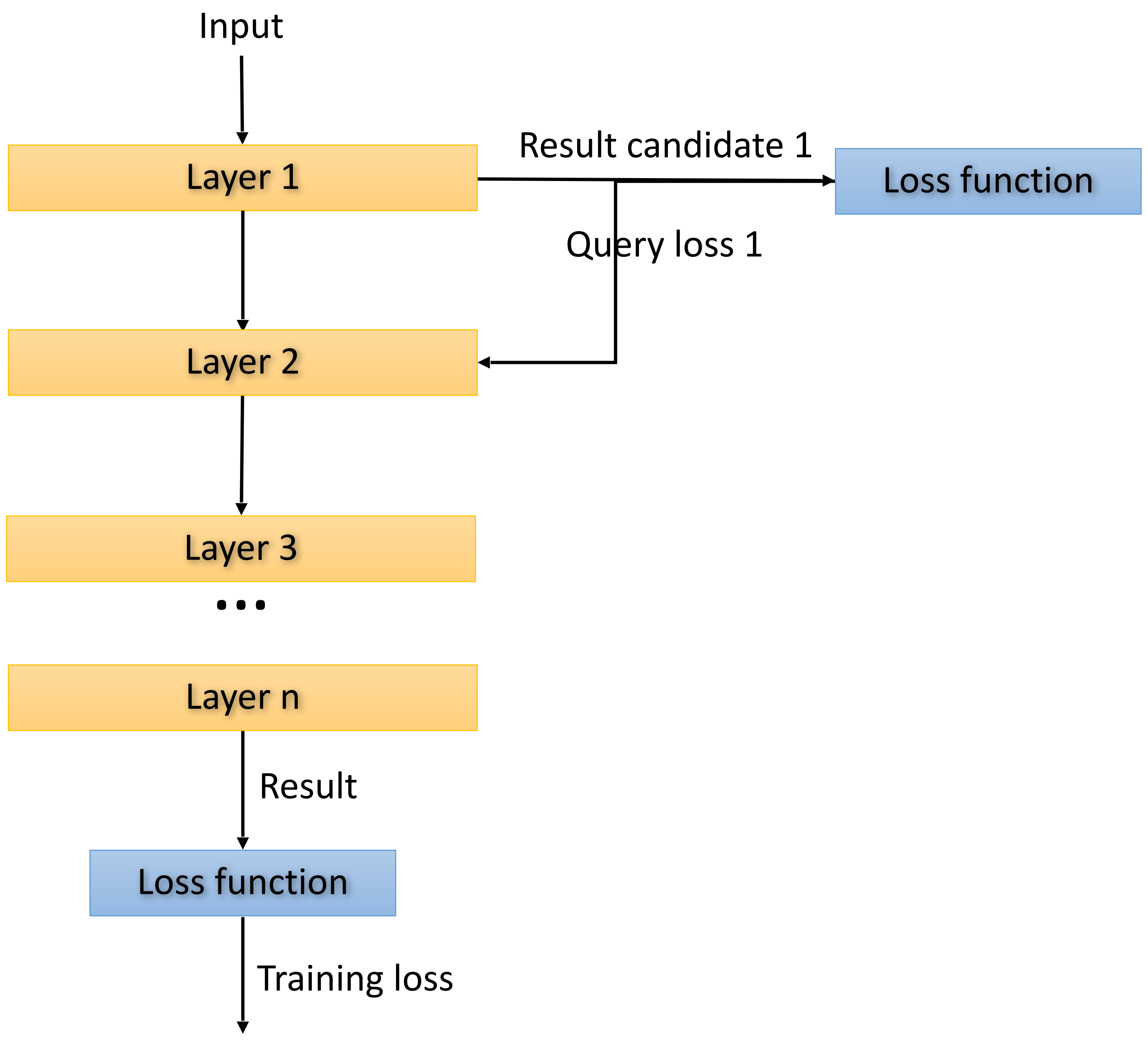


Neural Query

Produce a candidate result at an intermediate step

Check its correctness using the loss function

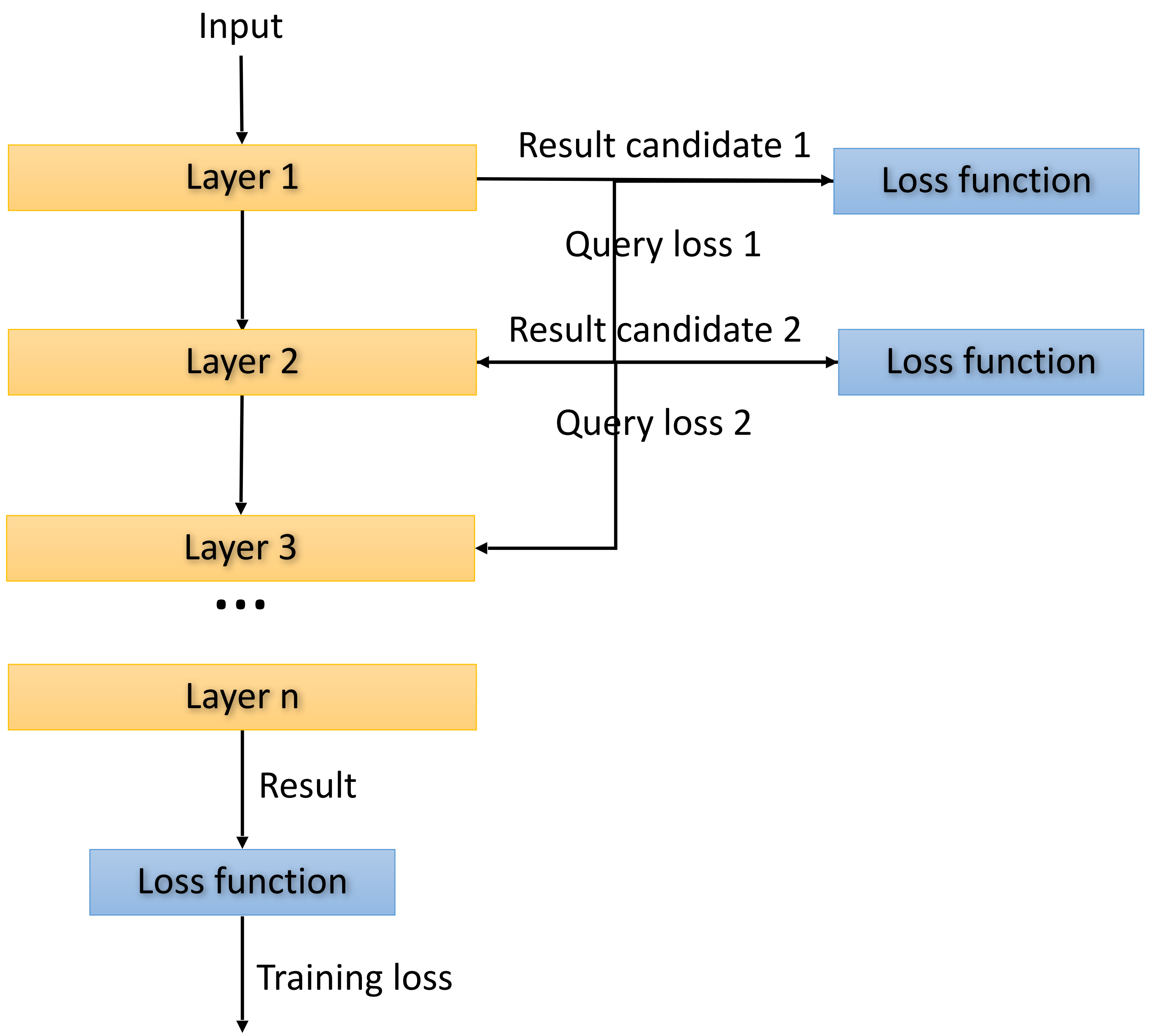
Pass the result to the next step



Neural Query

Do queries many times

The final result should be verified and correct…

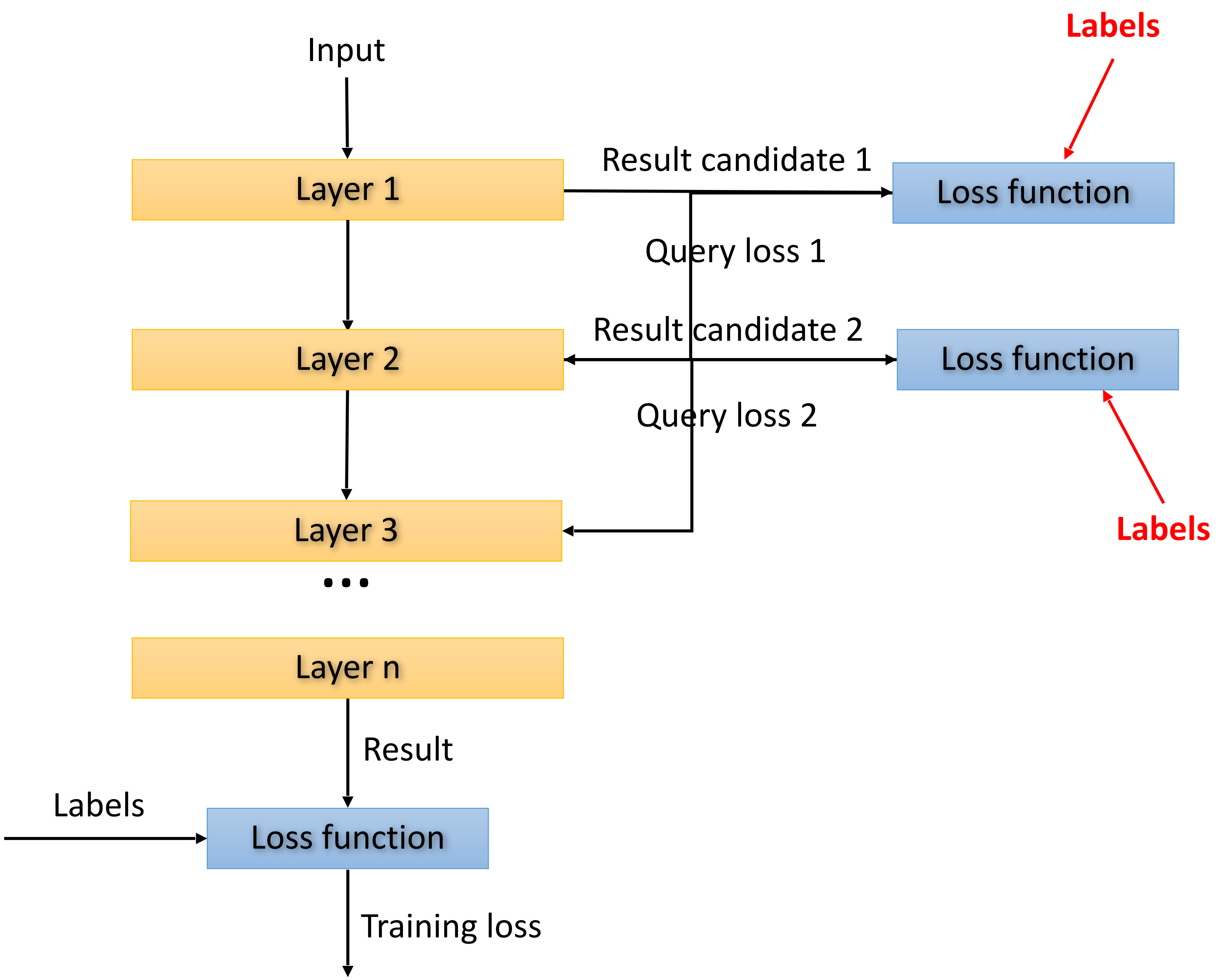


Neural Query

Problem at the inference!

The loss function uses labels

Solution: use unsupervised loss not requiring labels

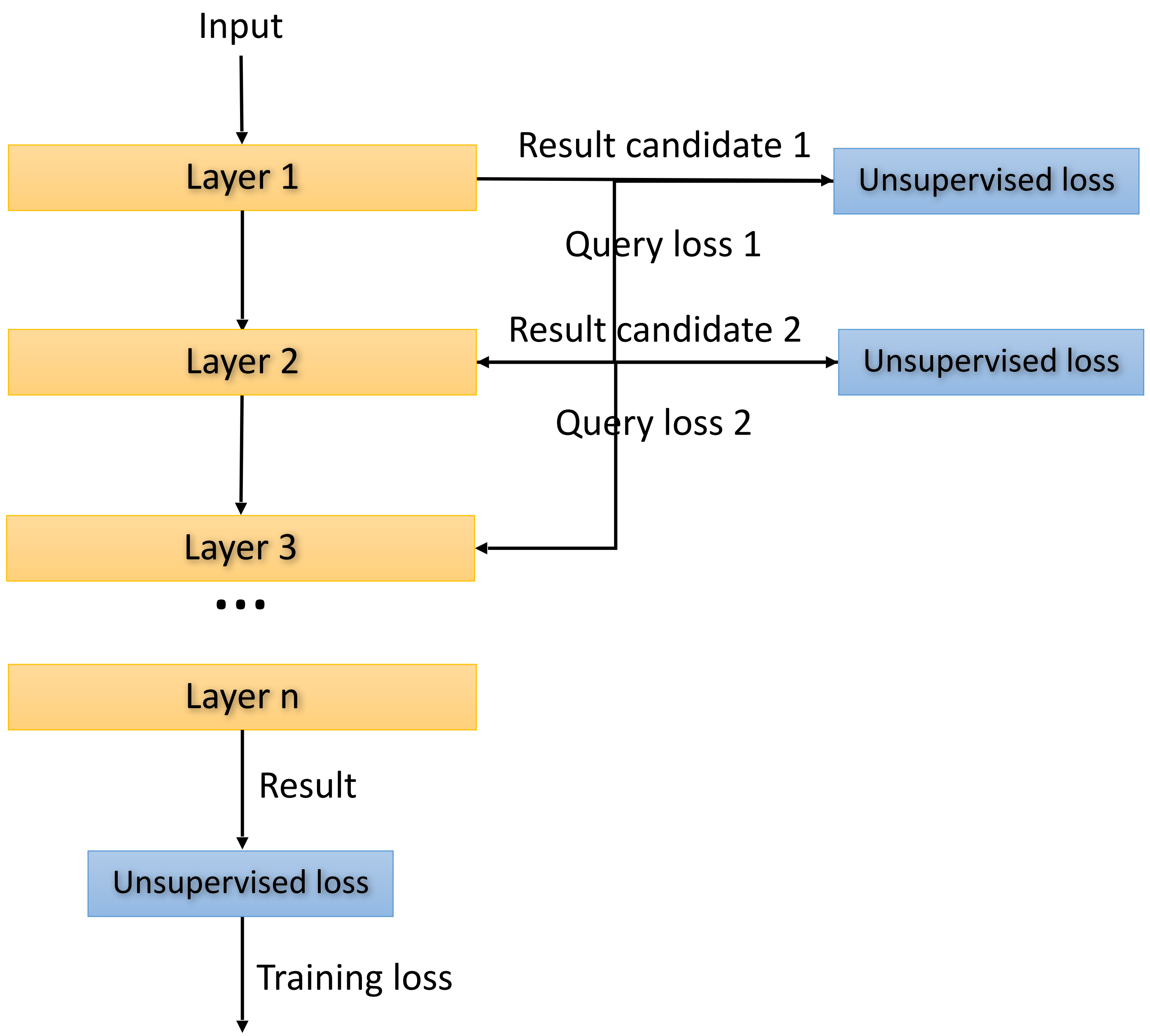


Neural Query

Unsupervised loss checks the solution correctness without using labels

Essential for queries

Allows training tasks with multiple solutions



Unsupervised Loss

Evaluates if the assignment is satisfiable

Variables relaxed to [0..1]

$$V_c(x) = 1 - \prod_{i \in c^+} (1 - x_i) \prod_{i \in c^-} x_i$$

$$\mathcal{L}_\phi(x) = \prod_{c \in \phi} V_c(x)$$

$$\mathcal{L}_\phi^{\log}(x) = -\log(\mathcal{L}_\phi(x)) = -\sum_{c \in \phi} \log(V_c(x))$$

Theorem 3.1. *For a binary query point x the losses $\mathcal{L}_\phi(x)$ or $V_c(x)$ are equal to 1 if the formula ϕ or clause c is satisfied and 0 otherwise.*

Theorem 3.2. *A single query that returns the loss V_c for each clause c is sufficient to uniquely identify the SAT formula ϕ to be solved.*

Factor-graph

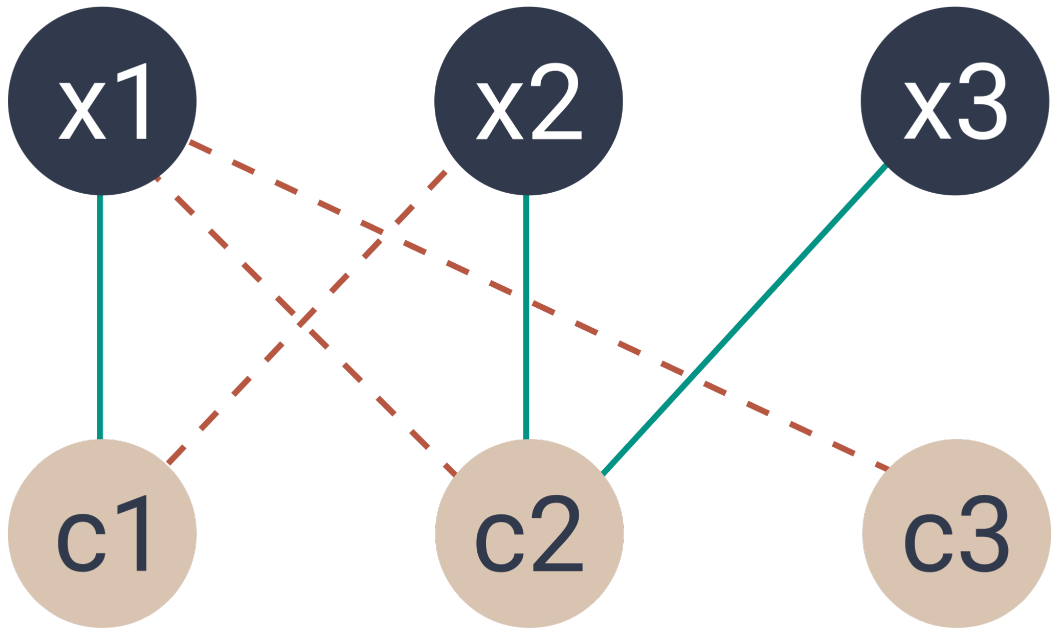
Conjunctive Normal Form (CNF)

Node for each variable and each clause

Edge represents that the variable is present in the clause

Different types of edges for positive and negative occurrence

$$(x1 \lor \neg x2) \land (\neg x1 \lor x2 \lor x3) \land \neg x1$$

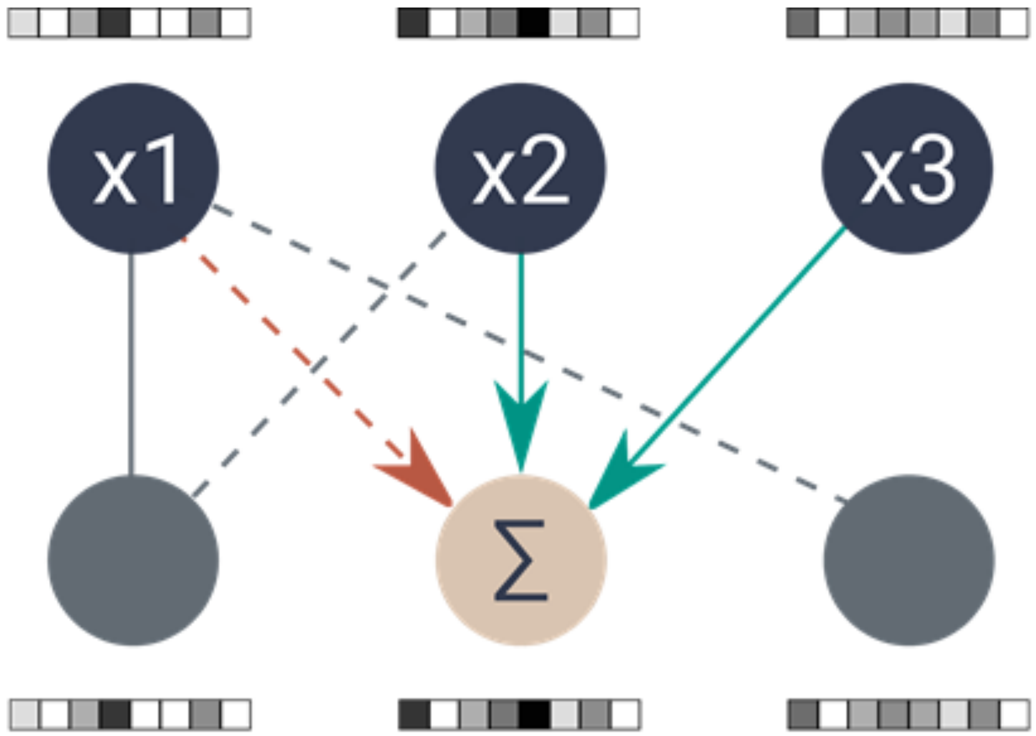


Graph Neural Network

Employs message passing

Feature vector is attached to each node

Features are updated by aggregating messages from graph neighbors



QuerySat

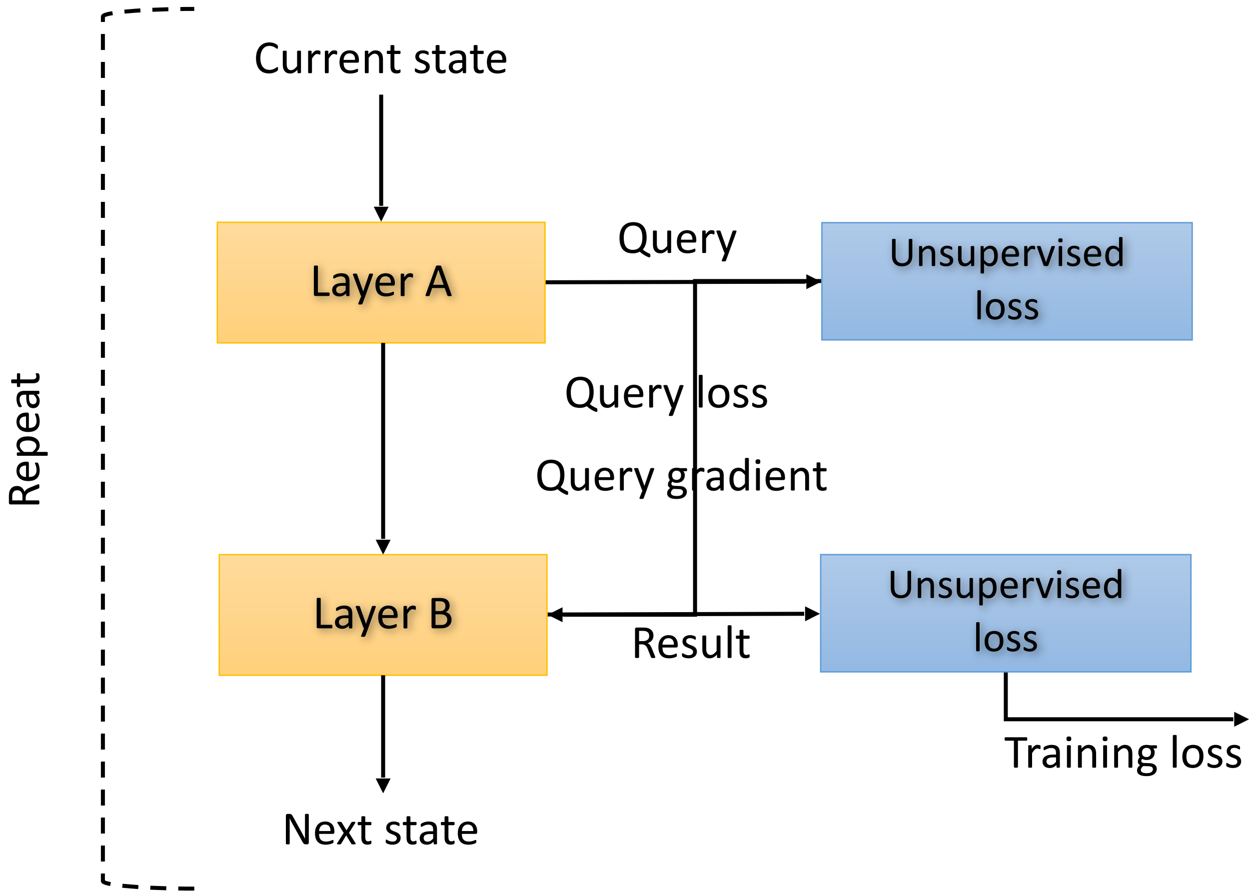
Recurrent

Training loss at each step

Also uses query gradient

Implemented as a GNN:

- Queries pass information from variables to clauses
- Use message passing from clauses to variables



Datasets

Random k-SAT

Random 3-SAT

3-Clique

k-Coloring

SHA-1 preimage attack

Methodology

Train with 32 recurrent steps, test up to 4096

Test on larger formulas than used for training

Compare with previously best: NeuroCore

Query helps

Tested with 4096 recurrent steps

	k-SAT	3-Clique
NeuroCore	60.97 ± 6.19	70.39 ± 2.68
+ *Query*	67.21 ± 13.54	84.55 ± 2.93
+ *Query* + *G*	$\mathbf{75.50 \pm 7.15}$	$\mathbf{95.50 \pm 1.95}$

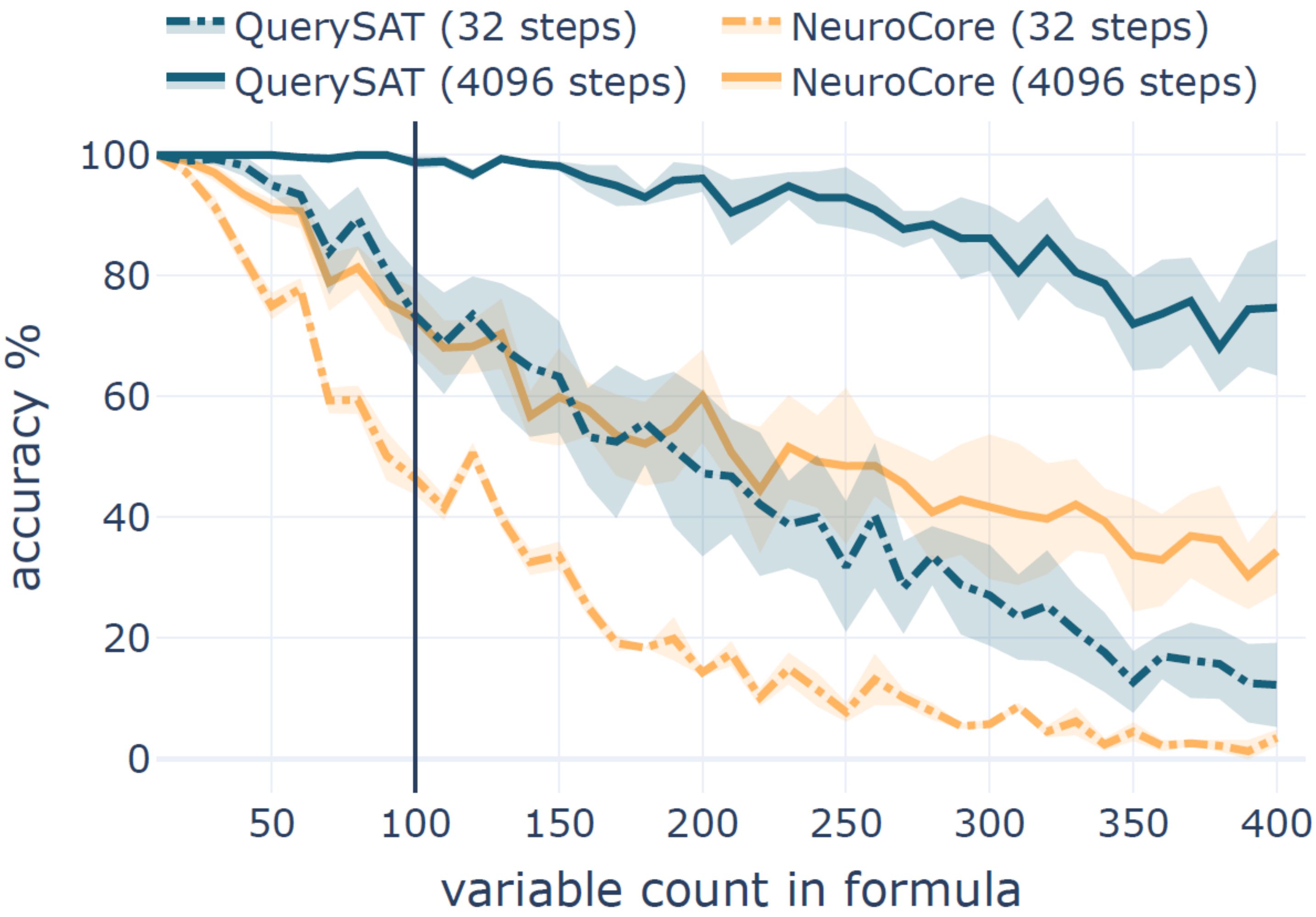
QuerySAT results

Task	QuerySAT			NeuroCore		
	$s_{test} = 32$	$s_{test} = 512$	$s_{test} = 4096$	$s_{test} = 32$	$s_{test} = 512$	$s_{test} = 4096$
k-SAT	72.12 ± 0.19	96.61 ± 0.78	$\mathbf{99.05} \pm 0.38$	21.64 ± 0.27	46.85 ± 5.02	50.82 ± 6.41
3-SAT	61.89 ± 5.19	88.20 ± 4.01	$\mathbf{93.32} \pm 3.21$	28.38 ± 3.24	53.49 ± 3.94	57.63 ± 4.38
3-Clique	82.00 ± 4.73	93.06 ± 4.67	$\mathbf{94.74} \pm 4.62$	1.03 ± 0.69	1.03 ± 0.66	1.04 ± 0.66
k-Coloring	91.70 ± 1.01	97.76 ± 0.98	$\mathbf{98.32} \pm 0.82$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
SHA-1	33.25 ± 4.17	$\mathbf{46.57} \pm 1.16$	46.45 ± 1.10	0.00 ± 0.0	0.27 ± 0.09	0.24 ± 0.09

Tested on larger instances than being trained on

Generalizes to larger instances

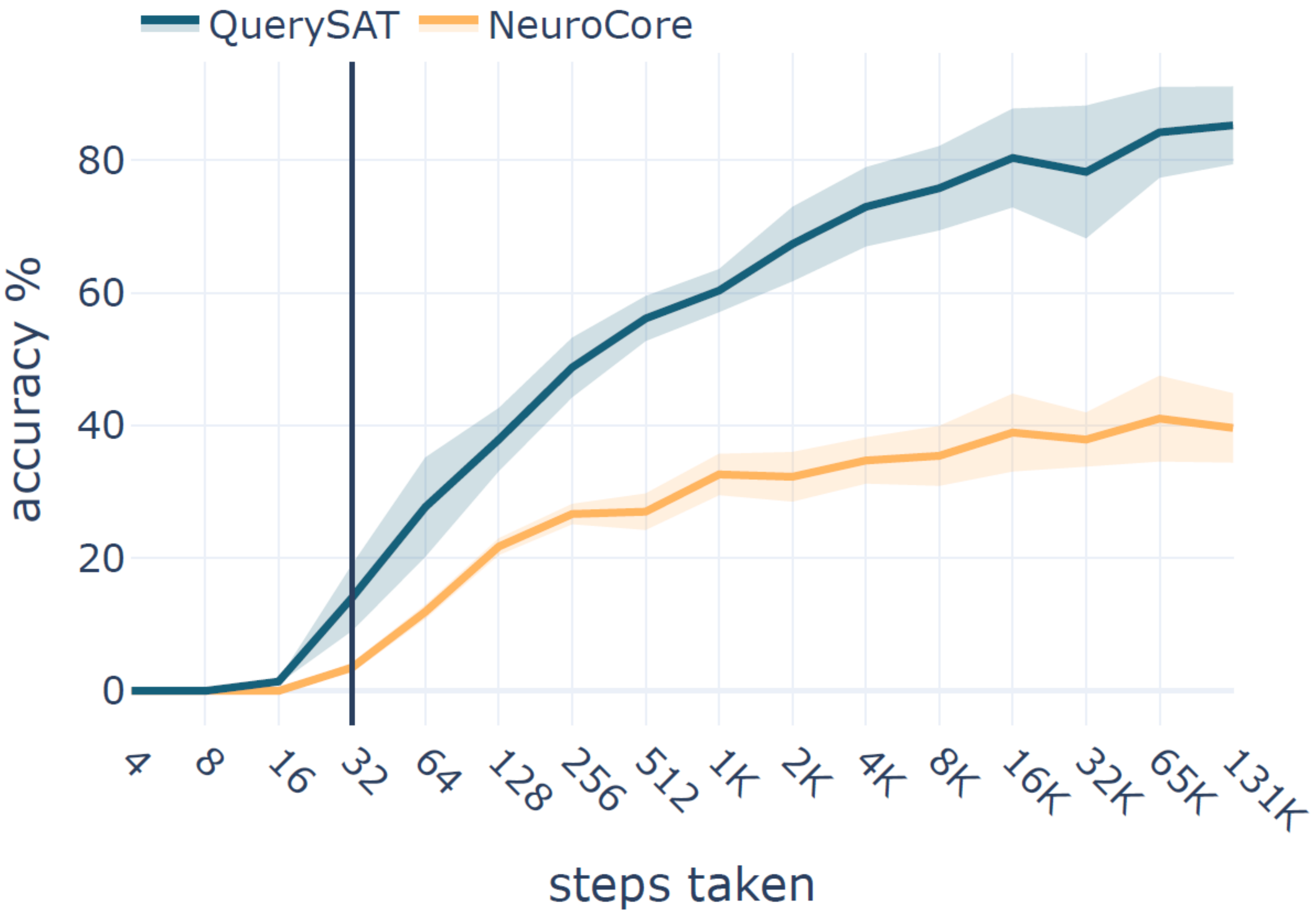
3-sat, trained up to 100 variables



More recurrent steps is better

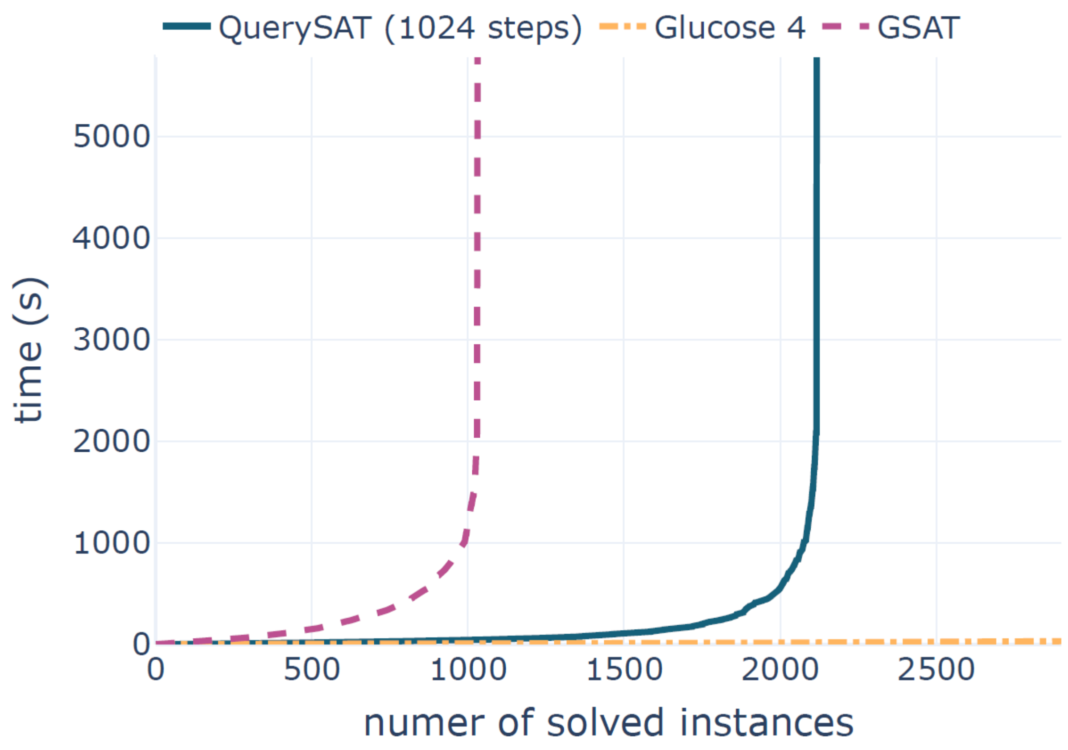
3-sat, trained for 32 steps

Trained on 100 variables,
tested on 400 variables

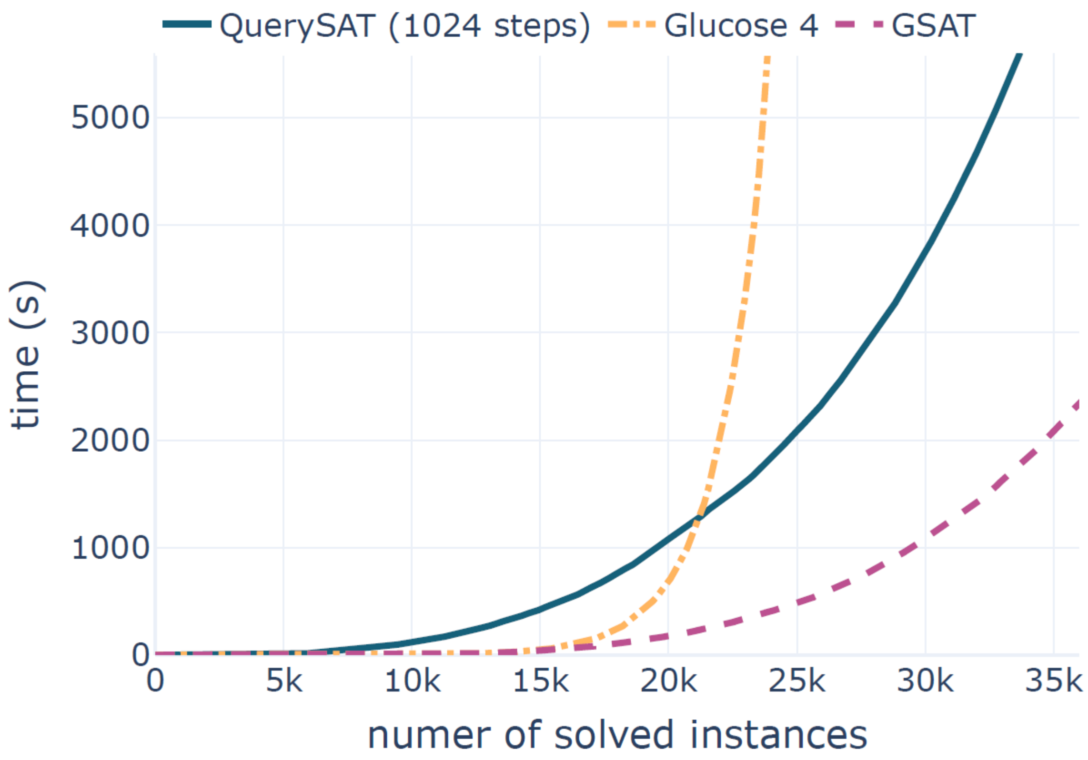


Neural vs. Classical

(towards bottom-right is better)



SHA-1 preimage



Random 3-SAT

Conclusions

QuerySAT: a fully neural SAT solver

Queries are useful, replace message passing

Unsupervised loss trains to find solutions without knowing them

Thank you!

Goal-Aware Neural SAT Solver

https://github.com/LUMII-Syslab/QuerySAT

https://arxiv.org/abs/2106.07162