A Survey on Deep Learning for Theorem Proving

Zhaoyu Li[1], **Jialiang Sun**[1], **Logan Murphy**[1], **Qidong Su**[1], **Zenan Li**[2], **Xian Zhang**[3]
Kaiyu Yang[4]*, **Xujie Si**[1,5]
[1]University of Toronto, [2]Nanjing University, [3]Microsoft Research Asia, [4]Meta FAIR,
[5]CIFAR AI Chair
{zhaoyu, six}@cs.toronto.edu

Abstract

Theorem proving is a fundamental aspect of mathematics, spanning from informal reasoning in natural language to rigorous derivations in formal systems. In recent years, the advancement of deep learning, especially the emergence of large language models, has sparked a notable surge of research exploring these techniques to enhance the process of theorem proving. This paper presents a comprehensive survey of deep learning for theorem proving by offering (i) a thorough review of existing approaches across various tasks such as autoformalization, premise selection, proofstep generation, and proof search; (ii) an extensive summary of curated datasets and strategies for synthetic data generation; (iii) a detailed analysis of evaluation metrics and the performance of state-of-the-art methods; and (iv) a critical discussion on the persistent challenges and the promising avenues for future exploration. Our survey aims to serve as a foundational reference for deep learning approaches in theorem proving, inspiring and catalyzing further research endeavors in this rapidly growing field. A curated list of papers is available at https://github.com/zhaoyu-li/DL4TP.

1 Introduction

Proving theorems is a cornerstone of mathematics. Since the era of Euclid around 300 B.C., people have crafted theorems and proofs using a blend of natural language and mathematical symbols, meticulously evaluating their correctness through manual inspection. In the 1950s, a paradigm shift occurred with the exploration of computer-assisted proofs (Davis, 1957; Davis & Putnam, 1960), wherein a machine automatically applies deduction rules to prove assertions. These innovations laid the groundwork for the subsequent development of interactive theorem provers (Bruijn, de, 1970; Milner, 1972), enabling people to construct more intricate theorems and proofs by interacting with these systems. Building upon these advancements, later research extended the scope of theorem proving beyond mathematics, applying it to various practical applications such as software verification (Schumann, 2001) and hardware design (Kern & Greenstreet, 1999).

Exploring learning-based approaches for theorem proving has been a long-standing research focus, dating back to the 1990s (Suttner & Ertel, 1990; Denzinger et al., 1999). The recent development of deep learning, especially with the evolution of large language models (LLMs), has ignited a wave of research interest in this area again. As shown in Figure 1, the volume of papers on deep learning for theorem proving has grown approximately from 2 in 2016 to 45 in 2023, continuing to rise in 2024. However, despite such remarkable growth, this domain is characterized by a wide range of tasks, methods, datasets, and evaluations, which lack a cohesive framework to comprehend the true extent of progress and identify the underlying challenges and potential future work.
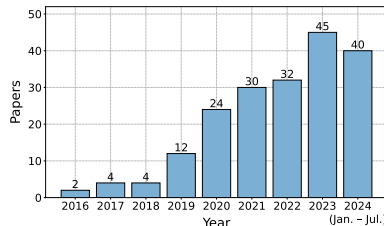

Figure 1: Papers on deep learning for theorem proving over the years (data for 2024 is up to July).

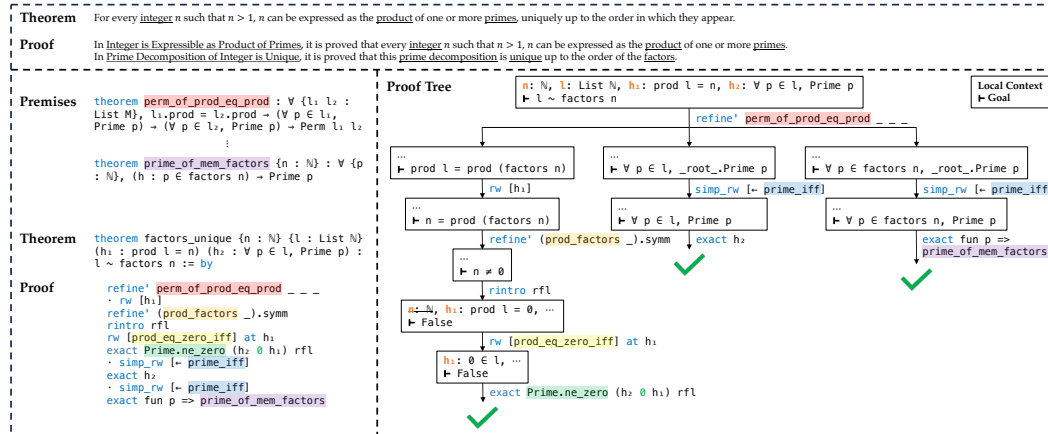*Research conducted while Kaiyu Yang was at Caltech.

Figure 2: **Top:** The informal statement and proof of the Fundamental Theorem of Arithmetic in ProofWiki. **Bottom Left:** The formal statement and proof of the same theorem in the mathlib library (mathlib Community, 2020) of Lean 4. **Bottom Right:** The corresponding proof tree illustrating the formal proof process in Lean 4. Only changes in the local context of each node are marked for clarity. The references and premises used in the informal and formal proof are highlighted by underlines and colors respectively.

In this paper, we provide a comprehensive survey of more than 180 research papers in deep learning for theorem proving, aiming to map out the current research landscape and highlight key advancements systematically. We begin with the background for informal and formal settings of theorem proving (§2). Subsequently, we delve into the details of the tasks and methods within this domain (§3), which include autoformalization, premise selection, proofstep generation, proof search, and other tasks. We also review the datasets for theorem proving (§4), including manually curated and synthetically generated ones. Moreover, we examine the evaluation metrics and assess state-of-the-art performance (§5). Following this, we discuss the prevailing challenges and conclude with future directions (§6).

2 Background

In this section, we recall some fundamental concepts of theorem proving, including both informal and formal settings. Figure 2 shows an illustrative example of these two settings.

2.1 Informal Theorem Proving

Informal theorem proving involves establishing the truth of mathematical statements building on existing knowledge via intuitive reasoning and natural language explanations. This mirrors how people learn and prove theorems in everyday mathematics. For instance, to prove the Fundamental Theorem of Arithmetic in Figure 2 (Top), one needs to comprehend basic concepts like primes and might apply established results to draw a conclusion. Despite the ubiquity of informal theorem proving, as mathematics evolves, the theories and proofs tend to be more intricate, making verifying their correctness increasingly difficult.

2.2 Formal Theorem Proving

Formal theorem proving represents theorems and proofs in a machine-verifiable format, ensuring their correctness using rigorous logical rules. This field can be broadly classified into two paradigms: automated theorem proving (ATP) and interactive theorem proving (ITP).

ATP aims to verify formal statements fully automatically. Saturation-based theorem provers, including E (Schulz, 2002) and Vampire (Kovács & Voronkov, 2013), mainly operate on first-order logic (FOL) to autonomously generate logical consequences from a set of axioms until a proof or refutation is derived, or computational limits are reached. Similarly, geometric ATP systems such as GEX (Chou et al., 2000) prove geometry problems by iteratively applying

deduction rules. Other approaches, such as tableau-based methods like leanCoP (Otten & Bibel, 2003) and instantiation-based methods like iProver (Korovin, 2008), use other forms of proof calculi for proof construction. In addition to these, Boolean satisfiability (SAT) solvers (e.g., MiniSat (Eén & Sörensson, 2003), CaDiCaL (Biere, 2019)) and satisfiability modulo theories (SMT) solvers (e.g., Z3 (De Moura & Bjørner, 2008), CVC5 (Barbosa et al., 2022)) play a crucial role in ATP by efficiently handling propositional logic and theories such as arithmetic, bit-vectors, and arrays. Despite the sophisticated designs of these ATP systems, the inherent vast search space often limits their practicality in more complex problems.

In ITP, humans collaboratively prove theorems by interacting with *proof assistants*, such as Isabelle (Paulson, 1994), HOL Light (Harrison, 1996), Coq (Barras et al., 1999), Metamath (Megill & Wheeler, 2019), and Lean (Moura & Ullrich, 2021). These proof assistants typically enable users to formalize theorems in higher-order logic and provide a language to build verifiable proofs. As shown in Figure 2 (Bottom Left), to prove a theorem (initial goal) in Lean, one can use *tactics* like `refine'` and `rw` as the proof steps. Applying a tactic either finishes the goal or decomposes it into simpler sub-goals, and the proof is complete when no further goals remain. When proving the current goal, one can apply assumptions in the local context and previously proven premises in the environment as tactic arguments. For example, the premise `perm_of_prod_eq_prod` is used as the argument of `refine'`. The proving process can be modeled as a proof tree, where each node is a proof state with a goal and its local context, and each edge is a tactic, as shown in Figure 2 (Bottom Right). Using proof assistants, researchers have successfully formalized and proved landmark theorems like the Four Color Theorem (Gonthier, 2008) and the Kepler Conjecture (Hales et al., 2017), and verified the correctness of critical software such as the seL4 microkernel (Klein et al., 2009) and the CompCert C compiler (Leroy et al., 2016). However, it is worth noting that these projects took several Ph.D. years to complete, requiring substantial labor and expertise.

3 Tasks and Methods

The emergence of deep learning has opened new avenues for the landscape of theorem proving, either enhancing or substituting traditional components involved in the process. This section categorizes and summarizes existing deep learning approaches into 5 tasks: autoformalization, premise selection, proofstep generation, proof search, and others.

3.1 Autoformalization

Autoformalization aims to convert informal theorems and proofs into machine-verifiable formats automatically. This task is notoriously challenging, requiring a profound understanding of semantics across informal and formal mathematics (Kaliszyk et al., 2014; 2015). Nonetheless, the success of autoformalization promises to facilitate the verification of mathematical papers and pave the way for general-purpose reasoning engines (Szegedy, 2020).

Wang et al. (2018; 2020) first explore using deep learning models in autoformalization. Inspired by the sequence-to-sequence models in neural machine translation (Sutskever et al., 2014; Cho et al., 2014), they experiment various encoder-decoder frameworks (Luong et al., 2017; Lample et al., 2018; Lample & Conneau, 2019) to convert LaTeX-written mathematical problem texts to the Mizar language (Rudnicki, 1992). Subsequent studies (Bansal & Szegedy, 2020; Cunningham et al., 2022) utilize similar neural architectures for HOL Light and Coq.

The recent development of LLMs and their in-context learning capabilities (Brown et al., 2020) have provided new opportunities for autoformalization. Some researchers study the potential of advanced LLMs using few-shot prompting techniques: Wu et al. (2022); Agrawal et al. (2022); Gadgil et al. (2022) leverage PaLM (Chowdhery et al., 2023) and Codex (Chen et al., 2021) to translate high school and undergraduate-level mathematical problems into Isabelle and Lean. LeanEuclid (Murphy et al., 2024) uses GPT-4 (Achiam et al., 2023) and GPT-4V to formalize both theorems and proofs in Euclidean geometry. Other research efforts (Jiang et al., 2023b; Patel et al., 2023; Zhao et al., 2024; Lu et al., 2024; Ying et al., 2024a; Poiroux et al., 2024) propose more structured approaches to autoformalization: For example, DSP (Jiang et al., 2023b) utilizes Minerva (Lewkowycz et al., 2022) to draft

informal proofs and map them into formal sketches, with ATP systems employed to fill in the missing details in the proof sketch. Zhao et al. (2024) improves informal proofs and formal sketches in DSP with sub-goal proofs and prompt selection respectively. Additionally, a line of research (Azerbayev et al., 2023; Jiang et al., 2023a; Azerbayev et al., 2024; Shao et al., 2024; Ying et al., 2024b) focuses on training LLMs on large-scale datasets containing both informal and formal mathematical data to evaluate their performance on autoformalization. Recent studies (Liu et al., 2023; Pan et al., 2023; Olausson et al., 2023; Ye et al., 2023; Zhou et al., 2024a; Huang et al., 2024; Xin et al., 2024b; Jiang et al., 2024; Quan et al., 2024; Xin et al., 2024a) also apply autoformalization as a key step in various downstream tasks. For instance, SAT-LM (Ye et al., 2023) uses LLMs to formalize natural language problems using declarative prompting and solve them using Z3 on several reasoning tasks. DTV (Zhou et al., 2024a) leverages autoformalization to ground LLM reasoning, formalizing LLM-generated answers and verifying them with ATP tools. Besides these efforts, Wu et al. (2022); Azerbayev et al. (2023); Jiang et al. (2023a); Lu et al. (2024) explore advanced LLMs like Codex and GPT-4 for informalization, i.e., the translation of formal statements into natural language.

3.2 Premise Selection

Given a large collection of previously proven lemmas, premise selection is to retrieve the helpful lemmas that can contribute to a successful proof. It is an enduring challenge in both mathematical research and ATP/ITP systems (Kühlwein et al., 2012; Alama et al., 2014).

The seminal works (Irving et al., 2016; Kaliszyk et al., 2017) model premise selection as a binary classification task, embedding theorems and premises with a variety of neural networks including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models. These embeddings are then combined to feed into a logit layer to predict their relevance. Follow-up works (Kucik & Korovin, 2018; Bansal et al., 2019; Piotrowski & Urban, 2020a; Proroković et al., 2021; Szegedy et al., 2021) extend previous frameworks by using a better representation of features or more sophisticated architectures like Wavenet (Van Den Oord et al., 2016) and Transformer (Vaswani et al., 2017).

Given the inherent structured nature of mathematical formulas, a stream of research (Wang et al., 2017; Peng & Ma, 2017; Olšák et al., 2019; Goertzel & Urban, 2019; Crouse et al., 2019; Paliwal et al., 2020; Rawson & Reger, 2020; Liu et al., 2022a; Goertzel et al., 2022; Holden & Korovin, 2023; Jakubüv et al., 2023) parses formal statements into trees or graphs and leverages tree-structured neural networks (Tai et al., 2015) or graph neural networks (GNNs) (Duvenaud et al., 2015; Veličković et al., 2018; Xu et al., 2019) for encoding. For example, FormulaNet (Wang et al., 2017) proposes a graph embedding method that preserves the information of edge ordering. Olšák et al. (2019) introduces a GNN framework that captures several logical invariances in FOL formulas. Paliwal et al. (2020) conducts comprehensive experiments to evaluate various designs for the graph representations of formulas in HOL Light. Subsequent works (Li et al., 2021b; Lin et al., 2021) explore graph contrastive learning (Oord et al., 2018; Chen et al., 2020) to train GNNs for premise selection. Moreover, Ferreira & Freitas (2020b); Bauer et al. (2023) construct a dependency graph over a large corpus by representing theorems and premises as nodes and their dependencies as edges, and leverage GNNs to predict the link between nodes for premise selection.

With the advancement of pre-trained language models, some efforts (Ferreira & Freitas, 2020a; Welleck et al., 2021) fine-tune BERT (Devlin et al., 2019)-like models to embed natural language statements into vectors and select premises using a linear classifier layer. Later works (Ferreira & Freitas, 2021; Tran et al., 2022; Trust et al., 2022; Dastgheib & Asgari, 2022; Yeh et al., 2023; Yang et al., 2023) leverage different pre-trained models (Liu et al., 2019; Song et al., 2020; Xue et al., 2022) for encoding and retrieve informal/formal premises based on several similarity metrics. Specifically, ReProver (Yang et al., 2023) trains its retriever based on dense passage retrieval (DPR) (Karpukhin et al., 2020) to select premises in Lean. Han et al. (2021) also explores fine-tuning over large informal mathematical corpora using the contrastive objective (Oord et al., 2018), while PACT (Han et al., 2022) uses the auto-regressive objective for formal premise selection. Additionally, several research (Kovriguina et al., 2022; Tworkowski et al., 2022; Mikuła et al., 2024) design a second phase to re-rank the selected subset of premises, enabling a more accurate selection.

3.3 Proofstep Generation

Proofstep generation is the core problem for theorem proving, which aims to predict one or more steps to build the proof of a theorem. This task also refers to tactic prediction in ITP, which has been widely studied in tactic-based ATP systems (hammers) (Böhme & Nipkow, 2010; Blanchette et al., 2016; Czajka & Kaliszyk, 2018).

A stream of research (Whalen, 2016; Huang et al., 2019; Bansal et al., 2019; Paliwal et al., 2020; Sanchez-Stern et al., 2020; Wu et al., 2021b; Rute et al., 2024) treats tactic prediction as a classification problem and uses separate neural networks to predict the tactic and its arguments. For example, Gamepad (Huang et al., 2019) employs TreeLSTM (Tai et al., 2015) to encode the proof states and two distinct linear layers for tactic and argument prediction. Proverbot9001 (Sanchez-Stern et al., 2020) uses a feed-forward neural network and an RNN to predict the tactic and its arguments respectively. Besides these works, ASTactic (Yang & Deng, 2019) proposes a decoder that generates the tactic as a program, using an RNN to control the generation based on a predefined context-free grammar. Later studies (First et al., 2020; First & Brun, 2022; Sanchez-Stern et al., 2023) improve ASTactic by incorporating prior proof scripts, combining varied models, and modeling identifiers of theorems.

Subsequent advancements (Polu & Sutskever, 2020; Polu et al., 2023; Han et al., 2022; Jiang et al., 2021; Zhang et al., 2023a; Yeh et al., 2023; Xiong et al., 2023; Wang et al., 2023a; Vishwakarma & Mishra, 2023; Gloeckle et al., 2023; First et al., 2023; Xin et al., 2024a; Wang et al., 2024; Lin et al., 2024a; Wu et al., 2024) formulate tactic prediction as language modeling. Specifically, GPT-f (Polu & Sutskever, 2020) first apply a conditional language modeling objective to train decoder-only Transformers to generate a proof step in the format of `GOAL <GOAL> PROOFSTEP <PROOFSTEP><EOT>`. Baldur (First et al., 2023) applies a similar objective to generate or repair the whole proof at once. POETRY (Wang et al., 2024) introduces a level-by-level approach, recursively generating a formal sketch of the proof at each level and solving the current level's theorem or conjecture until the proof is complete. Several studies (Szegedy et al., 2021; Tworkowski et al., 2022; Welleck et al., 2022; Jiang et al., 2022; Yang et al., 2023) also jointly train tactic prediction with premise selection. For instance, NaturalProver (Welleck et al., 2022) trains GPT-3 (Brown et al., 2020) with constrained decoding to encourage using retrieved references in the proof steps. Thor (Jiang et al., 2022) adds a `<hammer>` token to learn when to invoke a hammer system (Böhme & Nipkow, 2010) for premise selection to simplify the proof. In the geometry domain, Chen et al. (2022); Liang et al. (2023); Trinh et al. (2024) follow the same paradigm, auto-regressively generating the proof sequence at each step. Notably, AlphaGeometry (Trinh et al., 2024) trains a decoder-only Transformer to predict the auxiliary constructions in the proofs of International Mathematical Olympiad (IMO) geometry problems. Besides training on proof data, Azerbayev et al. (2024); Shao et al. (2024); Ying et al. (2024b) train LLMs on extensive general mathematical corpora and evaluate their abilities for generating formal proofs.

With the development of LLMs, some researchers (Zhang et al., 2023b; Yousefzadeh & Cao, 2023; Scheidt, 2023; Frieder et al., 2023a;b;c; Zhang et al., 2024; Poulsen et al., 2024) also explore prompting state-of-the-art LLMs without any additional training to generate proofs across various domains. For example, Frieder et al. (2023b) evaluates the performance of ChatGPT and GPT-4 on completing informal mathematical proofs, and Selene (Zhang et al., 2024) focuses on project-level automated proof in software verification based on the seL4 project. Recent explorations further propose more structured pipelines for formal proof generation (Jiang et al., 2023b; Zhao et al., 2024; Zheng et al., 2024; Xin et al., 2024b; Huang et al., 2024; Thakur et al., 2024): For instance, Lyra (Zheng et al., 2024) leverages two correction strategies, namely tool correction and conjecture correction, as post-processing and error feedback mechanisms to refine incorrect proofs generated by GPT-4.

3.4 Proof Search

Proof search seeks to systematically traverse the vast landscape of potential proof paths to construct a valid proof tree for a given theorem in formal systems. It is not only a long-standing research focus in ATP (Urban et al., 2011; Kaliszyk & Urban, 2015a; Jakubüv & Urban, 2017) but also a vital process for tactic-based models to complete the proof in ITP.

A thread of research treats branching as a classification task and trains deep learning models on successful proof paths in a supervised fashion to guide the search in various ATP systems. Specifically, a large body of works (Loos et al., 2017; Chvalovskỳ et al., 2019; Jakubův & Urban, 2019; Aygün et al., 2020; Jakubův et al., 2020; Suda, 2021b; Chvalovskỳ et al., 2021; Goertzel et al., 2021; Suda, 2021a; Firoiu et al., 2021; Aygün et al., 2022; Goertzel et al., 2022; Jakubův et al., 2023; Bártek & Suda, 2023) exploit various RNNs, GNNs, or hybrid models for the clause selection in saturation-based provers. Similarly, Piepenbrock et al. (2022b); Chvalovskỳ et al. (2023) and Piotrowski & Urban (2020b) focus on guiding instantiation and connection tableau respectively. Some works (Rawson & Reger, 2019; Olšák et al., 2019; Rawson & Reger, 2021; Zombori et al., 2021b; Wei et al., 2024) further combine the supervised trained models as the policy or value networks to guide the Monte Carlo Tree Search (MCTS) or A* search across various ATP systems. Additionally, another direction of research (Kusumoto et al., 2018; Fawzi et al., 2019; Abdelaziz et al., 2020; Zombori et al., 2021a; Crouse et al., 2021; Piepenbrock et al., 2021; 2022a; Liu et al., 2022b; Abdelaziz et al., 2022; Fokoue et al., 2023; McKeown & Sutcliffe, 2023; Shminke, 2023) models proof search as a Markov decision process and applies reinforcement learning (RL) to train and guide the proof search. For instance, TRAIL (Crouse et al., 2021) uses the policy gradient (Sutton et al., 1999) to train an attention-based action policy in saturation-based provers, and Fawzi et al. (2019) applies deep Q-learning (Mnih et al., 2013) to guide the choice of inference rules in a semi-algebraic proof system (Lovász & Schrijver, 1991) for polynomial inequalities.

Most tactic-based models in ITPs use beam search to sample multiple tactic predictions per step with breadth-first (Bansal et al., 2019), depth-first (Yang & Deng, 2019), or best-first heuristics (Polu & Sutskever, 2020) to traverse the search space. In particular, GPT-f (Polu & Sutskever, 2020) and FMSCL (Polu et al., 2023) train language models with outcome and proof size objectives as value functions to perform the best-first search. Besides these methods, Whalen (2016); Mo et al. (2020); Gauthier (2020); Wu et al. (2021a); Gauthier (2021); Lample et al. (2022); Wang et al. (2023a); Brandfonbrener et al. (2024) combine MCTS or use RL to train and guide the search procedure. For example, TacticZero (Wu et al., 2021a) employs the policy gradient to jointly learn tactic prediction and proof search, HTPS (Lample et al., 2022) adopts an AlphaZero (Silver et al., 2018)-like approach with online training, and DT-Solver (Wang et al., 2023a) improves MCTS with dynamic tree sampling and proof-level value function. Additionally, COPRA (Thakur et al., 2024) implements a language-agent method that uses GPT-4 to perform a backtracking search based on the proof history, and TrialMaster (An et al., 2024) finetunes LLMs with trial-and-error data to do backtracking.

3.5 Other Tasks

In addition to the primary tasks outlined previously, we briefly list several other prediction tasks that are related or helpful to theorem proving. One prominent line of research (Clark et al., 2020; Saha et al., 2020; Dalvi et al., 2021; Tafjord et al., 2021; Sanyal et al., 2022; Bostrom et al., 2022; Hong et al., 2022; Mishra et al., 2022; Yang et al., 2022; Tafjord et al., 2022; Morishita et al., 2023; Saparov & He, 2023) focuses on generating step-by-step rationale of a hypothesis from a set of known facts, which answers an open-domain question. These works primarily perform FOL rule deduction over natural language, in the form of an entailment tree, an analogy to the proof tree in formal theorem proving. Another area of interest (Urban & Jakubův, 2020; Piotrowski & Urban, 2020b; Rabe et al., 2021; Johansson & Smallbone, 2023; Bengio & Malkin, 2024; Poesia et al., 2024) is automated conjecturing, which aims to discover new and interesting theorems beyond existing data. Additionally, research efforts such as Huang et al. (2019); Glorot et al. (2019); Polu et al. (2023) predict the proof length remained for a goal. Lee et al. (2020); Wu & Wu (2021) investigate proving theorems in the latent space. IsarStep (Li et al., 2021a) predicts the intermediate proposition given surrounding proofs. Skip-tree (Rabe et al., 2021) and PACT (Han et al., 2022) propose several self-supervised tasks by masking various proof terms for training language models. LIME (Wu et al., 2021c) creates three synthetic pre-training tasks inspired by three reasoning primitives of deduction, induction, and abduction, to improve the performance of formal theorem proving. Recently, Li et al. (2023) proposes to match the informal proofs with theorem statements from a large database, REFACTOR (Zhou et al., 2024b) and ATG (Lin et al., 2024c) focus on extracting or generating new useful formal theorems from proofs.
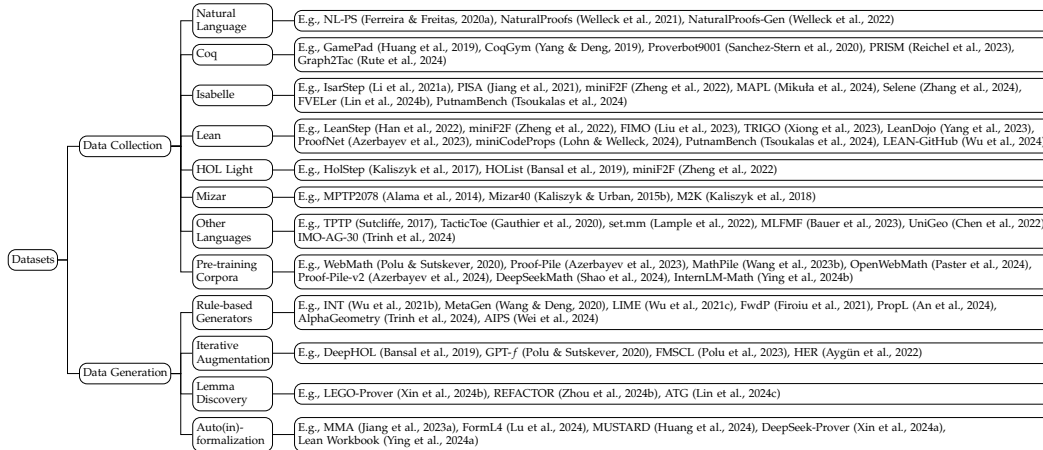
Figure 3: The taxonomy of datasets in theorem proving.

4 Datasets

This section classifies and summarizes datasets for theorem proving into 2 categories: i) datasets extracted from existing corpora or manually curated and ii) those using synthetic generation or augmentation methods. The taxonomy of these datasets is shown in Figure 3, and Table 1 in the Appendix provides more detailed information.

4.1 Data Collection

We begin with the review of informal datasets. NL-PS (Ferreira & Freitas, 2020a) first builds a natural language premise selection dataset source from ProofWiki. Similarly, NaturalProofs (Welleck et al., 2021) further incorporates theorems from Stacks, and textbooks for premise selection. Adapted from NaturalProofs, NaturalProofs-Gen (Welleck et al., 2022) utilizes a subset of theorems and proofs for informal proof generation.

For formal datasets, a line of efforts has been made to extract and clean theorems and proofs from established formal libraries and verification projects. Notable datasets for Coq include GamePad (Huang et al., 2019), CoqGym (Yang & Deng, 2019), Proverbot9001 (Sanchez-Stern et al., 2020), PRISM (Reichel et al., 2023), and Graph2Tac (Rute et al., 2024), which are constructed based on mathematical or software verification projects. For Isabelle, datasets like IsarStep (Li et al., 2021a), PISA (Jiang et al., 2021), and MAPL (Mikuła et al., 2024) are built on the Archive of Formal Proofs and Isabelle Standard Library, while Selene (Zhang et al., 2024) and FVELer (Lin et al., 2024b) are constructed based on the seL4 project (Klein et al., 2009). LeanStep (Han et al., 2022), LeanDojo (Yang et al., 2023), MLFMF (Bauer et al., 2023), and LEAN-GitHub (Wu et al., 2024) utilize Lean's open-source libraries (e.g., the mathlib library (mathlib Community, 2020)). Datasets for other proof assistants include Hol-Step (Kaliszyk et al., 2017) and HOList (Bansal et al., 2019) for HOL Light, MPTP2078 (Alama et al., 2014), Mizar40 (Kaliszyk & Urban, 2015b), and M2K (Kaliszyk et al., 2018) for Mizar, etc. Besides extracting data from existing projects, several works manually formalize or annotate the problems into formal languages: miniF2F (Zheng et al., 2022), FIMO (Liu et al., 2023), ProofNet (Azerbayev et al., 2023), and PutnamBench (Tsoukalas et al., 2024) manually formalize high school and college-level problems in mathematical competitions or textbooks in Lean or Isabelle, and miniCodeProps (Lohn & Welleck, 2024) translates Haskell programs into Lean. For other domains, TRIGO (Xiong et al., 2023) formalizes the trigonometric reduction problem in Lean, while UniGeo (Chen et al., 2022) and IMO-AG-30 (Trinh et al., 2024) annotate proof steps for geometry proving problems in their designed formal languages.

On the other hand, a large body of recent studies leverages large-scale online corpora with billions of tokens from informal and formal mathematical data to build the pre-training datasets that could aid in theorem proving. These datasets include WebMath (Polu & Sutskever, 2020), Proof-Pile (Azerbayev et al., 2023), DeepSeekMath (Shao et al., 2024), etc.

4.2 Data Generation

Beyond utilizing existing projects, researchers also study the generation of new theorems and proofs. A line of work (Wu et al., 2021b;c; Firoiu et al., 2021; Trinh et al., 2024; An et al., 2024; Wei et al., 2024) develops rule-based generators to produce new data by iteratively sampling inference rules from a pre-defined set. This process enables manual control over both the quantity and difficulty of the generated theorems. For example, INT (Wu et al., 2021b) produces 1.5 million inequality training problems with different proof lengths and axiom distributions than the testing ones, while AlphaGeometry (Trinh et al., 2024) synthesizes over 100 million training data with proof lengths ranging from 1 to 247. MetaGen (Wang & Deng, 2020) also trains a neural generator to produce theorems similar to human-write ones.

Alternative approaches turn to iteratively augment the training dataset with fixed theorems but newly generated proofs. A line of work (Bansal et al., 2019; Polu & Sutskever, 2020; Polu et al., 2023) adopts the idea of expert iteration (Silver et al., 2018), which repeatedly applies the trained prover on existing theorems and adds the successful proof paths as new data points to train the prover further. Aygün et al. (2022) also proposes to adapt hindsight experience replay (Andrychowicz et al., 2017) to FOL provers, which leverages previously unsuccessful proof trajectories by viewing their final states as the desired ones. Meanwhile, a line of RL-based methods can also be viewed in this category.

Additionally, some studies aim to generate intermediate helpful lemmas. REFACTOR (Zhou et al., 2024b) trains a GNN to extract lemmas from proofs, which can be used to streamline the proofs of other theorems. Similarly, ATG (Lin et al., 2024c) generates new theorems based on MCTS and self-play learning to shorten proofs. LEGO-Prover (Xin et al., 2024b) prompts GPT-4 to generate sub-goal lemmas for the proof of a theorem and finalize it by proving or retrieving these lemmas. Newly proven lemmas are added to a library for future use. Works on conjecturing are also relevant to this field, although the generated conjectures could be more challenging than existing theorems and may be incorrect or difficult to prove.

Moreover, several recent approaches leverage auto(in)formalization for data generation. Using GPT-4, MMA (Jiang et al., 2023a) informalizes all theorem statements in Archive of Formal Proofs and mathlib, while FormL4 (Lu et al., 2024) generates informal descriptions of both theorems and proofs extracted from mathlib. MUSTARD (Huang et al., 2024) synthesizes problems from a few sampled seed concepts, crafts informal proofs, and translates them into Lean to verify their correctness. DeepSeek-Prover (Xin et al., 2024a) autoformalizes mathematical competition problems and filters high-quality statements through model scoring and hypothesis rejection methods. Similarly, Lean Workbook (Ying et al., 2024a) proposes an active learning pipeline that autoformalizes mathematical questions and filters them through Lean's compilation, LLMs' evaluation, and human diagnostics at each round.

5 Evaluations

In this section, we focus on the evaluations of deep learning approaches in theorem proving, analyzing the key metrics and state-of-the-art performance for each task. Detailed accuracies of state-of-the-art methods on several key datasets are provided in Table 2 in the Appendix.

Autoformalization. The assessment of autoformalization mainly relies on manually checking the equivalence between informal and formalized statements. Recent studies (Wu et al., 2022; Azerbayev et al., 2023) indicate that state-of-the-art LLMs with few-shot prompting can only correctly formalize 25% and 13% high-school and undergraduate-level problems, highlighting the challenge in autoformalization. In addition, both studies reveal that LLMs exhibit considerably higher efficacy in informalization, achieving accuracies of around 76% and 62%, respectively. Despite the modest success in autoformalizing statements, subsequent research (Jiang et al., 2023b; Zhao et al., 2024; Xin et al., 2024b) reveals the benefit of autoformalizing intermediate sub-goals to generate formal proofs in modularized pipelines.

Premise Selection. Retrieval metrics are widely used as the evaluation metric for premise selection. For example, recall at k (R@k) measures the ratio of correctly used premises in ground-truth proof within the top-k selections, and the mean reciprocal rank (MRR)

computes the average reciprocal rank of the first correctly selected premise. Among existing methods, DPR with a Transformer encoder has significantly improved upon traditional methods, demonstrating a remarkable generalization ability to unseen data for premise selection. For example, ReProver (Yang et al., 2023) achieves 27.6% for R@10 and 0.24 for MRR on the LeanDojo benchmark for retrieving unseen premises in training, while the baseline method BM25 (Robertson et al., 2009) achieves 15.5% for R@10 and 0.14 for MRR. Furthermore, a DPR-based retriever, Magnushammer (Mikuła et al., 2024), outperforms the classic hammer system, Sledgehammer (Böhme & Nipkow, 2010), when used to select premises for the Thor prover (Jiang et al., 2022), improving the theorem proving success rate from 57% to 71% on the PISA dataset and from 28.3% to 36.9% on the miniF2F-valid dataset.

Theorem Proving. The effectiveness of proofstep generation, proof search, and support from autoformalization and premise selection can be collectively evaluated by their success rate in proving theorems within a test set. Recent research (Zheng et al., 2024; Xin et al., 2024b) shows impressive performance increases using state-of-the-art LLMs like GPT-4 in structured frameworks than fine-tuned tactic-based language models with search heuristics. For instance, LEGO-Prover (Xin et al., 2024b) achieves 57.0% accuracy and 50.0% accuracy on the valid and test set of miniF2F, while the previous best prover Thor (Jiang et al., 2022) with expert iteration training (Wu et al., 2022) achieves 37.3% and 35.2%. Moreover, advanced learning-based proof searches could further improve the performance of ITP/ATP systems. HTPS (Lample et al., 2022) achieves an accumulative successful rate of 58.6% on miniF2F-valid through online training and a 41.0% success rate with 64 search attempts on miniF2F-test. Besides, the RL-based ATP system NIAGRA (Fokoue et al., 2023) outperforms both E (Schulz, 2002) and Vampire (Kovács & Voronkov, 2013) on the MPTP2078 dataset.

Caveats. Tasks related to theorem proving can be tricky to evaluate. For autoformalization, manual evaluation is costly, whereas most automated metrics are inaccurate. For example, the compilation rate evaluates only syntactic correctness, and the BLEU score (Papineni et al., 2002) struggles with formalizations semantically similar but not logically equivalent to the ground truth. Although LeanEuclid (Murphy et al., 2024) provides automatic semantic evaluation between the autoformalized theorem statement and the ground truth, its domain is limited to Euclidean geometry, making it difficult to generalize to other areas. For premise selection, metrics rely on the premises that are used in ground-truth proofs, so they may neglect other valid premises for alternative correct proofs different from the ground-truth ones, causing false negatives. Additionally, the evaluation of theorem proving is further complicated by the variety of experimental setups, as detailed in §6.1.

6 Discussions

6.1 Challenges

Despite significant progress, deep learning for theorem proving still faces many challenges, including data scarcity, disunified evaluation protocols, and human-AI interaction.

Data Scarcity. The amount of formal proof data is growing but is still far behind other domains where LLMs are successful, e.g., code generation. The largest corpora of Isabelle proofs, Archive of Formal Proofs, currently contains 250K proofs. Lean's mathlib contains 140K proofs. This amount of data is decent for small models (e.g., billions of parameters) but insufficient for models with hundreds of billions of parameters. Although the use of rule-based generators could offer some assistance, the complexity and quality of the generated data often do not match that of human-written ones. Furthermore, autoformalization is even more data-scarce, due to the difficulty in obtaining aligned informal-formal pairs.

Evaluation. Compared to traditional deep learning tasks such as classification, it is considerably more complex to evaluate the performance of theorem provers comprehensively. Firstly, results across different proof assistants are not directly comparable. Even though miniF2F is available across multiple proof assistants, the impact of proof automation tools (e.g., Isabelle has Sledgehammer (Böhme & Nipkow, 2010) whereas Lean does not) often outweighs the differences attributable to deep learning models. Secondly, resource constraints during evaluation, such as time and the number of attempts, can significantly affect performance

and the relative ranking of different methods. While more attempts may favor larger models like LLMs, tight time constraints, especially in real-time applications, may favor simpler, faster models. Without a specific application as the context, it is unclear what evaluation setting makes the most sense. Thirdly, the use of pre-trained LLMs like GPT-4 may introduce data contamination issues (Rute et al., 2024), as these models might have been pre-trained on existing informal or formal testing problems, leading to unfair comparisons. We as a community still lack a systematic evaluation framework across different types of neural theorem provers, despite nascent efforts (Lamont et al., 2024; Rute et al., 2024).

Human-AI Interaction. One motivation for theorem proving is to assist human mathematicians (Castelvecchi, 2021; Sørensen et al., 2021). However, existing research has led to surprisingly few tools useful for them. Current methods are evaluated following standard deep learning protocols: running neural networks or querying LLMs in a Python program, testing the prover on a dataset, and calculating the percentage of successfully proved ones. This practice is misaligned with the needs of mathematicians. First, mathematicians need a tool that can be called easily in a proof assistant. Second, the tool must run on consumer CPUs with low latency. Third, instead of a performance measure, mathematicians care more about whether the prover can help with the specific theorems they are working on. These theorems are often out of the training distribution and pose a challenge for the prover to generalize to other domains. Although initial efforts have been made to develop user-oriented tools (Welleck & Saha, 2023; Song et al., 2024; Rute et al., 2024), there is abundant room to improve the user experience and explore other forms of interaction, which requires close collaboration between deep learning researchers and mathematicians (Collins et al., 2023).

6.2 Future Directions

Combining deep learning, especially LLMs, with theorem proving provides a promising avenue for enhancing the mathematical capabilities of AI and may significantly impact various disciplines. We conclude our survey paper by listing a few future directions we are particularly excited about, envisioning significant strides in these burgeoning domains:

Conjecturing. Beyond merely proving theorems, mathematicians would always explore theories in a domain, identify underlying problem structures, and formulate new conjectures. These explorative activities around conjecturing are indispensable for mathematicians but are relatively limited in current deep learning approaches (Urban & Jakubüv, 2020; Johansson & Smallbone, 2023; Bengio & Malkin, 2024). By enabling AI to generate useful conjectures, it can explore the space of mathematics autonomously. When combined with theorem proving, such exploration can also be used to discover new mathematical knowledge. Furthermore, a direct application of conjecturing is to generate more theorem (and proof) data, which could mitigate the data scarcity inherent in theorem proving.

Verified Code Generation. As AI coding assistants such as GitHub Copilot become prevalent, it is increasingly important to be able to verify LLM-generated code. Proof assistants, especially Coq, have been widely used for software verification (Leroy et al., 2016; Gu et al., 2016). Therefore, methods for theorem proving surveyed in this paper could potentially play a role in generating verified code. There is a plethora of problems to explore in this space. For example, one can train or prompt LLMs to synthesize program invariants (Pei et al., 2023; Kamath et al., 2023) or generate programs in verification-friendly languages such as Dafny (Sun et al., 2023), Verus (Yao et al., 2023), and F* (Chakraborty et al., 2024).

Math Education. A roadblock to democratizing math education is the lack of qualified tutors to provide feedback to students (Kumar et al., 2023). Formal mathematics can potentially mitigate this issue by providing an environment for students to explore and receive automatic and reliable feedback. AI has demonstrated promise in guiding students in this process. For example, Buzzard (2024) reported that Lean Copilot (Song et al., 2024) effectively assisted in proving a bunch of problems in his undergraduate course and solved questions on the Lean Zulip. To further integrate AI-driven formal tutoring into mainstream education, informalization is essential to make formal proofs accessible to students unfamiliar with formal languages. Looking ahead, we believe that theorem proving with LLMs will pave the way for intelligent tutors, enhancing math education for a broader audience.

Acknowledgments

References

Ibrahim Abdelaziz, Veronika Thost, Maxwell Crouse, and Achille Fokoue. An Experimental Study of Formula Embeddings for Automated Theorem Proving in First-Order Logic. *arXiv preprint arXiv:2002.00423*, 2020.

Ibrahim Abdelaziz, Maxwell Crouse, Bassem Makni, Vernon Austel, Cristina Cornelio, Shajith Ikbal, Pavan Kapanipathi, Ndivhuwo Makondo, Kavitha Srinivas, Michael Witbrock, et al. Learning to Guide a Saturation-Based Theorem Prover. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

Ayush Agrawal, Siddhartha Gadgil, Navin Goyal, Ashvni Narayanan, and Anand Tadipatri. Towards a Mathematics Formalisation Assistant using Large Language Models. *arXiv preprint arXiv:2211.07524*, 2022.

Jesse Alama, Tom Heskes, Daniel Kühlwein, Evgeni Tsivtsivadze, and Josef Urban. Premise Selection for Mathematics by Corpus Analysis and Kernel Methods. *Journal of Automated Reasoning*, 2014.

Chenyang An, Zhibo Chen, Qihao Ye, Emily First, Letian Peng, Jiayun Zhang, Zihan Wang, Sorin Lerner, and Jingbo Shang. Learn from Failure: Fine-Tuning LLMs with Trial-and-Error Data for Intuitionistic Propositional Logic Proving. *arXiv preprint arXiv:2404.07382*, 2024.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2017.

Eser Aygün, Zafarali Ahmed, Ankit Anand, Vlad Firoiu, Xavier Glorot, Laurent Orseau, Doina Precup, and Shibl Mourad. Learning to Prove from Synthetic Theorems. *arXiv preprint arXiv:2006.11259*, 2020.

Eser Aygün, Ankit Anand, Laurent Orseau, Xavier Glorot, Stephen M Mcaleer, Vlad Firoiu, Lei M Zhang, Doina Precup, and Shibl Mourad. Proving Theorems using Incremental Learning and Hindsight Experience Replay. In *Proceedings of the International Conference on Machine Learning*, 2022.

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W Ayers, Dragomir Radev, and Jeremy Avigad. ProofNet: Autoformalizing and Formally Proving Undergraduate-Level Mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An Open Language Model for Mathematics. In *Proceedings of the International Conference on Learning Representations*, 2024.

Kshitij Bansal and Christian Szegedy. Learning Alignment between Formal & Informal Mathematics. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2020.

Kshitij Bansal, Sarah M. Loos, Markus Norman Rabe, Christian Szegedy, and Stewart Wilcox. HOList: An Environment for Machine Learning of Higher Order Logic Theorem Proving. In *Proceedings of the International Conference on Machine Learning*, 2019.

Haniel Barbosa, Clark Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, et al. CVC5: A Versatile and Industrial-Strength SMT Solver. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2022.

Bruno Barras, Samuel Boutin, Cristina Cornes, Judicaël Courant, Yann Coscoy, David Delahaye, Daniel de Rauglaudre, Jean-Christophe Filliâtre, Eduardo Giménez, Hugo Herbelin, et al. The Coq Proof Assistant Reference Manual. *INRIA*, 1999.

Filip Bártek and Martin Suda. How Much Should This Symbol Weigh? A GNN-Advised Clause Selection. In *Proceedings of International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, 2023.

Andrej Bauer, Matej Petković, and Ljupco Todorovski. MLFMF: Data Sets for Machine Learning for Mathematical Formalization. In *Proceedings of the International Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Yoshua Bengio and Nikolay Malkin. Machine Learning and Information Theory Concepts Towards an AI Mathematician. *Bulletin of the American Mathematical Society*, 2024.

Armin Biere. CaDiCaL at the SAT Race 2019. In *Proceedings of SAT Race 2019 – Solver and Benchmark Descriptions*, 2019.

Jasmin Christian Blanchette, Cezary Kaliszyk, Lawrence C Paulson, and Josef Urban. Hammering Towards QED. *Journal of Formalized Reasoning*, 2016.

Sascha Böhme and Tobias Nipkow. Sledgehammer: Judgement Day. In *Proceedings of the International Joint Conference on Automated Reasoning*, 2010.

Kaj Bostrom, Zayne Sprague, Swarat Chaudhuri, and Greg Durrett. Natural Language Deduction through Search over Statement Compositions. In *Findings of the Association for Computational Linguistics: EMNLP*, 2022.

David Brandfonbrener, Sibi Raja, Tarun Prasad, Chloe Loughridge, Jianang Yang, Simon Henniger, William E Byrd, Robert Zinkov, and Nada Amin. Verified Multi-Step Synthesis using Large Language Models and Monte Carlo Tree Search. *arXiv preprint arXiv:2402.08147*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

N.G. Bruijn, de. The Mathematical Language Automath, Its Usage and Some of Its Extensions. In *Proceedings of the Symposium on Automatic Demonstration*, 1970.

Kevin Buzzard. Lean in 2024. https://xenaproject.wordpress.com/2024/01/20/lean-in-2024/, 2024.

Davide Castelvecchi. Mathematicians Welcome Computer-Assisted Proof in 'Grand Unification' Theory. *Nature*, 2021.

Saikat Chakraborty, Gabriel Ebner, Siddharth Bhat, Sarah Fakhoury, Sakina Fatima, Shuvendu Lahiri, and Nikhil Swamy. Towards Neural Synthesis for SMT-Assisted Proof-Oriented Programming. *arXiv preprint arXiv:2405.01787*, 2024.

Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. UniGeo: Unifying Geometry Logical Reasoning via Reformulating Mathematical Expression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the International Conference on Machine Learning*, 2020.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. A Deductive Database Approach to Automated Geometry Theorem Proving and Discovering. *Journal of Automated Reasoning*, 2000.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, 2023.

Karel Chvalovskỳ, Jan Jakubův, Martin Suda, and Josef Urban. ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E. In *Proceedings of the International Conference on Automated Deduction*, 2019.

Karel Chvalovskỳ, Jan Jakubův, Miroslav Olšák, and Josef Urban. Learning Theorem Proving Components. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 2021.

Karel Chvalovskỳ, Konstantin Korovin, Jelle Piepenbrock, and Josef Urban. Guiding an Instantiation Prover with Graph Neural Networks. In *Proceedings of International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, 2023.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as Soft Reasoners over Language. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2020.

Katherine M Collins, Albert Q Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B Tenenbaum, William Hart, et al. Evaluating Language Models for Mathematics through Interactions. *arXiv preprint arXiv:2306.01694*, 2023.

Maxwell Crouse, Ibrahim Abdelaziz, Cristina Cornelio, Veronika Thost, Lingfei Wu, Kenneth Forbus, and Achille Fokoue. Improving Graph Neural Network Representations of Logical Formulae with Subgraph Pooling. *arXiv preprint arXiv:1911.06904*, 2019.

Maxwell Crouse, Ibrahim Abdelaziz, Bassem Makni, Spencer Whitehead, Cristina Cornelio, Pavan Kapanipathi, Kavitha Srinivas, Veronika Thost, Michael Witbrock, and Achille Fokoue. A Deep Reinforcement Learning Approach to First-Order Logic Theorem Proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Garett Cunningham, Razvan C Bunescu, and David Juedes. Towards Autoformalization of Mathematics and Code Correctness: Experiments with Elementary Proofs. In *Proceedings of the Workshop on Mathematical Natural Language Processing*, 2022.

Łukasz Czajka and Cezary Kaliszyk. Hammer for Coq: Automation for Dependent Type Theory. *Journal of automated reasoning*, 2018.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining Answers with Entailment Trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021.

Doratossadat Dastgheib and Ehsaneddin Asgari. Keyword-Based Natural Language Premise Selection for an Automatic Mathematical Statement Proving. In *Proceedings of TextGraphs-16: Graph-Based Methods for Natural Language Processing*, 2022.

Martin Davis. A Computer Program for Presburger's Algorithm. *Symbolic Computation Automation of Reasoning 1*, 1957.

Martin Davis and Hilary Putnam. A Computing Procedure for Quantification Theory. *Journal of the ACM*, 1960.

Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2008.

Jörg Denzinger, Matthias Fuchs, Christoph Goller, and Stephan Schulz. Learning from Previous Proof Experience: A Survey. *Technical Report*, 1999.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2015.

Niklas Eén and Niklas Sörensson. An Extensible SAT-Solver. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing*, 2003.

Alhussein Fawzi, Mateusz Malinowski, Hamza Fawzi, and Omar Fawzi. Learning Dynamic Polynomial Proofs. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2019.

Deborah Ferreira and André Freitas. Natural Language Premise Selection: Finding Supporting Statements for Mathematical Text. In *Proceedings of the Language Resources and Evaluation Conference*, 2020a.

Deborah Ferreira and André Freitas. Premise Selection in Natural Language Mathematical Texts. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020b.

Deborah Ferreira and André Freitas. STAR: Cross-Modal [STA]tement [R]epresentation for Selecting Relevant Mathematical Premises. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistic*, 2021.

Vlad Firoiu, Eser Aygun, Ankit Anand, Zafarali Ahmed, Xavier Glorot, Laurent Orseau, Lei Zhang, Doina Precup, and Shibl Mourad. Training a First-Order Theorem Prover from Synthetic Data. In *International Conference on Learning Representations Workshop on Mathematical Reasoning in General Artificial Intelligence*, 2021.

Emily First and Yuriy Brun. Diversity-Driven Automated Formal Verification. In *Proceedings of the International Conference on Software Engineering*, 2022.

Emily First, Yuriy Brun, and Arjun Guha. TacTok: Semantics-Aware Proof Synthesis. *Proceedings of the ACM on Programming Languages*, 2020.

Emily First, Markus Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-Proof Generation and Repair with Large Language Models. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023.

Achille Fokoue, Ibrahim Abdelaziz, Maxwell Crouse, Shajith Ikbal, Akihiro Kishimoto, Guilherme Lima, Ndivhuwo Makondo, and Radu Marinescu. An Ensemble Approach for Automated Theorem Proving Based on Efficient Name Invariant Graph Neural Representations. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2023.

Simon Frieder, Julius Berner, Philipp Petersen, and Thomas Lukasiewicz. Large Language Models for Mathematicians. *arXiv preprint arXiv:2312.04556*, 2023a.

Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, and Julius Berner. Mathematical Capabilities of ChatGPT. In *Proceedings of the International Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023b.

Simon Frieder, Martin Trimmel, Rashid Alawadhi, and Klaus Gy. LLM vs ITP. In *International Conference on Neural Information Processing Systems Workshop on MATH-AI*, 2023c.

Siddhartha Gadgil, Anand Rao Tadipatri, Ayush Agrawal, Ashvni Narayanan, and Navin Goyal. Towards Automating Formalisation of Theorem Statements using Large Language Models. In *International Conference on Neural Information Processing Systems Workshop on MATH-AI*, 2022.

Thibault Gauthier. Deep Reinforcement Learning for Synthesizing Functions in Higher-Order Logic. In *Proceedings of the International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, 2020.

Thibault Gauthier. Learned Provability Likelihood for Tactical Search. In *Proceedings of the International Symposium on Symbolic Computation in Software Science*, 2021.

Thibault Gauthier, Cezary Kaliszyk, and Josef Urban. TacticToe: Learning to Prove with Tactics. *Journal of Automated Reasoning*, 2020.

Fabian Gloeckle, Baptiste Roziere, Amaury Hayat, and Gabriel Synnaeve. Temperature-Scaled Large Language Models for Lean Proofstep Prediction. In *International Conference on Neural Information Processing Systems Workshop on MATH-AI*, 2023.

Xavier Glorot, Ankit Anand, Eser Aygun, Shibl Mourad, Pushmeet Kohli, and Doina Precup. Learning Representations of Logical Formulae using Graph Neural Networks. In *International Conference on Neural Information Processing Systems Workshop on Graph Representation Learning*, 2019.

Zarathustra A Goertzel and Josef Urban. Usefulness of Lemmas via Graph Neural Networks. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2019.

Zarathustra A Goertzel, Karel Chvalovský, Jan Jakubův, Miroslav Olšák, and Josef Urban. Fast and Slow ENIGMAs and Parental Guidance. In *Proceedings of the International Symposium on Frontiers of Combining Systems*, 2021.

Zarathustra A Goertzel, Jan Jakubův, Cezary Kaliszyk, Miroslav Olšák, Jelle Piepenbrock, and Josef Urban. The Isabelle ENIGMA. In *Proceedings of the International Conference on Interactive Theorem Proving*, 2022.

Georges Gonthier. The Four Colour Theorem: Engineering of a Formal Proof. In *Proceedings of the Asian Symposium on Computer Mathematics*, 2008.

Ronghui Gu, Zhong Shao, Hao Chen, Xiongnan Newman Wu, Jieung Kim, Vilhelm Sjöberg, and David Costanzo. CertiKOS: An Extensible Architecture for Building Certified Concurrent OS Kernels. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, 2016.

Thomas Hales, Mark Adams, Gertrud Bauer, Tat Dat Dang, John Harrison, Hoang Le Truong, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Tat Thang Nguyen, et al. A Formal Proof of the Kepler Conjecture. In *Forum of Mathematics, Pi*, 2017.

Jesse Michael Han, Tao Xu, Stanislas Polu, Arvind Neelakantan, and Alec Radford. Contrastive Finetuning of Generative Language models for Informal Premise Selection. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2021.

Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W Ayers, and Stanislas Polu. Proof Artifact Co-Training for Theorem Proving with Language Models. In *Proceedings of the International Conference on Learning Representations*, 2022.

John Harrison. HOL Light: A Tutorial Introduction. In *Proceedings of the International Conference on Formal Methods in Computer-Aided Design*, 1996.

Edvard K Holden and Konstantin Korovin. Graph Sequence Learning for Premise Selection. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2023.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. MetGen: A Module-Based Entailment Tree Generation Framework for Answer Explanation. In *Findings of the Association for Computational Linguistics: NAACL*, 2022.

Daniel Huang, Prafulla Dhariwal, Dawn Song, and Ilya Sutskever. GamePad: A Learning Environment for Theorem Proving. In *Proceedings of the International Conference on Learning Representations*, 2019.

Yinya Huang, Xiaohan Lin, Zhengying Liu, Qingxing Cao, Huajian Xin, Haiming Wang, Zhenguo Li, Linqi Song, and Xiaodan Liang. MUSTARD: Mastering Uniform Synthesis of Theorem and Proof Data. In *Proceedings of the International Conference on Learning Representations*, 2024.

Geoffrey Irving, Christian Szegedy, Alexander A Alemi, Niklas Eén, François Chollet, and Josef Urban. DeepMath - Deep Sequence Models for Premise Selection. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2016.

Jan Jakubüv and Josef Urban. ENIGMA: Efficient Learning-Based Inference Guiding Machine. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2017.

Jan Jakubüv and Josef Urban. Hammering Mizar by Learning Clause Guidance. In *Proceedings of the International Conference on Interactive Theorem Proving*, 2019.

Jan Jakubüv, Karel Chvalovskỳ, Miroslav Olšák, Bartosz Piotrowski, Martin Suda, and Josef Urban. ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine (System Description). In *Proceedings of the International Joint Conference on Automated Reasoning*, 2020.

Jan Jakubüv, Karel Chvalovskỳ, Zarathustra A Goertzel, Cezary Kaliszyk, Mirek Olšák, Bartosz Piotrowski, Stephan Schulz, Martin Suda, and Josef Urban. MizAR 60 for Mizar 50. In *Proceedings of the International Conference on Interactive Theorem Proving*, 2023.

Albert Q Jiang, Wenda Li, Jesse Michael Han, and Yuhuai Wu. LISA: Language Models of Isabelle Proofs. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2021.

Albert Q Jiang, Wenda Li, Szymon Tworkowski, Konrad Czechowski, Tomasz Odrzygóźdź, Piotr Miłoś, Yuhuai Wu, and Mateja Jamnik. Thor: Wielding Hammers to Integrate Language Models and Automated Theorem Provers. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

Albert Q Jiang, Wenda Li, and Mateja Jamnik. Multilingual Mathematical Autoformalization. *arXiv preprint arXiv:2311.03755*, 2023a.

Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs. In *Proceedings of the International Conference on Learning Representations*, 2023b.

Dongwei Jiang, Marcio Fonseca, and Shay B Cohen. LeanReasoner: Boosting Complex Logical Reasoning with Lean. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

Moa Johansson and Nicholas Smallbone. Exploring Mathematical Conjecturing with Large Language Models. In *Proceedings of the International Workshop on Neural-Symbolic Learning and Reasoning*, 2023.

Cezary Kaliszyk and Josef Urban. FEMaLeCoP: Fairly Efficient Machine Learning Connection Prover. In *Proceedings of the International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, 2015a.

Cezary Kaliszyk and Josef Urban. MizAR 40 for Mizar 40. *Journal of Automated Reasoning*, 2015b.

Cezary Kaliszyk, Josef Urban, Jiří Vyskočil, and Herman Geuvers. Developing Corpus-Based Translation Methods between Informal and Formal Mathematics: Project Description. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2014.

Cezary Kaliszyk, Josef Urban, and Jiří Vyskočil. Learning to Parse on Aligned Corpora (Rough Diamond). In *Proceedings of the International Conference on Interactive Theorem Proving*, 2015.

Cezary Kaliszyk, François Chollet, and Christian Szegedy. HolStep: A Machine Learning Dataset for Higher-Order Logic Theorem Proving. In *Proceedings of the International Conference on Learning Representations*, 2017.

Cezary Kaliszyk, Josef Urban, Henryk Michalewski, and Miroslav Olšák. Reinforcement Learning of Theorem Proving. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2018.

Adharsh Kamath, Aditya Senthilnathan, Saikat Chakraborty, Pantazis Deligiannis, Shuvendu K Lahiri, Akash Lal, Aseem Rastogi, Subhajit Roy, and Rahul Sharma. Finding Inductive Loop Invariants using Large Language Models. *arXiv preprint arXiv:2311.07948*, 2023.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020.

Christoph Kern and Mark R Greenstreet. Formal Verification in Hardware Design: A Survey. *ACM Transactions on Design Automation of Electronic Systems*, 1999.

Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, et al. seL4: Formal Verification of an OS Kernel. In *Proceedings of the ACM SIGOPS Symposium on Operating Systems Principles*, 2009.

Konstantin Korovin. iProver–An Instantiation-Based Theorem Prover for First-Order Logic (System Description). In *Proceedings of the International Joint Conference on Automated Reasoning*, 2008.

Laura Kovács and Andrei Voronkov. First-Order Theorem Proving and Vampire. In *Proceedings of the International Conference on Computer Aided Verification*, 2013.

Liubov Kovriguina, Roman Teucher, and Robert Wardenga. TextGraphs-16 Natural Language Premise Selection Task: Zero-Shot Premise Selection with Prompting Generative Language Models. In *Proceedings of TextGraphs-16: Graph-Based Methods for Natural Language Processing*, 2022.

Andrzej Stanisław Kucik and Konstantin Korovin. Premise Selection with Neural Networks and Distributed Representation of Features. *arXiv preprint arXiv:1807.10268*, 2018.

Daniel Kühlwein, Twan van Laarhoven, Evgeni Tsivtsivadze, Josef Urban, and Tom Heskes. Overview and Evaluation of Premise Selection Techniques for Large Theory Mathematics. In *Proceedings of the International Joint Conference on Automated Reasoning*, 2012.

Harsh Kumar, David M Rothschild, Daniel G Goldstein, and Jake M Hofman. Math Education with Large Language Models: Peril or Promise? *SSRN Electronic Journal*, 2023.

Mitsuru Kusumoto, Keisuke Yahata, and Masahiro Sakai. Automated Theorem Proving in Intuitionistic Propositional Logic by Deep Reinforcement Learning. *arXiv preprint arXiv:1811.00796*, 2018.

Sean Lamont, Michael Norrish, Amir Dezfouli, Christian Walder, and Paul Montague. BAIT: Benchmarking (Embedding) Architectures for Interactive Theorem-Proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Guillaume Lample and Alexis Conneau. Cross-Lingual Language Model Pretraining. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2019.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-Based & Neural Unsupervised Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. Hypertree Proof Search for Neural Theorem Proving. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

Dennis Lee, Christian Szegedy, Markus N Rabe, Sarah M Loos, and Kshitij Bansal. Mathematical Reasoning in Latent Space. In *Proceedings of the International Conference on Learning Representations*, 2020.

Xavier Leroy, Sandrine Blazy, Daniel Kästner, Bernhard Schommer, Markus Pister, and Christian Ferdinand. CompCert–A Formally Verified Optimizing Compiler. In *Proceeding of the European Congress on Embedded Real Time Software and Systems*, 2016.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving Quantitative Reasoning Problems with Language Models. *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

Weixian Waylon Li, Yftah Ziser, Maximin Coavoux, and Shay B Cohen. BERT Is Not The Count: Learning to Match Mathematical Statements with Proofs. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 2023.

Wenda Li, Lei Yu, Yuhuai Wu, and Lawrence C Paulson. IsarStep: A Benchmark for High-Level Mathematical Reasoning. In *Proceedings of the International Conference on Learning Representations*, 2021a.

Zhaoyu Li, Binghong Chen, and Xujie Si. Graph Contrastive Pre-Training for Effective Theorem Reasoning. *International Conference on Machine Learning Workshop on Self-Supervised Learning for Reasoning and Perception*, 2021b.

Zhenwen Liang, Tianyu Yang, Jipeng Zhang, and Xiangliang Zhang. UniMath: A Foundational and Multimodal Mathematical Reasoner. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

Haohan Lin, Zhiqing Sun, Yiming Yang, and Sean Welleck. Lean-STaR: Learning to Interleave Thinking and Proving. *arXiv preprint arXiv:2407.10040*, 2024a.

Qika Lin, Jun Liu, Lingling Zhang, Yudai Pan, Xin Hu, Fangzhi Xu, and Hongwei Zeng. Contrastive Graph Representations for Logical Formulas Embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Xiaohan Lin, Qingxing Cao, Yinya Huang, Haiming Wang, Jianqiao Lu, Zhengying Liu, Linqi Song, and Xiaodan Liang. FVEL: Interactive Formal Verification Environment with Large Language Models via Theorem Proving. *arXiv preprint arXiv:2406.14408*, 2024b.

Xiaohan Lin, Qingxing Cao, Yinya Huang, Zhicheng Yang, Zhengying Liu, Zhenguo Li, and Xiaodan Liang. ATG: Benchmarking Automated Theorem Generation for Generative Language Models. In *Findings of the Association for Computational Linguistics: NAACL*, 2024c.

Chengwu Liu, Jianhao Shen, Huajian Xin, Zhengying Liu, Ye Yuan, Haiming Wang, Wei Ju, Chuanyang Zheng, Yichun Yin, Lin Li, et al. FIMO: A Challenge Formal Dataset for Automated Theorem Proving. *arXiv preprint arXiv:2309.04295*, 2023.

Qinghua Liu, Yang Xu, and Xingxing He. Attention Recurrent Cross-Graph Neural Network for Selecting Premises. *International Journal of Machine Learning and Cybernetics*, 2022a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhou Liu, Yujun Li, Zhengying Liu, Lin Li, and Zhenguo Li. Learning to Prove Trigonometric Identities. *arXiv preprint arXiv:2207.06679*, 2022b.

Evan Lohn and Sean Welleck. miniCodeProps: A Minimal Benchmark for Proving Code Properties. *arXiv preprint arXiv:2406.11915*, 2024.

Sarah Loos, Geoffrey Irving, Christian Szegedy, and Cezary Kaliszyk. Deep Network Guided Proof Search. In *Proceedings of the International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, 2017.

László Lovász and Alexander Schrijver. Cones of Matrices and Set-Functions and 0–1 Optimization. *SIAM journal on optimization*, 1991.

Jianqiao Lu, Zhengying Liu, Yingjia Wan, Yinya Huang, Haiming Wang, Zhicheng Yang, Jing Tang, and Zhijiang Guo. Process-Driven Autoformalization in Lean 4. *arXiv preprint arXiv:2406.01940*, 2024.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. Neural Machine Translation (seq2seq) Tutorial. *https://github.com/tensorflow/nmt*, 2017.

The mathlib Community. The Lean Mathematical Library. In *Proceedings of the ACM SIGPLAN International Conference on Certified Programs and Proofs*, 2020.

Jack McKeown and Geoff Sutcliffe. Reinforcement Learning for Guiding the E Theorem Prover. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference*, 2023.

Norman Megill and David A Wheeler. *Metamath: A Computer Language for Mathematical Proofs*. Lulu Press, 2019.

Maciej Mikuła, Szymon Antoniak, Szymon Tworkowski, Albert Q Jiang, Jin Peng Zhou, Christian Szegedy, Łukasz Kuciński, Piotr Miłoś, and Yuhuai Wu. Magnushammer: A Transformer-Based Approach to Premise Selection. In *Proceedings of the International Conference on Learning Representations*, 2024.

Robin Milner. Implementation and Applications of Scott's Logic for Computable Functions. In *Proceedings of the ACM Conference on Proving Assertions about Programs*, 1972.

Bhavana Dalvi Mishra, Oyvind Tafjord, and Peter Clark. Towards Teachable Reasoning Systems: Using a Dynamic Memory of User Feedback for Continual System Improvement. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*, 2013.

Guangshuai Mo, Yan Xiong, Wenchao Huang, and Lu Ma. Automated Theorem Proving via Interacting with Proof Assistants by Dynamic Strategies. In *Proceedings of the International Conference on Big Data Computing and Communications*, 2020.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Learning Deductive Reasoning from Synthetic Corpus based on Formal Logic. In *Proceedings of the International Conference on Machine Learning*, 2023.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 Theorem Prover and Programming Language. In *Proceedings of the International Conference on Automated Deduction*, 2021.

Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. Autoformalizing Euclidean Geometry. In *Proceedings of the International Conference on Machine Learning*, 2024.

Theo X Olausson, Alex Gu, Benjamin Lipkin, Cedegao E Zhang, Armando Solar-Lezama, Joshua B Tenenbaum, and Roger Levy. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

Miroslav Olšák, Cezary Kaliszyk, and Josef Urban. Property Invariant Embedding for Automated Reasoning. In *Proceedings of the European Conference on Artificial Intelligence*, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jens Otten and Wolfgang Bibel. leanCoP: Lean Connection-Based Theorem Proving. *Journal of Symbolic Computation*, 2003.

Aditya Paliwal, Sarah Loos, Markus Rabe, Kshitij Bansal, and Christian Szegedy. Graph Representations for Higher-Order Logic and Theorem Proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, 2002.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text. In *Proceedings of the International Conference on Learning Representations*, 2024.

Nilay Patel, Jeffrey Flanigan, and Rahul Saha. A New Approach Towards Autoformalization. *arXiv preprint arXiv:2310.07957*, 2023.

Lawrence C Paulson. *Isabelle: A Generic Theorem Prover*. Springer, 1994.

Kexin Pei, David Bieber, Kensen Shi, Charles Sutton, and Pengcheng Yin. Can Large Language Models Reason about Program Invariants? In *Proceedings of the International Conference on Machine Learning*, 2023.

Kebin Peng and Dianfu Ma. Tree-Structure CNN for Automated Theorem Proving. In *Proceedings of the International Conference on Neural Information Processing*, 2017.

Jelle Piepenbrock, Tom Heskes, Mikoláš Janota, and Josef Urban. Learning Equational Theorem Proving. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2021.

Jelle Piepenbrock, Tom Heskes, Mikoláš Janota, and Josef Urban. Guiding An Automated Theorem Prover with Neural Rewriting. In *Proceedings of the International Joint Conference on Automated Reasoning*, 2022a.

Jelle Piepenbrock, Josef Urban, Konstantin Korovin, Miroslav Olšák, Tom Heskes, and Mikoláš Janota. Machine Learning Meets The Herbrand Universe. *arXiv preprint arXiv:2210.03590*, 2022b.

Bartosz Piotrowski and Josef Urban. Stateful Premise Selection by Recurrent Neural Networks. In *Proceedings of the International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, 2020a.

Bartosz Piotrowski and Josef Urban. Guiding Inferences in Connection Tableau by Recurrent Neural Networks. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2020b.

Gabriel Poesia, David Broman, Nick Haber, and Noah D Goodman. Learning formal mathematics from intrinsic motivation. *arXiv preprint arXiv:2407.00695*, 2024.

Auguste Poiroux, Gail Weiss, Viktor Kunčak, and Antoine Bosselut. Improving Autoformalization using Type Checking. *arXiv preprint arXiv:2406.07222*, 2024.

Stanislas Polu and Ilya Sutskever. Generative Language Modeling for Automated Theorem Proving. *arXiv preprint arXiv:2009.03393*, 2020.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal Mathematics Statement Curriculum Learning. In *Proceedings of the International Conference on Learning Representations*, 2023.

Seth Poulsen, Sami Sarsa, James Prather, Juho Leinonen, Brett A Becker, Arto Hellas, Paul Denny, and Brent N Reeves. Solving Proof Block Problems Using Large Language Models. In *Proceedings of the ACM Technical Symposium on Computer Science Education*, 2024.

Krsto Proroković, Michael Wand, and Jürgen Schmidhuber. Improving Stateful Premise Selection with Transformers. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2021.

Xin Quan, Marco Valentino, Louise A Dennis, and André Freitas. Verification and Refinement of Natural Language Explanations through LLM-Symbolic Theorem Proving. *arXiv preprint arXiv:2405.01379*, 2024.

Markus N Rabe, Dennis Lee, Kshitij Bansal, and Christian Szegedy. Mathematical Reasoning via Self-Supervised Skip-Tree Training. In *Proceedings of the International Conference on Learning Representations*, 2021.

Michael Rawson and Giles Reger. A Neurally-Guided, Parallel Theorem Prover. In *Proceedings of the International Symposium on Frontiers of Combining Systems*, 2019.

Michael Rawson and Giles Reger. Directed Graph Networks for Logical Reasoning (Extended Abstract). In *Joint Proceedings of the Workshop on Practical Aspects of Automated Reasoning (PAAR) and the Satisfiability Checking and Symbolic Computation Workshop (SC-Square) Workshop*, 2020.

Michael Rawson and Giles Reger. lazyCoP: Lazy Paramodulation Meets Neurally Guided Search. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 2021.

Tom Reichel, R Henderson, Andrew Touchet, Andrew Gardner, and Talia Ringer. Proof Repair Infrastructure for Supervised Models: Building a Large Proof Repair Dataset. In *Proceedings of the International Conference on Interactive Theorem Proving*, 2023.

Stephen Robertson, Hugo Zaragoza, et al. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 2009.

Piotr Rudnicki. An Overview of the Mizar Project. In *Proceedings of the Workshop on Types for Proofs and Programs*, 1992.

Jason Rute, Miroslav Olšák, Lasse Blaauwbroek, Fidel Ivan Schaposnik Massolo, Jelle Piepenbrock, and Vasily Pestun. Graph2Tac: Online Representation Learning of Formal Math Concepts. In *Proceedings of the International Conference on Machine Learning*, 2024.

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. PRover: Proof Generation for Interpretable Reasoning over Rules. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020.

Alex Sanchez-Stern, Yousef Alhessi, Lawrence Saul, and Sorin Lerner. Generating Correctness Proofs with Neural Networks. In *Proceedings of the ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 2020.

Alex Sanchez-Stern, Emily First, Timothy Zhou, Zhanna Kaufman, Yuriy Brun, and Talia Ringer. Passport: Improving Automated Formal Verification Using Identifiers. *ACM Transactions on Programming Languages and Systems*, 2023.

Soumya Sanyal, Harman Singh, and Xiang Ren. FaiRR: Faithful and Robust Deductive Reasoning over Natural Language. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022.

Abulhair Saparov and He He. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought. In *Proceedings of the International Conference on Learning Representations*, 2023.

Gregor vom Scheidt. Experimental Results from Applying GPT-4 to An Unpublished Formal Language. *arXiv preprint arXiv:2305.12196*, 2023.

Stephan Schulz. E–A Brainiac Theorem Prover. *AI Communications*, 2002.

Johann M Schumann. *Automated Theorem Proving in Software Engineering*. Springer Science & Business Media, 2001.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Boris Shminke. gym-saturation: Gymnasium Environments for Saturation Provers (System description). In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 2023.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go Through Self-Play. *Science*, 2018.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: Masked and Permuted Pre-Training for Language Understanding. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

Peiyang Song, Kaiyu Yang, and Anima Anandkumar. Towards Large Language Models as Copilots for Theorem Proving in Lean. *arXiv preprint arXiv:2404.12534*, 2024.

Henrik Kragh Sørensen, Mikkel Willum Johansen, Renee Hoekzema, and Hester Breman. Augmenting the Human Mathematician. In *International Conference on Learning Representations Workshop on Mathematical Reasoning in General Artificial Intelligence*, 2021.

Martin Suda. Vampire with a Brain Is a Good ITP Hammer. In *Proceedings of the International Symposium on Frontiers of Combining Systems*, 2021a.

Martin Suda. Improving ENIGMA-Style Clause Selection While Learning From History. In *Proceedings of the International Conference on Automated Deduction*, 2021b.

Chuyue Sun, Ying Sheng, Oded Padon, and Clark Barrett. Clover: Closed-Loop Verifiable Code Generation. *arXiv preprint arXiv:2310.17807*, 2023.

Geoff Sutcliffe. The TPTP Problem Library and Associated Infrastructure. *Journal of Automated Reasoning*, 2017.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2014.

Christian Suttner and Wolfgang Ertel. Automatic Acquisition of Search Guiding Heuristics. In *Proceedings of the International Conference on Automated Deduction*, 1990.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Proceedings of the International Conference on Neural Information Processing Systems*, 1999.

Christian Szegedy. A Promising Path Towards Autoformalization and General Artificial Intelligence. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2020.

Christian Szegedy, Markus Rabe, and Henryk Michalewski. Retrieval-Augmented Proof Step Synthesis. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2021.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, 2021.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. Entailer: Answering Questions with Faithful and Truthful Chains of Reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 2015.

Amitayush Thakur, George Tsoukalas, Yeming Wen, Jimmy Xin, and Swarat Chaudhuri. An In-Context Learning Agent for Formal Theorem-Proving. *arXiv preprint arXiv:2310.04353*, 2024.

Thi Hong Hanh Tran, Matej Martinc, Antoine Doucet, and Senja Pollak. IJS at TextGraphs-16 Natural Language Premise Selection Task: Will Contextual Information Improve Natural Language Premise Selection? In *Proceedings of TextGraphs-16: Graph-Based Methods for Natural Language Processing*, 2022.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving Olympiad Geometry without Human Demonstrations. *Nature*, 2024.

Paul Trust, Provia Kadusabe, Haseeb Younis, Rosane Minghim, Evangelos Milios, and Ahmed Zahran. UNLPS at TextGraphs 2022 Shared Task: Unsupervised Natural Language Premise Selection in Mathematical Texts Using Sentence-MPNet. In *Proceedings of TextGraphs-16: Graph-Based Methods for Natural Language Processing*, 2022.

George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. PutnamBench: Evaluating Neural Theorem-Provers on the Putnam Mathematical Competition. *arXiv preprint arXiv:2407.11214*, 2024.

Szymon Tworkowski, Maciej Mikuła, Tomasz Odrzygóźdź, Konrad Czechowski, Szymon Antoniak, Albert Jiang, Christian Szegedy, Łukasz Kuciński, Piotr Miłoś, and Yuhuai Wu. Formal Premise Selection With Language Models. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2022.

Josef Urban and Jan Jakubův. First Neural Conjecturing Datasets and Experiments. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2020.

Josef Urban, Jiří Vyskočil, and Petr Štěpánek. MaLeCoP Machine Learning Connection Prover. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 2011.

Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations*, 2018.

Rahul Vishwakarma and Subhankar Mishra. Enhancing Neural Theorem Proving through Data Augmentation and Dynamic Sampling Method. *arXiv preprint arXiv:2312.14188*, 2023.

Haiming Wang, Ye Yuan, Zhengying Liu, Jianhao Shen, Yichun Yin, Jing Xiong, Enze Xie, Han Shi, Yujun Li, Lin Li, et al. DT-Solver: Automated Theorem Proving with Dynamic-Tree Sampling Guided by Proof-level Value Function. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2023a.

Haiming Wang, Huajian Xin, Zhengying Liu, Wenda Li, Yinya Huang, Jianqiao Lu, Zhicheng Yang, Jing Tang, Jian Yin, Zhenguo Li, et al. Proving Theorems Recursively. *arXiv preprint arXiv:2405.14414*, 2024.

Mingzhe Wang and Jia Deng. Learning to Prove Theorems by Learning to Generate Theorems. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. Premise Selection for Theorem Proving by Deep Graph Embedding. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2017.

Qingxiang Wang, Cezary Kaliszyk, and Josef Urban. First Experiments with Neural Translation of Informal to Formal Mathematics. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2018.

Qingxiang Wang, Chad Brown, Cezary Kaliszyk, and Josef Urban. Exploration of Neural Machine Translation in Autoformalization of Mathematics in Mizar. In *Proceedings of the ACM SIGPLAN International Conference on Certified Programs and Proofs*, 2020.

Zengzhi Wang, Rui Xia, and Pengfei Liu. Generative AI for Math: Part I–MathPile: A Billion-Token-Scale Pretraining Corpus for Math. *arXiv preprint arXiv:2312.17120*, 2023b.

Chenrui Wei, Mengzhou Sun, and Wei Wang. Proving Olympiad Algebraic Inequalities without Human Demonstrations. *arXiv preprint arXiv:2406.14219*, 2024.

Sean Welleck and Rahul Saha. LLMSTEP: LLM Proofstep Suggestions in Lean. In *International Conference on Neural Information Processing Systems Workshop on MATH-AI*, 2023.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. NaturalProofs: Mathematical Theorem Proving in Natural Language. In *Proceedings of the International Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. NaturalProver: Grounded Mathematical Proof Generation with Language Models. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

Daniel Whalen. Holophrasm: A Neural Automated Theorem Prover for Higher-Order Logic. *arXiv preprint arXiv:1608.02644*, 2016.

Minchao Wu and Yuhuai Wu. Latent Action Space for Efficient Planning in Theorem Proving. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2021.

Minchao Wu, Michael Norrish, Christian Walder, and Amir Dezfouli. TacticZero: Learning to Prove Theorems from Scratch with Deep Reinforcement Learning. *Proceedings of the International Conference on Neural Information Processing Systems*, 2021a.

Yuhuai Wu, Albert Q Jiang, Jimmy Ba, and Roger Grosse. INT: An Inequality Benchmark for Evaluating Generalization in Theorem Proving. In *Proceedings of the International Conference on Learning Representations*, 2021b.

Yuhuai Wu, Markus N Rabe, Wenda Li, Jimmy Ba, Roger B Grosse, and Christian Szegedy. LIME: Learning Inductive Bias for Primitives of Mathematical Reasoning. In *Proceedings of the International Conference on Machine Learning*, 2021c.

Yuhuai Wu, Albert Q Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with Large Language Models. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

Zijian Wu, Jiayu Wang, Dahua Lin, and Kai Chen. Lean-github: Compiling github lean repositories for a versatile lean prover. *arXiv preprint arXiv:2407.17227*, 2024.

Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. DeepSeek-Prover: Advancing Theorem Proving in LLMs through Large-Scale Synthetic Data. *arXiv preprint arXiv:2405.14333*, 2024a.

Huajian Xin, Haiming Wang, Chuanyang Zheng, Lin Li, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, et al. LEGO-Prover: Neural Theorem Proving with Growing Libraries. In *Proceedings of the International Conference on Learning Representations*, 2024b.

Jing Xiong, Jianhao Shen, Ye Yuan, Haiming Wang, Yichun Yin, Zhengying Liu, Lin Li, Zhijiang Guo, Qingxing Cao, Yinya Huang, et al. TRIGO: Benchmarking Formal Mathematical Proof Reduction for Generative Language Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *Proceedings of the International Conference on Learning Representations*, 2019.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*, 2022.

Kaiyu Yang and Jia Deng. Learning to Prove Theorems via Interacting with Proof Assistants. In *Proceedings of the International Conference on Machine Learning*, 2019.

Kaiyu Yang, Jia Deng, and Danqi Chen. Generating Natural Language Proofs with Verifier-Guided Search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2023.

Jianan Yao, Ziqiao Zhou, Weiteng Chen, and Weidong Cui. Leveraging Large Language Models for Automated Proof Synthesis in Rust. *arXiv preprint arXiv:2311.03739*, 2023.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. SATLM: Satisfiability-Aided Language Models using Declarative Prompting. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2023.

Eric Yeh, Briland Hitaj, Sam Owre, Maena Quemener, and Natarajan Shankar. CoProver: A Recommender System for Proof Construction. In *Proceedings of the International Conference on Intelligent Computer Mathematics*, 2023.

Huaiyuan Ying, Zijian Wu, Yihan Geng, Jiayu Wang, Dahua Lin, and Kai Chen. Lean Workbook: A Large-Scale Lean Problem Set Formalized from Natural Language Math Problems. *arXiv preprint arXiv:2406.03847*, 2024a.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, et al. InternLM-Math: Open Math Large Language Models Toward Verifiable Reasoning. *arXiv preprint arXiv:2402.06332*, 2024b.

Roozbeh Yousefzadeh and Xuenan Cao. Large Language Models' Understanding of Math: Source Criticism and Extrapolation. *arXiv preprint arXiv:2311.07618*, 2023.

Liao Zhang, Lasse Blaauwbroek, Cezary Kaliszyk, and Josef Urban. Learning Proof Transformations and Its Applications in Interactive Theorem Proving. In *Proceedings of the International Symposium on Frontiers of Combining Systems*, 2023a.

Lichen Zhang, Shuai Lu, and Nan Duan. Selene: Pioneering Automated Proof in Software Verification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2024.

Shizhuo Dylan Zhang, Talia Ringer, and Emily First. Getting More out of Large Language Models for Proofs. In *Proceedings of the Conference on Artificial Intelligence and Theorem Proving*, 2023b.

Xueliang Zhao, Wenda Li, and Lingpeng Kong. Subgoal-Based Demonstration Learning for Formal Theorem Proving. In *Proceedings of the International Conference on Machine Learning*, 2024.

Chuanyang Zheng, Haiming Wang, Enze Xie, Zhengying Liu, Jiankai Sun, Huajian Xin, Jianhao Shen, Zhenguo Li, and Yu Li. Lyra: Orchestrating Dual Correction in Automated Theorem Proving. *Transactions on Machine Learning Research*, 2024.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. MiniF2F: A Cross-System Benchmark for Formal Olympiad-Level Mathematics. In *Proceedings of the International Conference on Learning Representations*, 2022.

Jin Peng Zhou, Charles E Staats, Wenda Li, Christian Szegedy, Kilian Q Weinberger, and Yuhuai Wu. Don't Trust: Verify–Grounding LLM Quantitative Reasoning with Autoformalization. In *Proceedings of the International Conference on Learning Representations*, 2024a.

Jin Peng Zhou, Yuhuai Wu, Qiyang Li, and Roger Grosse. REFACTOR: Learning to Extract Theorems from Proofs. In *Proceedings of the International Conference on Learning Representations*, 2024b.

Zsolt Zombori, Adrián Csiszárik, Henryk Michalewski, Cezary Kaliszyk, and Josef Urban. Towards Finding Longer Proofs. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 2021a.

Zsolt Zombori, Josef Urban, and Miroslav Olšák. The Role of Entropy in Guiding a Connection Prover. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 2021b.

A Lists of Datasets and State-of-the-Art Approaches

Dataset	Language[1]	Size[2]	Source	Task[3]
NL-PS (Ferreira & Freitas, 2020a)	NL	20,401	ProofWiki	PS$_1$
NaturalProofs (Welleck et al., 2021)	NL	48,783	ProofWiki, Stacks, textbooks	PS$_1$
NaturalProofs-Gen (Welleck et al., 2022)	NL	~33K	NaturalProofs	PG
GamePad (Huang et al., 2019)	Coq	1,602	Feit-Thompson theorem	PG
CoqGym (Yang & Deng, 2019)	Coq	70,856	123 open-source projects from Github	TP
Proverbot9001 (Sanchez-Stern et al., 2020)	Coq	501*	CompCert	TP
Graph2Tac (Rute et al., 2024)	Coq	~520K	120 open-source packages from opam	TP
PISA (Jiang et al., 2021)	Isabelle	~183K	Achieve of Formal Proofs, Isabelle standard library	TP
MAPL (Mikuła et al., 2024)	Isabelle	~433K	Achieve of Formal Proofs, Isabelle standard library	PS$_1$
Selene (Zhang et al., 2024)	Isabelle	360*	seL4	TP
FVELer (Lin et al., 2024b)	Isabelle	29,125	seL4	TP
LeanStep (Han et al., 2022)	Lean	~20K	Lean core library, mathlib	TP
TRIGO (Xiong et al., 2023)	Lean	427	trigonometry problems from tiku	TP
LeanDojo (Yang et al., 2023)	Lean	98,734	mathlib	TP
MLFMF (Bauer et al., 2023)	Lean, Agda	270,647	mathlib, Agda standard library, UniMath, TypeTopology	PS$_1$
miniCodeProps (Lohn & Welleck, 2024)	Lean, Haskell	177*	Tons of Inductive Programs	TP
LEAN-GitHub (Wu et al., 2024)	Lean	28,597	147 repositories on the web	TP
FIMO (Liu et al., 2023)	NL, Lean	149*	IMO Shortlisted Problems	AF
ProofNet (Azerbayev et al., 2023)	NL, Lean	371*	undergraduate textbooks	AF
miniF2F (Zheng et al., 2022)	NL, Lean, Isabelle, Metamath, HOL Light	488*	IMO, AIME, AMC, MATH, custom	TP
PutnamBench (Tsoukalas et al., 2024)	NL, Lean, Isabelle	640*	William Lowell Putnam Mathematical Competition	TP
HolStep (Kaliszyk et al., 2017)	HOL Light	11,410	multivariate analysis library & Kepler conjecture	PS$_1$
HOList (Bansal et al., 2019)	HOL Light	31,662	Kepler conjecture	TP
MPTP2078 (Alama et al., 2014)	Mizar, TPTP	2,078	Mizar Mathematical Library	PS$_1$, PS$_2$
Mizar40 (Kaliszyk & Urban, 2015b)	Mizar, TPTP	~58k	Mizar Mathematical Library	PS$_1$, PS$_2$
M2K (Kaliszyk et al., 2018)	Mizar, TPTP	2,003	Mizar40	PS$_2$
TPTP (Sutcliffe, 2017)	TPTP	11,395	ATP system evaluation	PS$_2$
TacticToe (Gauthier et al., 2020)	HOL4	7,164*	HOL4 standard library	TP
set.mm (Lample et al., 2022)	Metamath	37,091	set.mm	TP
UniGeo (Chen et al., 2022)	NL, DSL	9,543	high school geometry problems from IXL	PG
IMO-AG-30 (Trinh et al., 2024)	DSL	30*	IMO geometry problems	TP
INT (Wu et al., 2021b)	DSL	1,501K	synthetic inequality problems	TP
AlphaGeometry (Trinh et al., 2024)	DSL	~100M	synthetic geometry problems	TP
MMA (Jiang et al., 2023a)	NL, Lean, Isabelle	332,774	autoinformalization from Achieve of Formal Proofs and mathlib	AF
FormL4 (Lu et al., 2024)	NL, Lean	17,461	autoinformalization from mathlib, Arithmo test set	AF
MUSTARD (Huang et al., 2024)	NL, Lean	5,866	autoformalization from synthetic problems	TP
DeepSeek-Prover (Xin et al., 2024a)	NL, Lean	8,066,621	autoformalization from competition problems	TP
Lean Workbook (Ying et al., 2024a)	NL, Lean	~57K	autoformalization from Art of Problem Solving	AF
WebMath (Polu & Sutskever, 2020)	mixed	35B	GitHub, arXiv, Stack Exchange	PT
Proof-Pile (Azerbayev et al., 2023)	mixed	8.3B	arXiv, Stack Exchange, formal libraries, ProofWiki, Wikipedia, books, MATH	PT
MathPile (Wang et al., 2023b)	mixed	9.5B	arXiv, textbooks, Stack Exchange, Wikipedia, ProofWiki, Web	PT
OpenWebMath (Paster et al., 2024)	mixed	14.7B	forum posts, educational content, reference pages, scientific papers, blogs, and others	PT
Proof-Pile-v2 (Azerbayev et al., 2024)	mixed	55B	Github, Stack, formal proof steps, OpenWebMath, arXiv	PT
DeepSeekMath (Shao et al., 2024)	mixed	120B	OpenWebMath, Web	PT
InternLM-Math (Ying et al., 2024b)	mixed	125B	Knowledge Pile, Proof-Pile-v2, synthetic data	PT

Table 1: Summary of existing datasets for theorem proving. For datasets derived from generation methods, we primarily highlight several key examples here.

Language	Dataset	Best Accuracy (%)	Evaluation Metric	Method
Coq	CoqGym	33.8	timeout of 600s	Diva (First & Brun, 2022) + CoqHammer (Czajka & Kaliszyk, 2018)
Lean	LeanDojo (random)	57.7	pass@1 + timeout of 600s	temperature scaling (Gloeckle et al., 2023)
	ProofNet (valid/test)	49.2/53.2	top-1 accuracy@50	type checking (Poiroux et al., 2024)
	miniF2F (valid/test)	60.2/46.3	cumulative/pass@64	DeepSeek-Prover (Xin et al., 2024a)
Isabelle	miniF2F (valid/test)	57.0/51.2	pass@100/pass@200	LEGO-Prover (Xin et al., 2024b)/Lyra (Zheng et al., 2024)
	PISA	71.0	pass@300 + timeout of 500s	Magnushammer (Mikuła et al., 2024)+Thor(Jiang et al., 2022)
Metamath	set.mm (valid/test)	81.2/72.4	pass@32	HTPS (Lample et al., 2022)
Mizar	MPTP2078	75.5	timeout of 100s	NIAGRA (Fokoue et al., 2023)

Table 2: Summary of state-of-the-art methods and their performance on several key datasets.

---

[1]NL: natural language. DSL: domain-specific language.

[2]We use the number of tokens for pre-training datasets and the number of entries (e.g., definitions, premises, theorems) for other tasks. "*" indicates the size of the testing/validation set.

[3]We list the main task of each dataset. AF: autoformalization. PS$_1$: premise selection. PG: proofstep generation. PS$_2$: proof search. PT: pre-training. TP: theorem proving (combination of premise selection, proofstep generation, and proof search).