The Isabelle ENIGMA

Zarathustra Goertzel, Jan Jakubův, Cezary Kaliszyk,
Miroslav Olšák, Jelle Piepenbrock, and Josef Urban

Czech Technical University in Prague
University of Innsbruck, Austria
Institut des Hautes Études Scientifiques
Radboud University

ITP 2022

Outline 1/2

- Related work on Mizar:
  - ENIGMA
  - Parental Guidance
  - Graph Neural Network
- The Isabelle Dataset
- Strategy Specialization for Isabelle

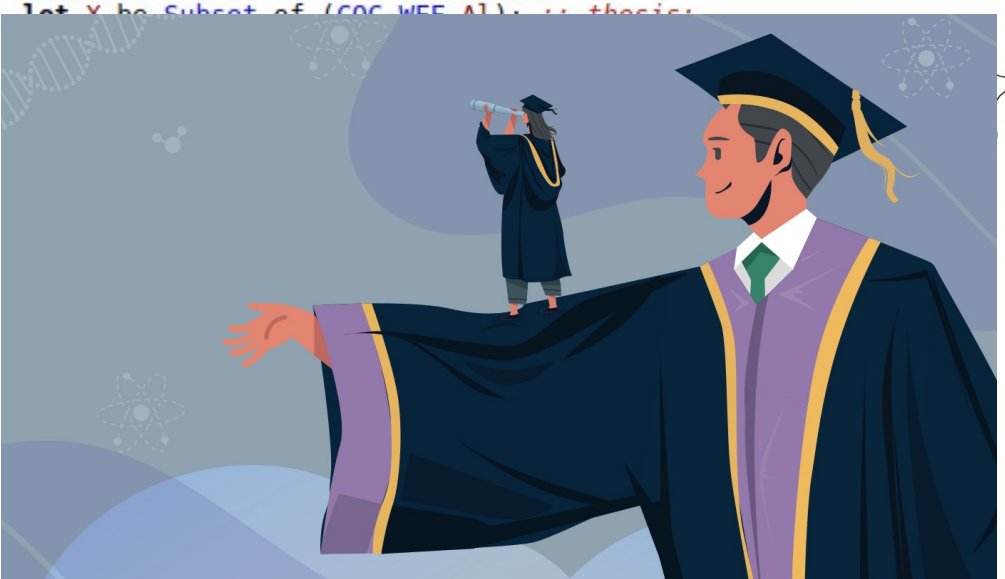
Outline 2/2

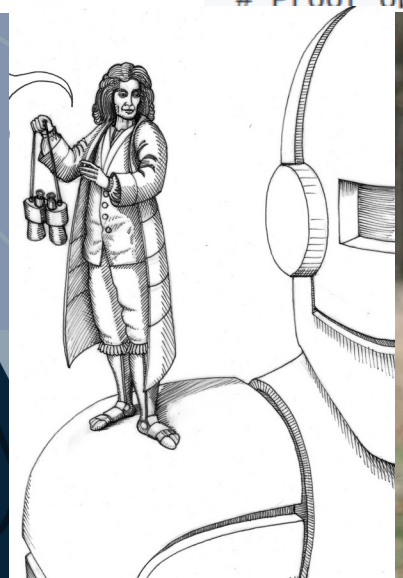
- ENIGMA and GNN model overviews
- ENIGMA training paradigms:
  - Looping, greedy covers, grid search, etc.
- Experimental Evaluations
- Conclusion

Related Work over Mizar

```
theorem Th6: :: HENMODEL:6
  for Al being QC-alphabet
  for X being Subset of (CQC-WFF Al) holds
  ( X is Inconsistent iff for p being Element of CQC-WFF Al holds X |- p )
proof
  let Al be QC-alphabet ; :: thesis:
```

```
# Proof object clause steps            : 76
# Proof object initial clauses used    : 35
# Proof object initial formulas used   : 21
# Proof object simplifying inferences  : 52
```





Enigma over Mizar in 2021

- Mizar Math Library (MML): 57880 toplevel/MPTP1148 problems
- **ENIGMA**: many faster/slower ML methods for clause selection
- 3-phase ENIGMA: 2 fast GBDT models + 1 slow GNN model
  - 60% stronger than E auto-schedule on 2896-big holdout set
  - 56% in 30s (CPU+GPU) by a single strategy (1632/2896)
  - 17.4% stronger than 1 fast ML model (1632/1390)

See https://github.com/ai4reason/ATP_Proofs

Enigma over Mizar in 2021

- Mizar Math Library (MML): 57880 toplevel/MPTP1148 problems
- **ENIGMA**: many faster/slower ML methods for clause selection
- 3-phase ENIGMA: 2 fast GBDT models + 1 slow GNN model
  - 60% stronger than E auto-schedule on 2896-big holdout set
  - 56% in 30s (CPU+GPU) by a single strategy (1632/2896)
  - 17.4% stronger than 1 fast ML model (1632/1390)
- With Vampire/Deepire (M. Suda) and many strategies/times:

  - 75.5% (43717/57880) of MML by September 2021 (bushy)

  - 58.4% on the holdout set (420s, chainy - *hammering*)

See https://github.com/ai4reason/ATP_Proofs

Enigma over Mizar in 2021

- Mizar Math Library (MML): 57880 toplevel/MPTP1148 problems
- **ENIGMA**: many faster/slower ML methods for clause selection
- 3-phase ENIGMA: 2 fast GBDT models + 1 slow GNN model
  - 60% stronger than E auto-schedule on 2896-big holdout set
  - 56% in 30s (CPU+GPU) by a single strategy (1632/2896)
  - 17.4% stronger than 1 fast ML model (1632/1390)
- With Vampire/Deepire (M. Suda) and many strategies/times:
  - 75.5% (43717/57880) of MML by September 2021 (bushy)
  - 58.4% on the holdout set (420s, chainy - *hammering*)
- **“So, does this work on other formal math libraries?”**

See https://github.com/ai4reason/ATP_Proofs

Isabelle/HOL

- Our dataset:
  - 179 Isabelle/Mirabelle sessions
    - 80 from the AFP (Archive of Formal Proofs)
    - 75 from Isabelle 2021-1
    - 24 from IsaFoR (Isabelle Formalization of Rewriting)
  - 1902 theory files
  - 276,363 problems
    - All goals from the sessions: typically not toplevel
    - 248k for training and 13.8k for devel and holdout

Isabelle/HOL

HOL-Nonstandard-Analysis	1699	Groebner-Macaulay	4227
Category2	1776	HOL-ODE-Numerics	4422
Poincare-Bendixson	1983	HOL-MicroJava	5183
HOL-Number-Theory	2071	HOL-Auth	5304
MonoidalCategory	2238	HOL-Complex-Analysis	5489
HOL-Cardinals	2268	Groebner-Bases	5710
Core-DOM	2280	HOL-Computational-Algebra	6280
HOL-IMP	2324	Jordan-Normal-Form	6786
HOL-Data-Structures	2353	Category3	6818
Dirichlet-Series	2435	HOL-Probability	6954
Slicing	2517	HOL-Decision-Procs	7103
HOLCF	2524	CR	7341
Formal-SSA	2899	HOL-Bali	7804
HOL-UNITY	2938	HOL	7818
HOL-Homology	3022	Goedel-HFSet-Semanticless	8697
HOL-ex	3047	HOL-Algebra	9674
CTRS	3328	HRB-Slicing	10052
HOL-Hoare-Parallel	3733	Jinja	11520
Signature-Groebner	3762	HOL-Library	15627
Valuation	3786	Bicategory	16965
Ordinary-Differential-Equations	3885	HOL-Nominal-Examples	17145
Smith-Normal-Form	4045	Group-Ring-Module	19718
Differential-Dynamic-Logic	4158	HOL-Analysis	44172

Table 1 The largest included sessions and their respective problem numbers

Isabelle/HOL

- Inspired by “Seventeen Provers under the Hammer.”
- Sledgehammer: Isabelle/HOL <-> ATPs
  - Translate goals to TPTP
  - Give to ATPs
  - Reconstruct the proof in Isabelle/HOL

- We had no ENIGMA model for Isabelle
- And hadn’t used ENIGMA with TFF data yet.

Isabelle/HOL

- 276,363 problems exported via Mirabelle
- MePo selects 512 premises
- -> FOF: use the “g??” encoding
  - Preserves polymorphism with type guards
- -> TFF: use the monomorphic encoding

Isabelle/HOL

- 276,363 problems exported via Mirabelle
- MePo selects 512 premises
- -> FOF: use the “g??” encoding
  - Preserves polymorphism with type guards
- -> TFF: use the monomorphic encoding
- -> THF: *ongoing work*

Isabelle/HOL vs Mizar

- Mizar's translation uses symbol and formula names consistently.
- Isabelle problems are "more ground and less equational" than the Mizar problems.
- Mizar (top-level) problems have on average:
  - 3.5x the clauses
  - 4-5x the clauses with variables and equality

Dataset	Problems	AC	VC	EC	iProver-10s	iProver-10s ratio
Isabelle FOF	88888	10.15	4.51	2.63	83015	0.93
Mizar	113332	35.55	23.16	10.31	65679	0.58
Ratio Miz/Isa		3.50	5.14	3.92		0.62

E-Tuning

- Find good strategies for Isabelle:
  - Run 550 BliStr/Tune strategies on 500 Vampire solved problems.
  - Run the best 76 strategies on 2000 problems.
- SInE or no SInE, that is the question.

E-Tuning

- Find good strategies for Isabelle:
  - Run 550 BliStr/Tune strategies on 500 Vampire solved problems.
  - Run the best 76 strategies on 2000 problems.
- With SInE + “hypos” parameter,
  f1711 out-performs E’s auto-mode.
- F1711 also works well with ML on Mizar
- (This is the baseline strategy.)

E-Tuning

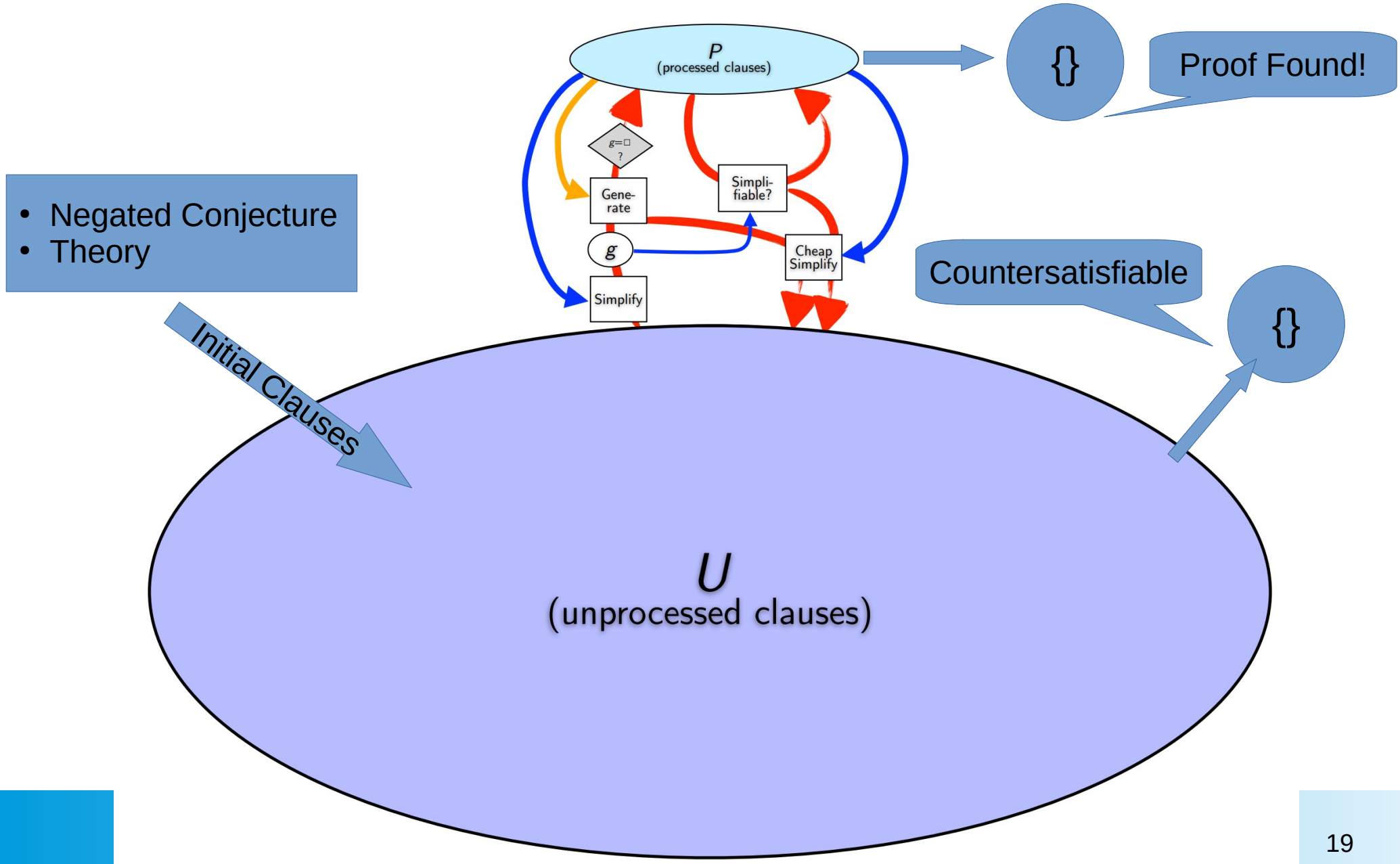
- E’s default clausification setting:
  - Add a definition when a sub-term is seen 24 times.
  - On the FOF, 10% (28k) problems timed out in 60s.
  - Because the problems are large and explosive.

E-Tuning

- E’s default clausification setting:
  - Add a definition when a sub-term is seen 24 times.
  - On the FOF, 10% (28k) problems timed out in 60s.
  - Because the problems are large and explosive.
- Setting *definitional-cnf* to 3 had no timeouts.

ATP + ML Overview

E (a Saturation-based ATP)



- Negated Conjecture
- Theory

Image partially thanks to Stephan Schulz's presentation on E

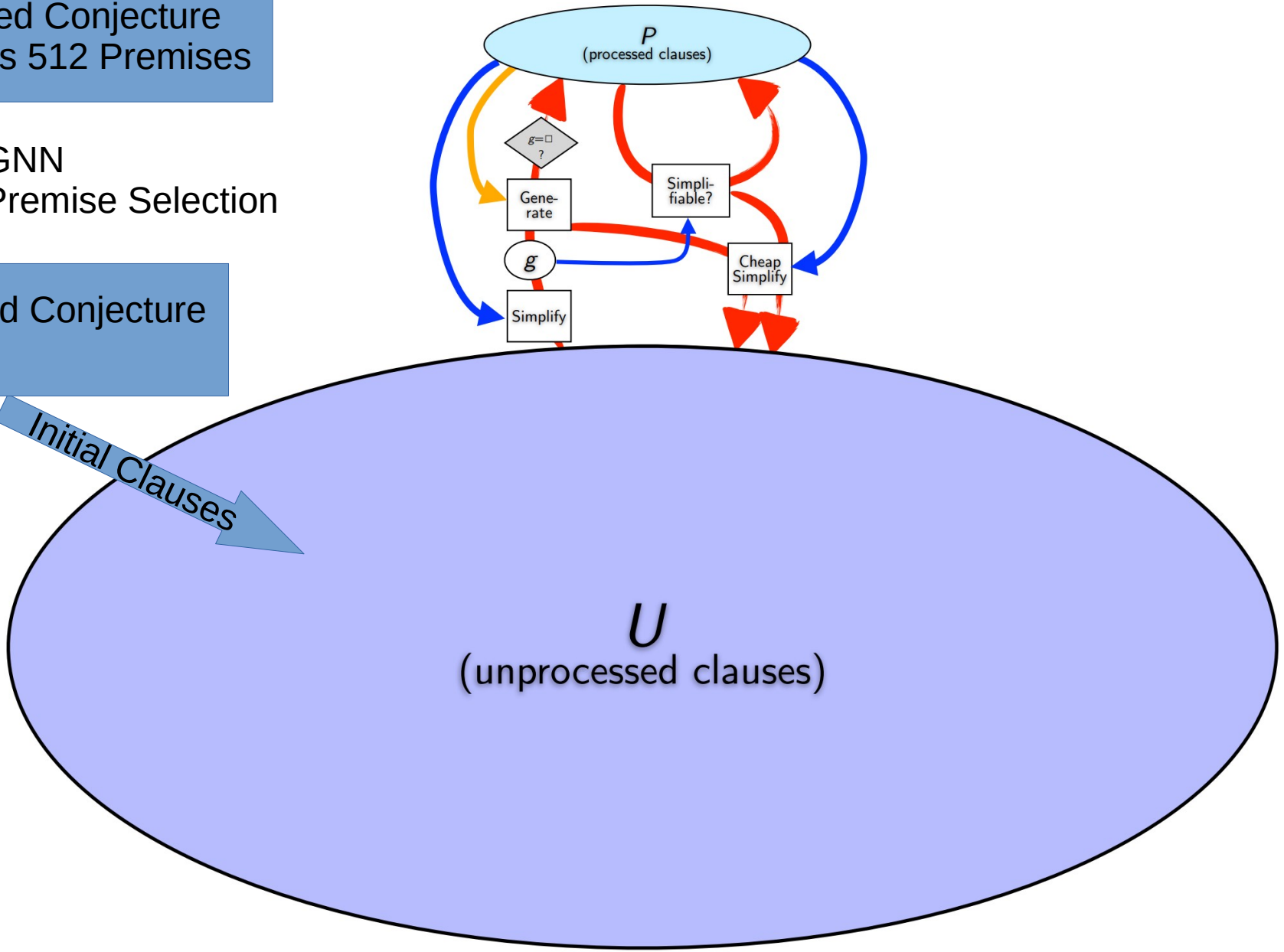
E: Premise Selection

- Negated Conjecture
- MePo's 512 Premises

GNN
Premise Selection

- Negated Conjecture
- Theory

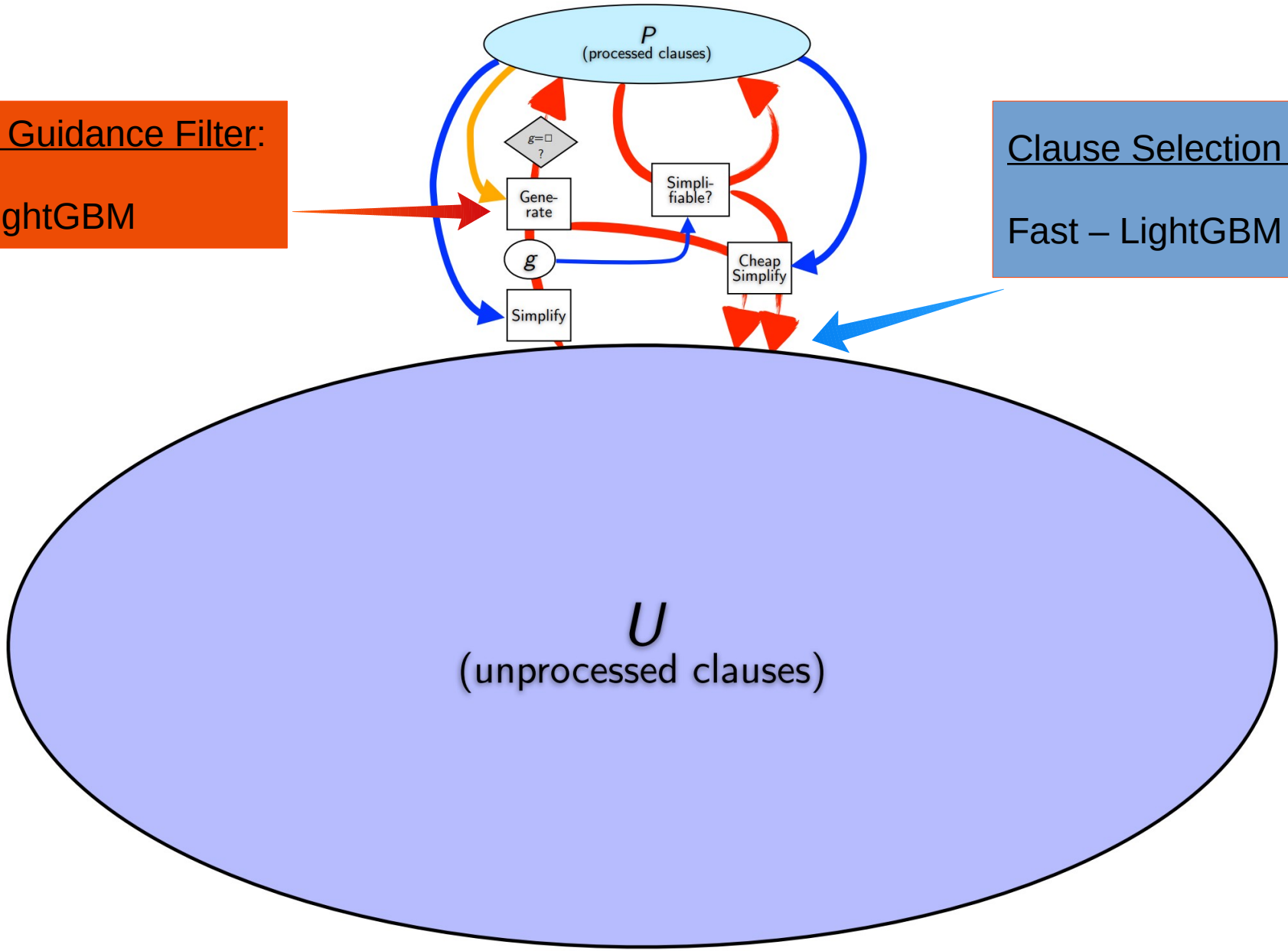
Initial Clauses



Given Clause Loop in E + ML Guidance

Parental Guidance Filter:

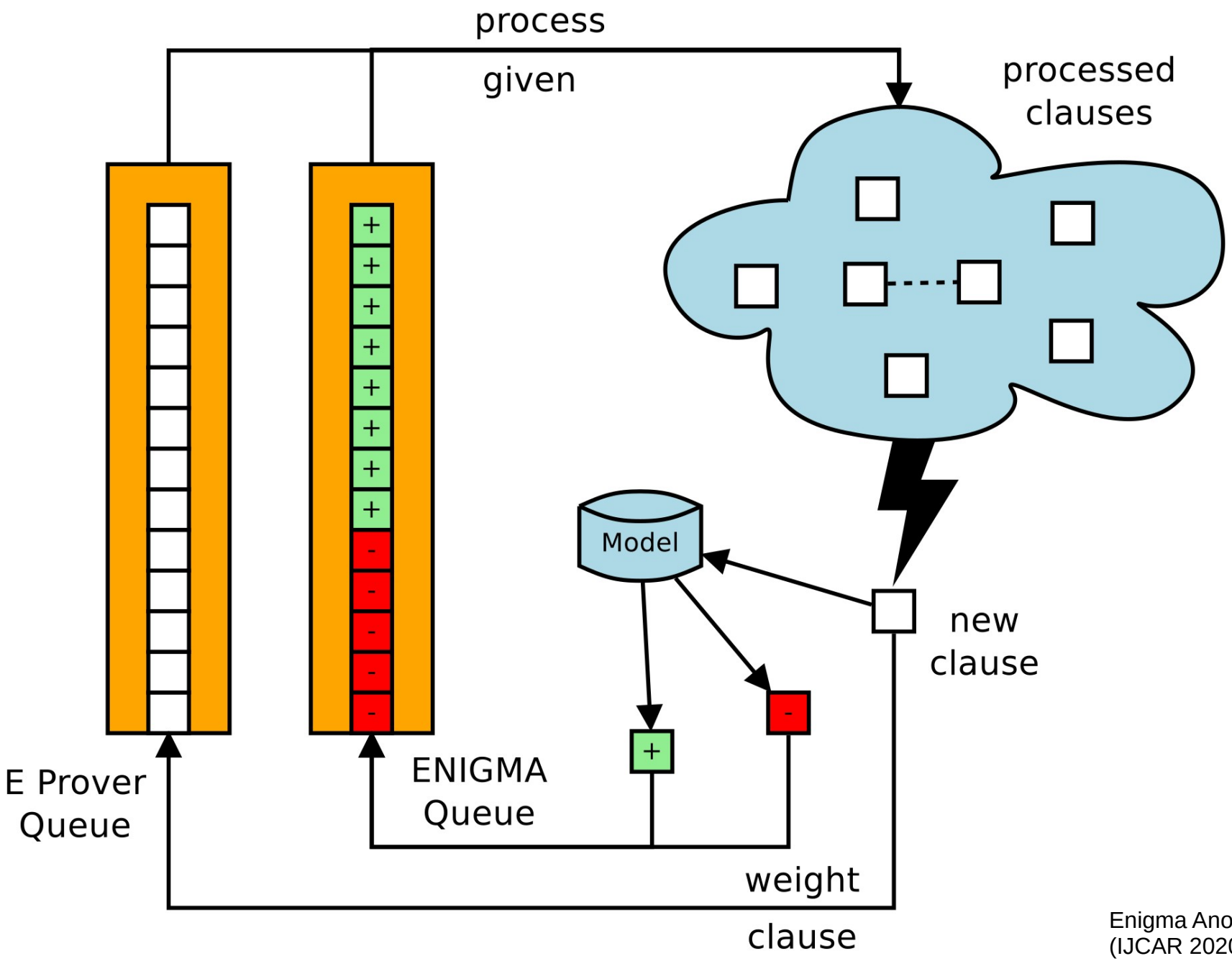
Fast – LightGBM



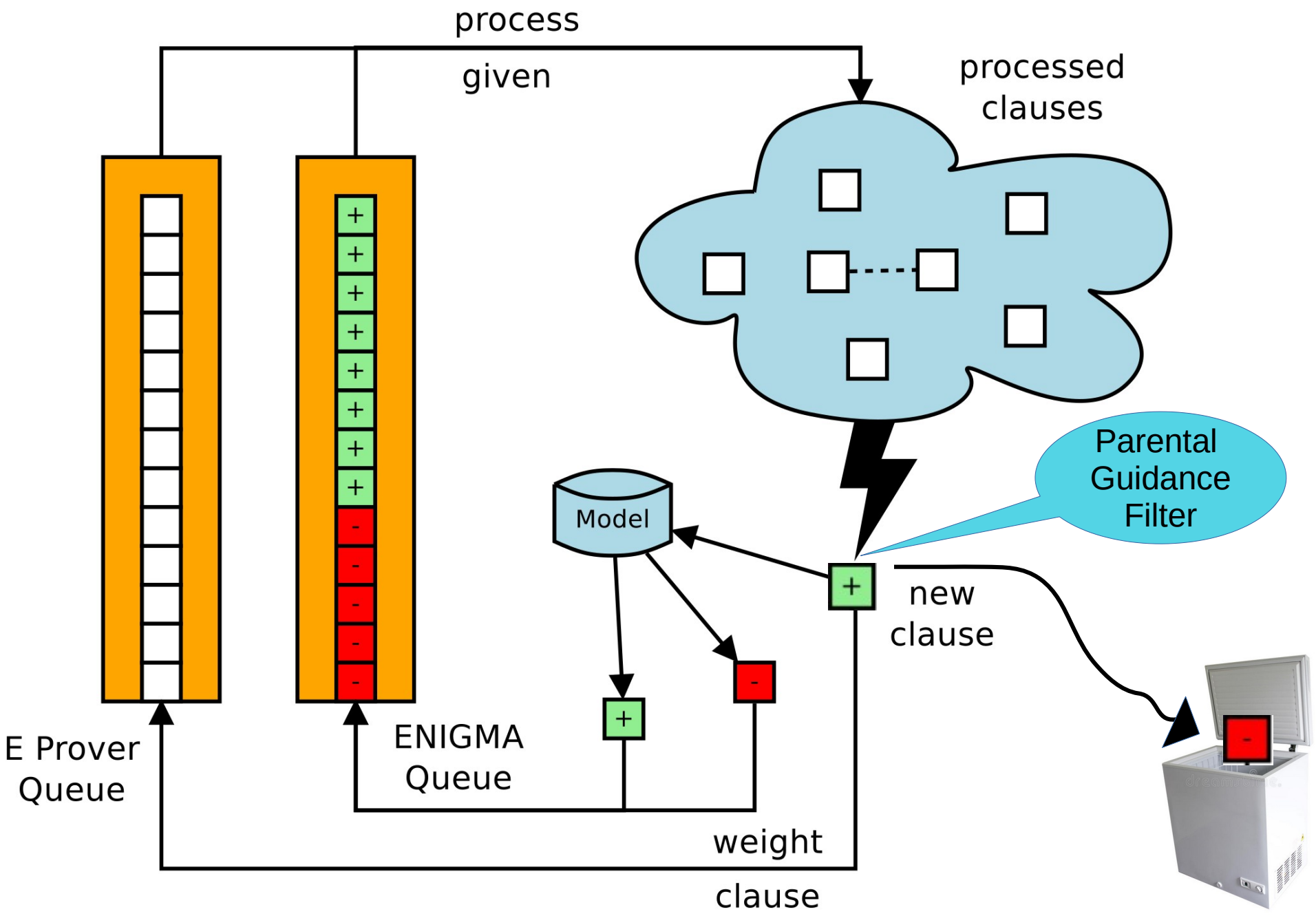
Clause Selection Model:

Fast – LightGBM

Given Clause Loop in E + ML Guidance



Given Clause Loop in E + ML Guidance



ML Model Details

ENIGMA Anonymous: Clause Selection

- Statistical machine learning for clause selection.
  - LightGBM (a gradient boosted decision tree framework)
  - (Not yet on Isabelle: Graph Neural Network)
- Learns from given clauses:
  - *Positive* if in a proof
  - *Negative* otherwise
- Features :- given clause + conjecture + theory.
- Guides E via a weight function.

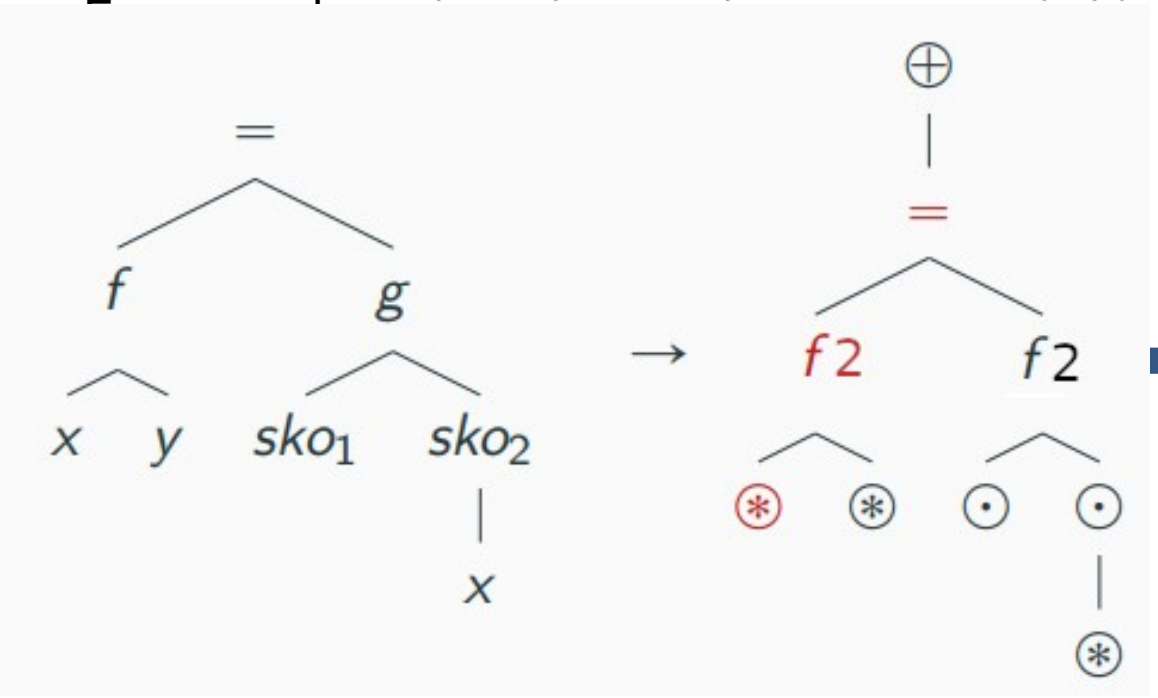
ENIGMA Anonymous: Parental Guidance

- Statistical machine learning for clause filtering.
  - LightGBM (a gradient boosted decision tree framework)
- Learns from *all* generated clauses' parents:
  - *Positive* if *any* child is in a proof
  - *Negative* otherwise
- Features :- parent clause 1 + parent clause 2 + conjecture + theory.
- Scores *valid* pairs of parents:
  - Freezes children whose parents score below a threshold.
  - Unfreeze and simplify clauses if the unprocessed set empties.

Featurization: clauses ➜ vectors

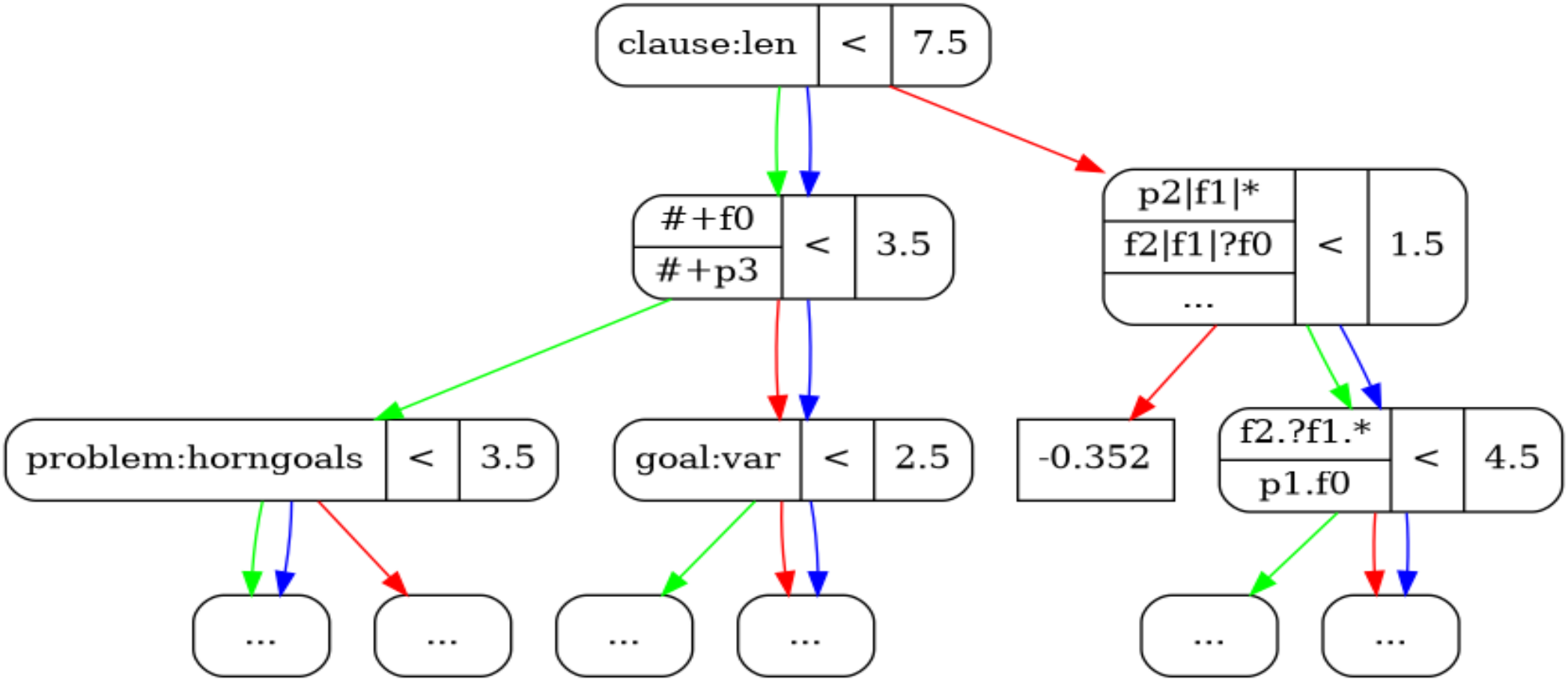
- Treat clauses as trees.
- Abstract vars and skolem symbols.
- Anonymize function and predicate symbols of arity *n* with "**f***n*" or "**p***n*".
  - (We simply removed types from the data for now . . ..)
- Hash features to reduce dimensionality.
- The clause vector consists of feature counts.

$$f(x, y) = g(\mathrm{sko}_1, \mathrm{sko}_2(x))$$



#	feature	count
1	(⊕,=,a)	0
⋮	⋮	⋮
11	(⊕,=,f2)	1
12	(⊕,=,f2)	1
13	(=,f2,⊛)	2
14	(=,f2,⊙)	2
15	(f2,⊙,⊛)	1
⋮	⋮	⋮

Gradient Boosted Decision Tree



*XGBoost tree with non-anonymized watchlist features

ENIGMA-GNN

- **Graph Neural Network**
- **Directed hypergraph for a set of clauses**
- Anonymized symbol names
- Nodes: clauses, functions and predicate symbols, unique (sub)terms, and literals
- Hyperedges:
  1) Clauses and literals
  2) Functions and predicates with subterms
- **Message passing rounds follow formula structure**

➔ Clause embedding and a prediction layer

Argument ordering is (partially) preserved:

- Application $a = f(x_1, x_2, \ldots, x_n)$ is represented by a set of 4-ary hyperedges $(f, a, x_1, x_2), (f, a, x_2, x_3), \ldots, (f, a, x_{n-1}, x_n)$.

Hyperedges

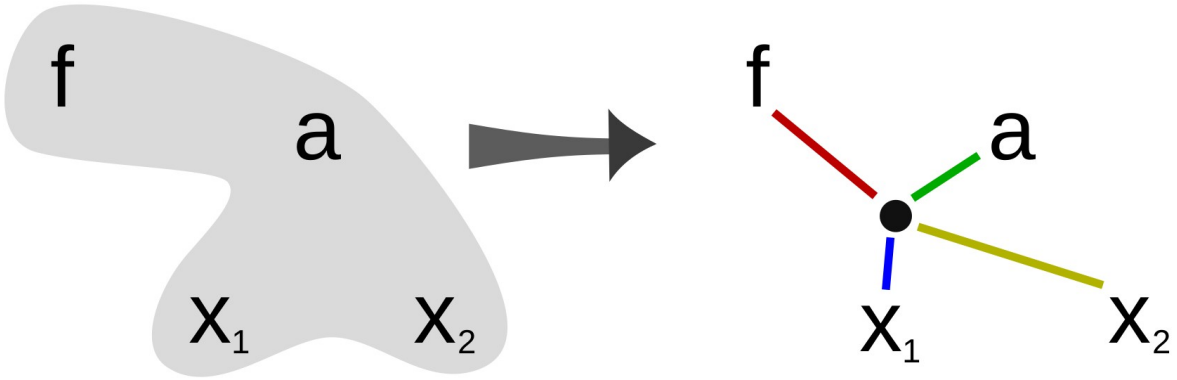
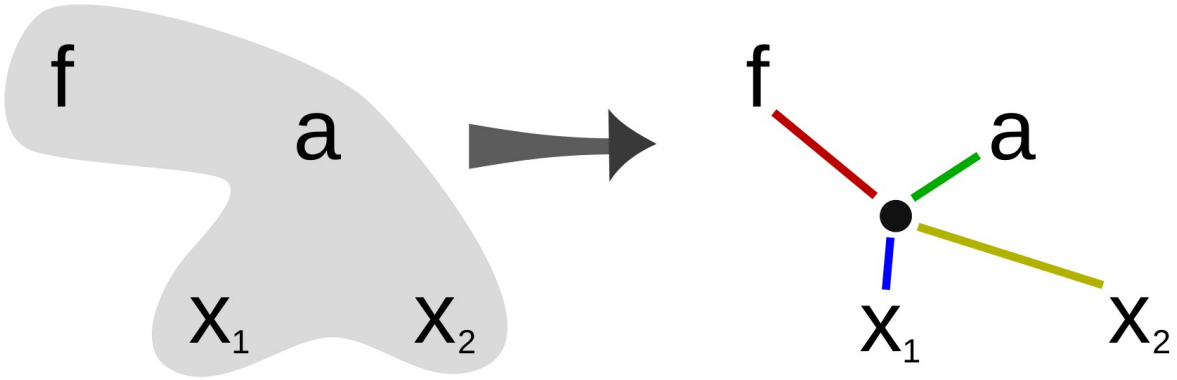


Image courtesy of
http://arg.ciirc.cvut.cz/teaching/mlr20/slides/ai4reason-mlr20-gnn.pdf

Argument ordering is (partially) preserved:

- Application $a = f(x_1, x_2, \ldots, x_n)$ is represented by a set of 4-ary hyperedges $(f, a, x_1, x_2), (f, a, x_2, x_3), \ldots, (f, a, x_{n-1}, x_n)$.

Hyperedges



- **Invarient under negation: embedding of ¬S is the negation of S's.**

Image courtesy of
http://arg.ciirc.cvut.cz/teaching/mlr20/slides/ai4reason-mlr20-gnn.pdf

ENIGMA-GNN for TFF

- **Graph Neural Network**
- **Directed hypergraph for a set of clauses**
- Anonymized symbol names
- Nodes: clauses, functions and predicate symbols, unique (sub)terms, and literals
- Node types receive different initial embeddings (of size 1-4)
  - Initial embeddings can be trained for specific TFF types.
  - Future work: add type nodes to preserve anonymity.

ENIGMA-GNN for Premise Selection

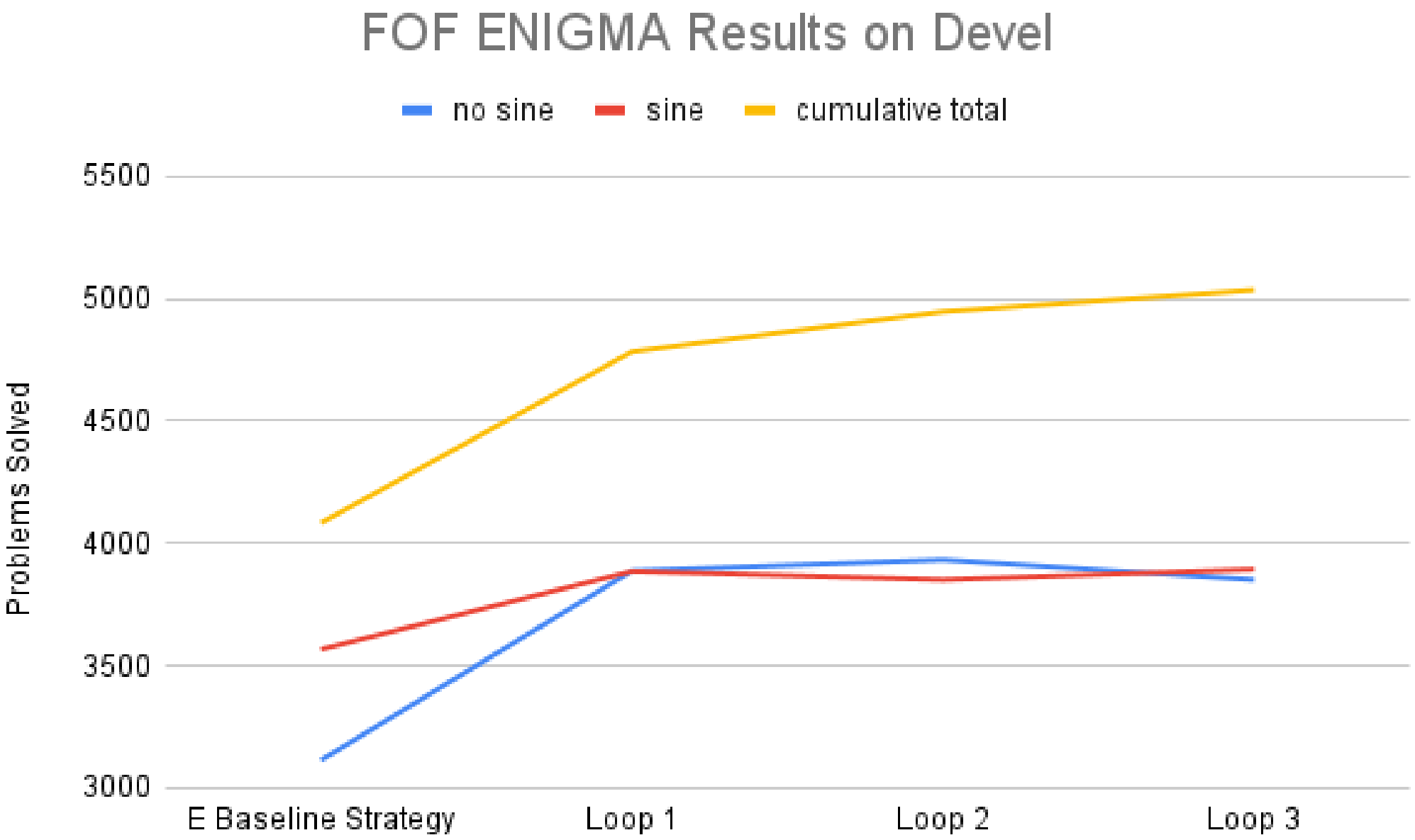
- Learns from unguided E and ENIGMA proof data:
  - Premises are *positive* if in a proof.
  - Premises are *negative* if not.

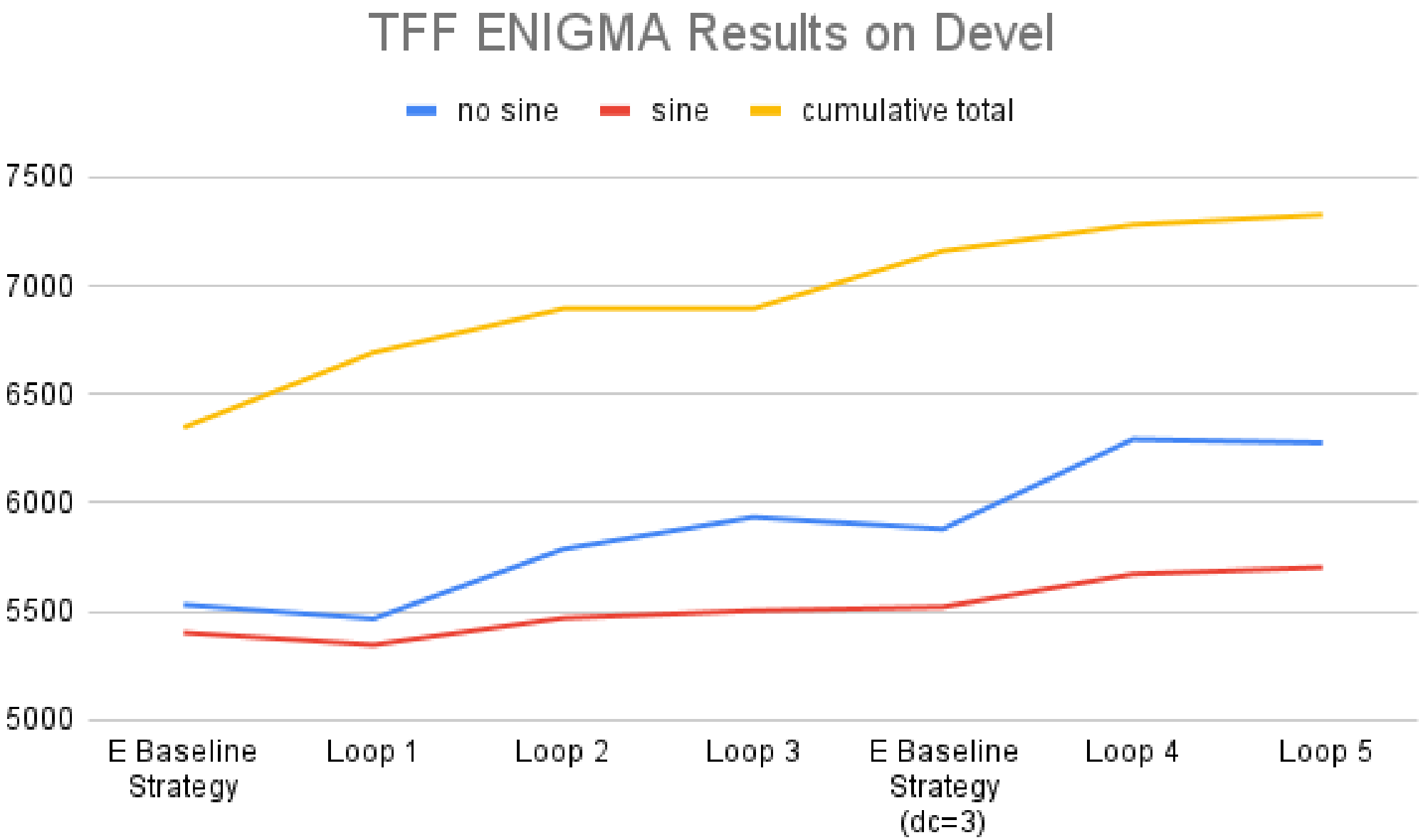
1) Load conjecutre and MePo-suggested premises.
2) 10 layers of message passing.
3) Final layer predicts the score of each premise.
4) Form *premise slices*:
   1) The top-k premises for k in {16, 32, 64, 128, 256}.
   2) Premises with a score above k in {1,0,−1,−2,−3,−4}.

Model Training

Training Paradigms

- Looping:
  1) Run E (with ENIGMA) to grow proof data
  2) Train models on proof data
  3) Go to 1
- If there's too much data:
  - Take data from runs in a greedy cover (– PG –)
  - Take up to 3 proofs for each problem (– CS –)
    - (shortest, longest, random)
- Optuna for LightGBM parameter grid search.


FOF ENIGMA Results on Devel
no sine
sine
cumulative total
Problems Solved
5500
5000
4500
4000
3500
3000
E Baseline Strategy
Loop 1
Loop 2
Loop 3


TFF ENIGMA Results on Devel
no sine
sine
cumulative total
7500
7000
6500
6000
5500
5000
E Baseline Strategy
Loop 1
Loop 2
Loop 3
E Baseline Strategy (dc=3)
Loop 4
Loop 5

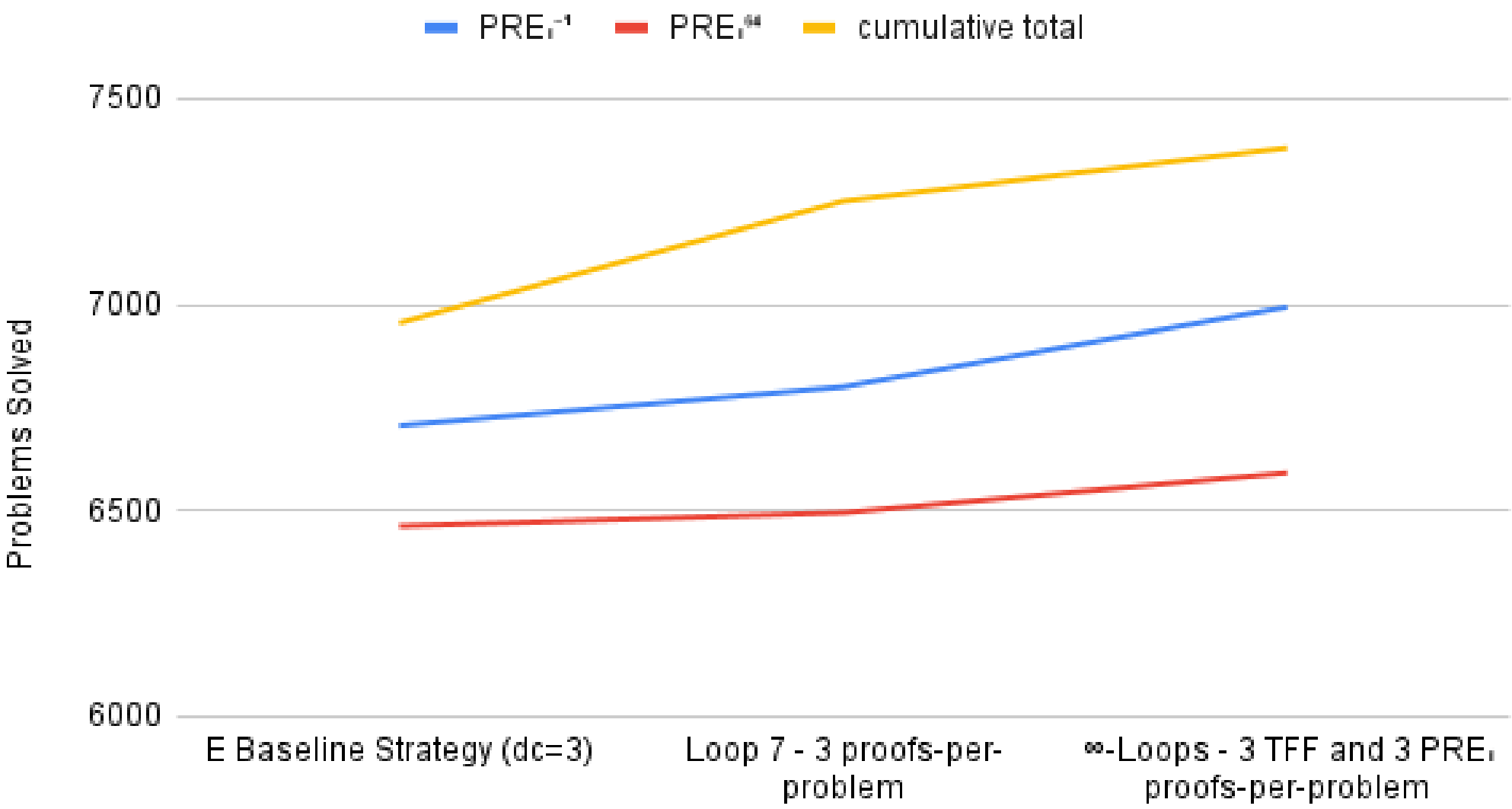
Observations

- As noted in Seventeen, E and ENIGMA perform better on TFF.
- Note the boost from getting the E parameter right: “--definitional-cnf=3”.
- The loops and use of sine are complementary.

Premise Selection 1

- Here the first GNN premise selection is done:
  - Types are forgotten.
- Training:
  - Lots of proof-dependence deduplication
  - Randomly remove negatives until size is 500KB
  - 2 epochs, 12 hours on NVIDIA Volta 100
- PRE_1^{-1} and PRE_1^{64} are the best two slices*.
- Note that unguided E's performance grows by 15%.

ENIGMA Results on Devel with 1st Round Premise Selection



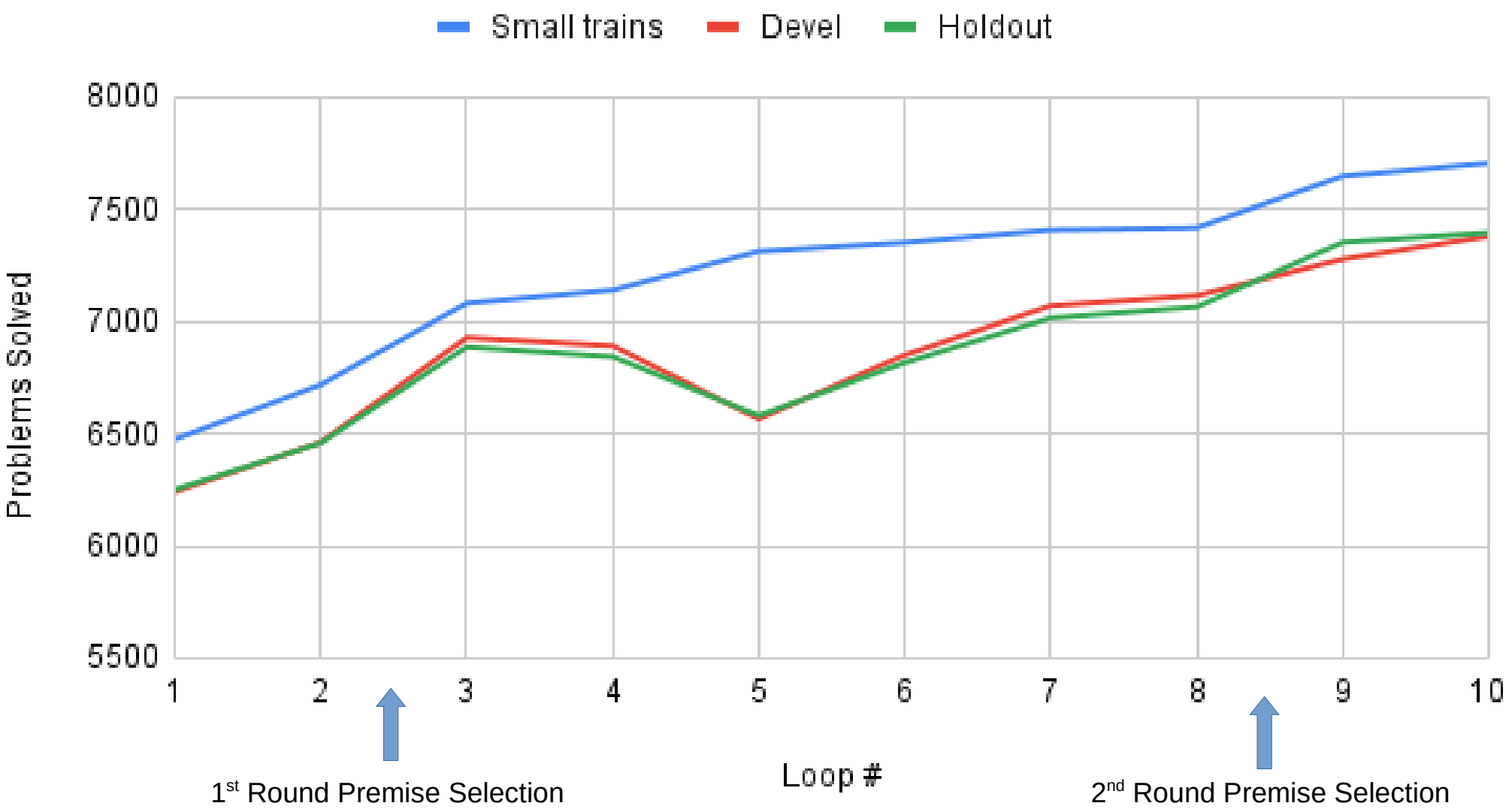
Parental Guidance Training

- Parental Guidance (PG) models are co-trained with Clause Selection (CS) models.
  - Both co-trained model and the best standalone model are run with PG.
  - Training PG with a fixed CS model seems best.
- Parental filtering threshold is tuned on 300-prob devel set (from {0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7}).
- The best model(s) are run on the full training set.

Premise Selection 2

- The second GNN premise selection round:
  - Identify 2539 types occurring more than 10,000 times.
  - Add typed variables for these to learn initial embeddings.
  - Other types are mapped to a generic variable embedding.
- Training:
  - Vampire and CVC5 TFF data is also included.
  - 2 epochs, 24 hours on NVIDIA Volta 100
- PRE_2^{-1} and PRE_2^{0} are the best two slices*.

Parental Guidance Training: small trains, devel and holdout



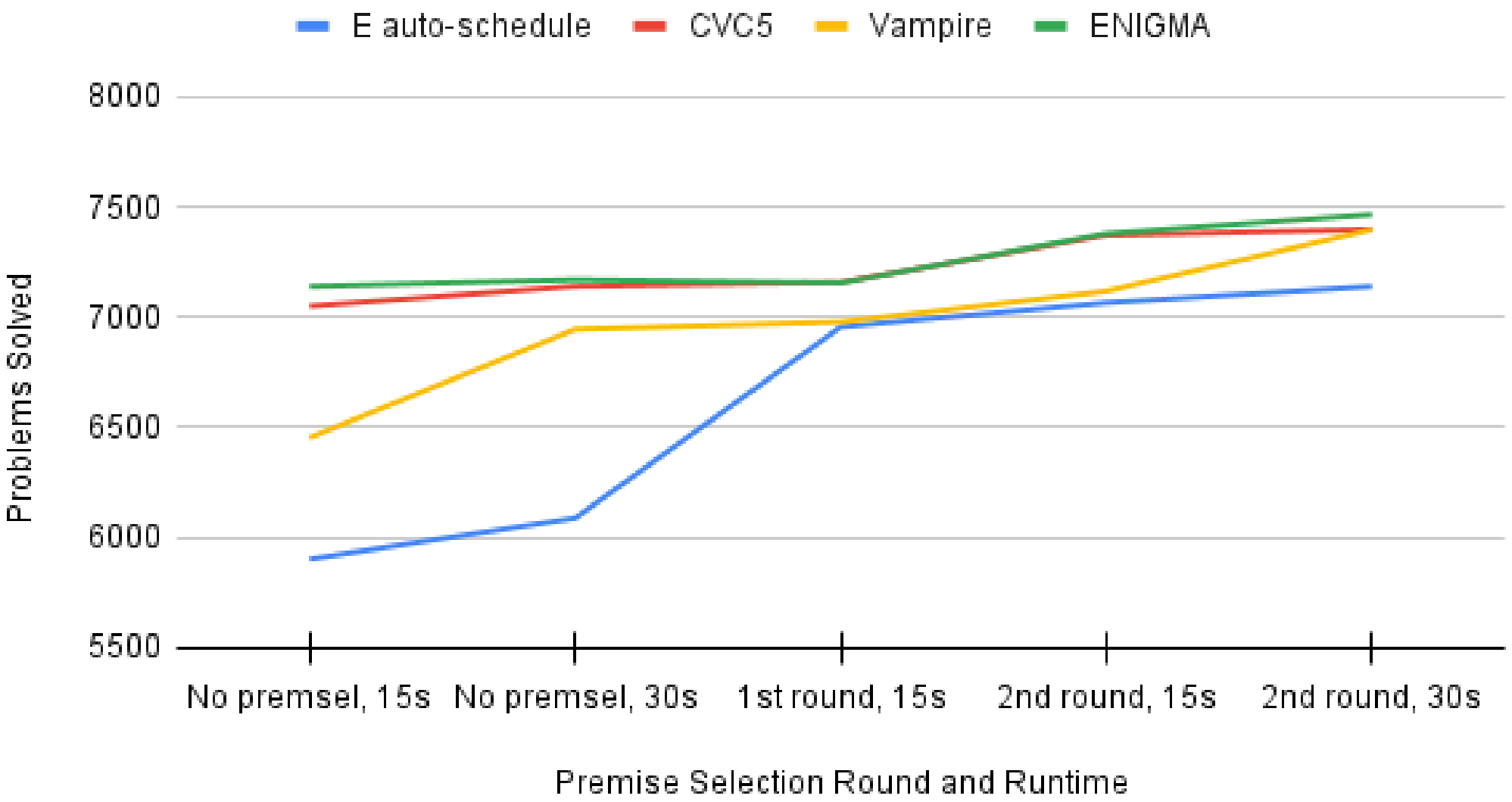
Observations

- The ENIGMA loop 10 model solves:
  - 137,893 (55.4%) on the PRE_2^{-1} training set.
  - 133,390 (53.6%) without premise selection.
  - 7379 (53.4%) on PRE_2^{-1} devel set.
  - 7133 (51.6%) without premise selection.

Final Comparison with ATPs/SMTs

method	*E auto-schedule*	*CVC5*	*Vampire (CASC)*	*ENIGMA (1 strat)*
15s dev, noprem	5891	7053	6452	7133
15s test, noprem	5903	7051	6454	7139
30s test, noprem	6089	7140	6945	7170
15s dev, preds -1 (1st round)	6968	7211	7023	7191
15s test, preds -1 (1st round)	6956	7158	6978	7155
15s dev, preds -1 (2nd round)	7074	7394	7132	7379
15s test, preds -1 (2nd round)	7066	7372	7118	**7395**
30s test, preds -1 (2nd round)	7139	7398	7397	**7466**

Final ATP&SMT Comparison on 13,818 problem holdout set



Conclusions

- We developed effective versions of ENIGMA systems for Isabelle Sledgehammer problems.
- The GNN premise selection improves (in 15s):
  - E auto-schedule by 19.7%
  - Vampire by 10.2%
  - CVC5 by 4.5%
- The best single-strategy ENIGMA performs on par with CVC5 (with a slight 0.1 to 1% gain).