# Learning Rules Explaining Interactive Theorem Proving Tactic Prediction

Liao Zhang, David M. Cerna, and Cezary Kaliszyk

by Yuzhe Zhou

2024-10-21

# Authors

Liao Zhang
University of Innsbruck, Innsbruck, Austria

David M. Cerna
Czech Academy of Sciences, Prague, Czechia

artificial intelligence, automated reasoning, unification, anti-unification, computational logic, and proof theory.

Cezary Kaliszyk
University of Melbourne, Melbourne, Australia

formal verification, interactive theorem proving, automated reasoning, and the application of machine learning to mathematics

# Interactive Theorem Provers

**Automated Theorem Proving (ATP)** attempts to fully automate the process of proving mathematical theorems.

**Limitations**: Complex proofs often exceed the capabilities of fully automated systems.

**ITP** Combines **human expertise** with **automated reasoning tools** to ensure the correctness of proofs in mathematics and computer science.

Humans provide insight and intuition where automation fails, guiding the proof process.

# Coq

Theorem **add_assoc** : forall **n m p** : **nat**, (n + m) + p = n + (m + p).

proof.

```
1 subgoal
n, m, p : nat
______________________(1/1)
(n + m) + p = n + (m + p)
```

induction n.

```
2 subgoals
m, p : nat
________________________(1/2)
(0 + m) + p = 0 + (m + p)

n' : nat
IHn' : forall m p : nat, (n' + m) + p = n' + (m + p)
m, p : nat
________________________(2/2)
((S n') + m) + p = S n' + (m + p)
```

simpl.

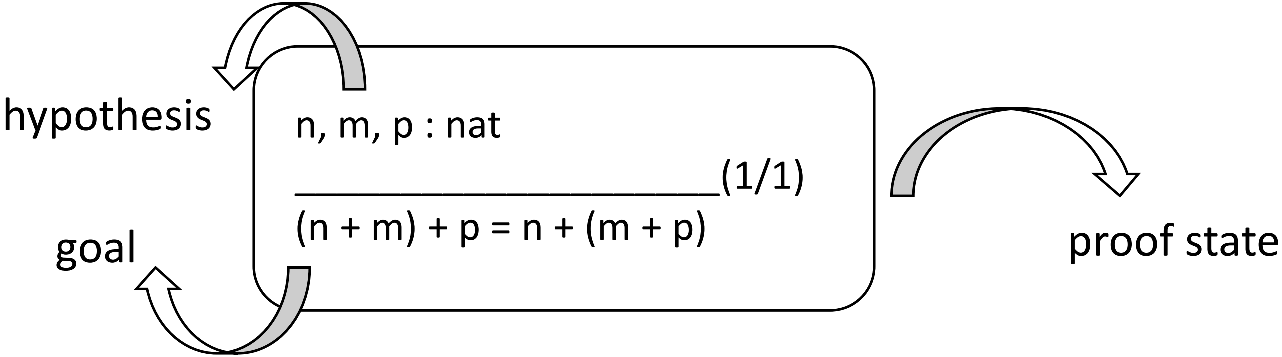
```
1 subgoal
m, p : nat
__________________(1/1)
m + p = m + p
```

reflexivity.

```
No more subgoals.
```

...

# Coq




**tactic**:
Input: proof state
Output: proof state

Numerous investigations have focused on providing the user with guidance through tactic suggestion.

# k-NN method

k-NN method take a goal $g$, select a goal $g$' most similar to $g$, and rank the particular tactics relevant for solving $g$' based on their likelihood of solving $g$.

**Weaknesses**:

Cannot be used in **new** theory.

It lack interpretability.

This work combines **k-NN** with **ILP** to provide tactics.

# Learned Rules

```
tac(A,"simpl") :-
  goal_node(const,A,B,C), goal_node(construct,A,D,E),
  goal_above(A,B,D), goal_node(construct,A,F,E),dif(F,D),
  goal_above(A,B,F).
```

x, y : nat

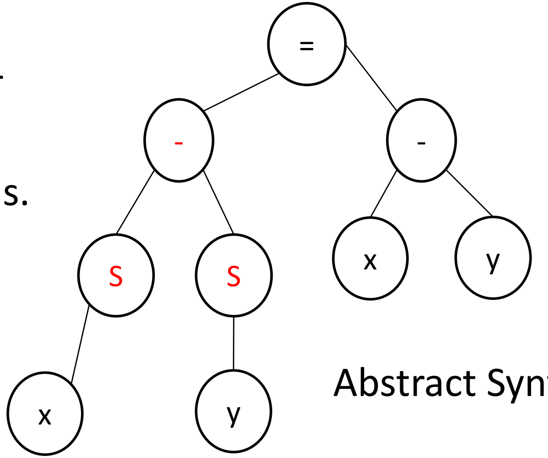
S x – S y = x - y

use tactic “simpl”

x, y : nat

x – y = x - y

The goal may be simplified if its AST contains **a constant above two constructors** with different positions.




Abstract Syntax Tree (AST)

# ILP Problem

For a **specific** tactic $tac$,

$$B \cup H \vDash E$$

$H$: Rules about the tactic

$E$:

**Positive examples**: The proof states to which it is applied are regarded as the positive examples

**Negative examples**: The proof states to which the tactics different from $tac$ are applied are regarded as the negative examples

$B$: User-defined background knowledge

# Background Knowledge

**Representation Predicates**: Nodes in the AST in the proof state. (e.g. goal_node/3)

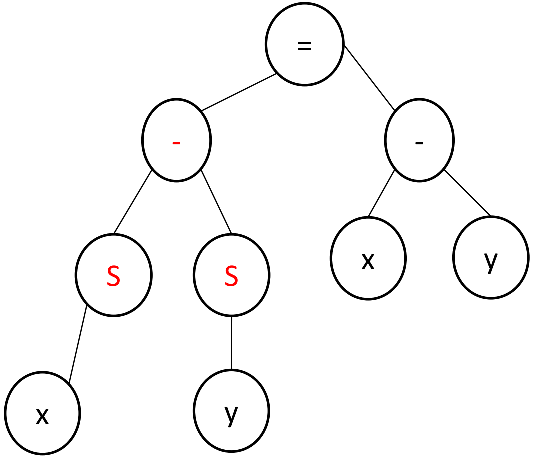
**Feature Predicates**:

left, above, equality between terms, dif, root

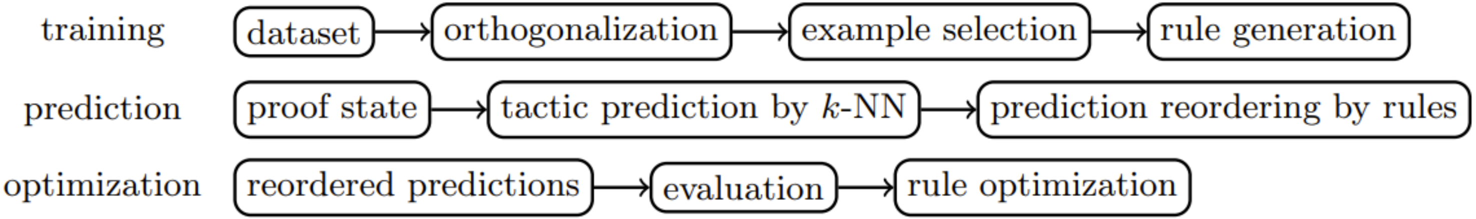
**Anonymous Predicates**:

for generalization

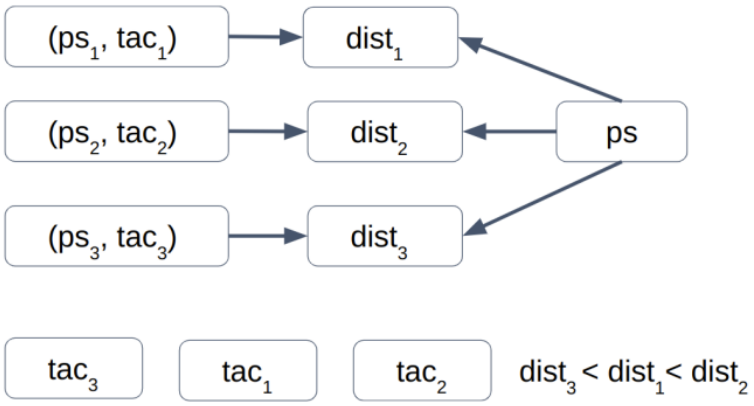
goal_node(A, B, C) -> goal_node(const, A, B, C)

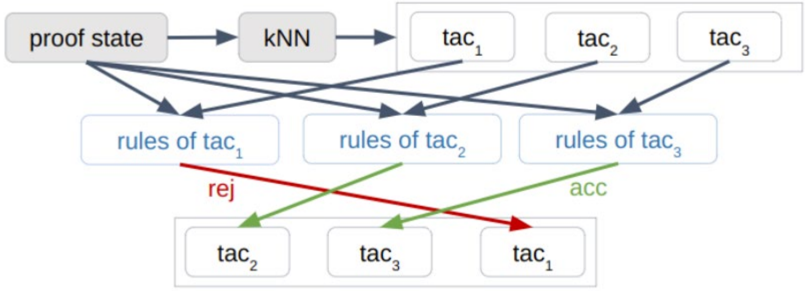


# Learning Framework



# Prediction

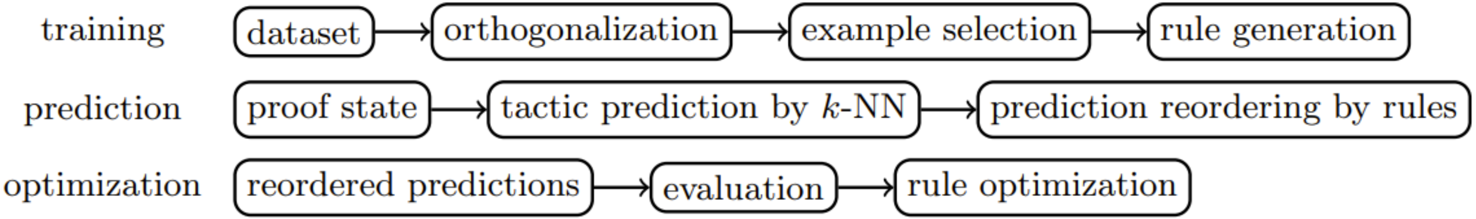



k-NN

Reorder k-NN predictions

# Learning Framework



# Optimization

**Reason**: To remove some low-quality rules to increase the overall performance of rules

If a rule is overly general → $FPs \;>>\; TPs$ → Remove it

They use $precision$ as the metric of the quality of a single rule

$$precision = \frac{TP}{TP + FP}$$

# Experiments

**Dataset**: Coq standard library. It consists of 41 theories and 151,678 proof states.

**ILP**: Aleph

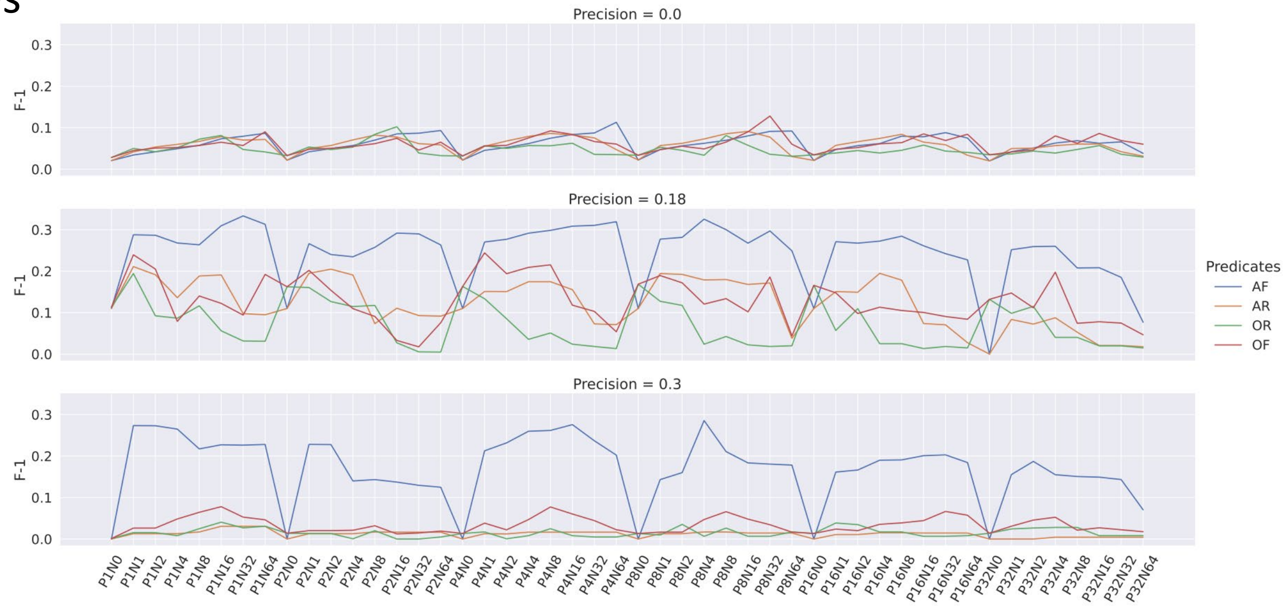
They conducted the experiments in the transfer-theory setting, which means different Coq theories are used for training, validation, and testing.

# Parameter Optimization

**Hyperparameters:**

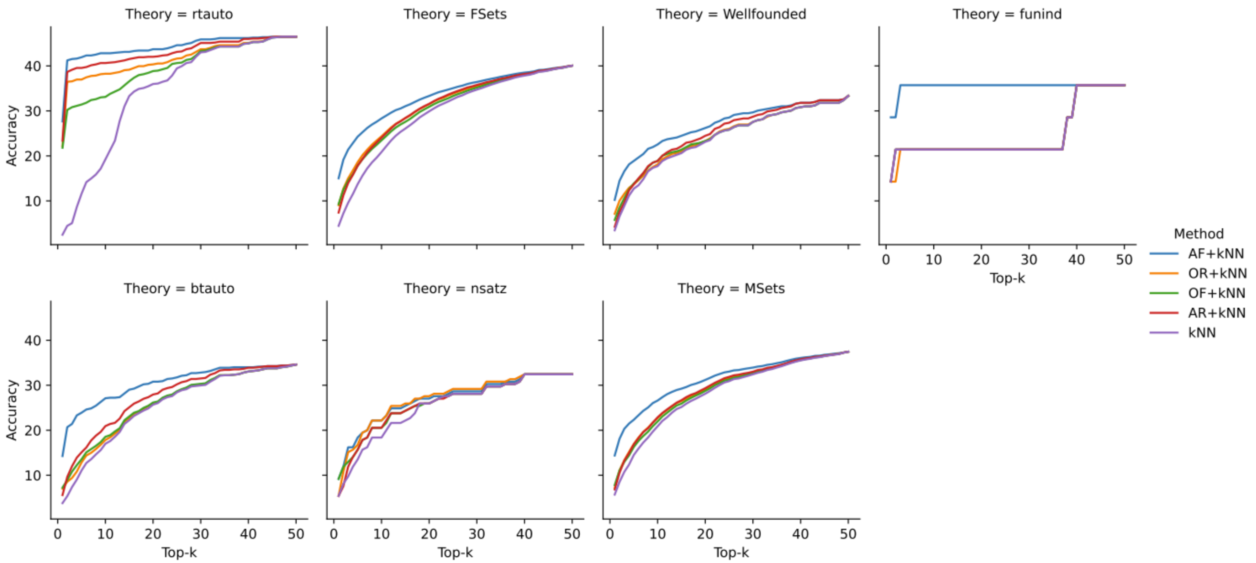
1. The number of positive examples and negative examples
2. Background knowledge
3. Filter threshold of rules

**AF:** anonymous feature predicates
**AR:** anonymous representation predicates
**OF:** original feature predicates
**OR:** original representation predicates



# Testing

| Parameter | AF | AR | OF | OR |
|---|---|---|---|---|
| Precision | 0.18 | 0.12 | 0.18 | 0.12 |
| Positive | 1 | 16 | 4 | 1 |
| Negative | 32 | 1 | 1 | 1 |



# Conclusion

**Pros**

First application of ILP to ITP.

New feature predicates, allowing us to calculate features in learning if necessary.

Empirically show improvement over the non-filtering approaches.

**Cons**

Can not use modern ILP approach

The usage of some tactics such as induction is inherently complicated