arXiv:2303.06686v1 [cs.AI] 12 Mar 2023

Mizar 50 的 MizAR 60

JanJakubův ！
捷克，布拉格，布拉格捷克理工大学
KarelChvalovský ！捷克，布拉格，布拉格捷克理工大学
Zarathustra Goertzel
捷克，布拉格，布拉格捷克理工大学
CezaryKaliszyk ！因斯布鲁克大学，奥地利和 INDRC，布拉格，捷克
Mirek Olšák
高等科学研究所，巴黎，法国
Bartosz Piotrowski 布拉格捷克理工大学，布拉格，捷克
Stephan Schulz
巴登-符腾堡双元制应用技术大学斯图加特校区，斯图加特，德国
Martin Suda
布拉格捷克理工大学，布拉格，捷克
Josef Urban
布拉格捷克理工大学，布拉格，捷克

摘要

作为献给 Mizar 诞生 50 周年的礼物，我们开发了一种人工智能/定理证明系统，可在 hammer 设置下自动证明约 60% 的 Mizar 定理。当自动证明器仅使用人类编写的 Mizar 证明中所用的前提时，我们还能自动证明 75% 的 Mizar 定理。我们介绍了促成这些成果的方法和大规模实验，尤其包括 E Prover 和吸血鬼证明器、它们的 ENIGMA 和 Deepire 学习改进版、多种基于学习的前提选择方法，以及一种增量式循环：在不断扩充包含数百万条 ATP 证明的语料库的同时，训练越来越强大的人工智能/定理证明系统。我们还展示了一组选自自动证明的 Mizar 问题。

2012年 ACM 学科分类计算理论 → 逻辑 → 自动推理

关键词与短语Mizar、ENIGMA、自动推理、机器学习

1 引言：Mizar、MML、hammer 系统与 AITP

近年来，结合机器学习（ML）、人工智能（AI）和自动定理证明（ATP） [45] 的方法取得了长足发展，其主要目标是 ITP 社区开发的大型形式化数学库。这些方法涵盖前提选择方法 [2] 、*hammer* [7] 系统，以及针对从 ITP 库中提取的数千至数百万个问题，开发并训练基于学习的自动定理证明系统内部引导，例如 E [48, 50] 和 Vampire [37] 。如此庞大的 ITP 语料库还推动了自动策略发明 [58]、战术指导 [15], 、基于学习的猜想生成 [59], 、自动形式化 [34, 62], 等研究课题的发展，以及结合学习与推理、通过各种反馈回路运行的元系统的开发 [60]。

从 MPTP 系统于2003年3月发布[1] 以及基于该系统开展首批机器学习/定理证明和自动证明器实验 [55] 开始，Mizar 数学库 [3, 4, 22]（MML）及其

[1] http://mizar.uwb.edu.pl/forum/archive/0303/msg00004.html

子集截至2023年已在这项研究中使用了二十年，这或许使其成为过去二十年里历史最悠久、研究最深入的人工智能/定理证明资源。

1.1 贡献

上一次针对 MML 开展的大规模 *Mizar40* 评估，是在将近十年前、Mizar 成立 40 周年之际进行的。此后，前提选择和自动定理证明器的内部引导等领域发展出了许多强大的方法。因此，在这项工作中，我们以一种可与 Mizar40 评估进行比较的方式评估这些方法，从而全面展示该领域取得了多大进展。我们的主要成果如下：

1. 如今，当证明所需的前提可以由人工或机器从文库中选出时，超过 75% 的 Mizar 顶层引理可以由人工智能/定理证明系统证明。这应与 Mizar40 在同一版本 MML 上达到的 56 % 相比较。我们在网页上分析了超过 200 个自动获得的证明示例。[2]2. 如今，58.4% 的 Mizar 顶层引理无需用户提供任何帮助即可得到证明，即处于大理论（锤击）模式下。这应与 Mizar40 在同一版本 MML 上达到的约 40.6% 相比较。在这两种情况下，使用的都是由大量人工智能/定理证明方法组成的方法组合，并将 CPU 时间限制为 420 s。3. 仅凭我们最强的单个人工智能/定理证明方法，如今在 30 s 内即可在锤击模式下证明 40% 的引理，即达到 Mizar40 中完整 420 s 方法组合的同等能力。4. 我们最强的单个人工智能/定理证明方法如今在 120 s 内即可在人类前提（分支）模式（第6.6节）下证明 60% 的顶层引理，即超过 Mizar40 中开发的所有方法的并集（56%）。5. 我们展示了最强方法可以迁移到一个明显更新的 MML 版本，该版本包含大量新术语和引理。特别是在 MML 版本 1382 中新增的 242 篇文章所包含的 13 370 个定理上，我们最强的方法比标准 E Prover 高出 58.2%，而在用于训练和实验的 Mizar40 文库版本上，这一数值仅为 56.1%。这得益于我们开发并使用了匿名的 [25] 逻辑感知机器学习方法；这些方法只从数学问题的结构中学习。这在如今的机器学习中并不常见，因为该领域由大型语言模型主导，而大型语言模型通常难以应对新术语。

1.2 方法与实验概述

本次评估的核心方法是 ENIGMA（后来也包括 Deepire）系统提供的内部引导，以及前提选择方法。我们还采用了若干其他方法，例如多种先前提出的策略及其组合构建的新方法、针对数百万条 ATP 证明进行大规模训练的高效方法、交替执行多轮自动定理证明器运行，并在每轮重新启动时使用基于机器学习的方法选择最佳推断子句的方法（蛙跳），以及通过将问题分解为多个自动定理证明子问题来最小化解决问题所需前提的方法。这些方法将在第2节介绍 MML 后，于第3、4、5节中加以说明。第6节介绍大规模评估及其最终结果，第7节则展示所得证明。

2 Mizar 数学文库与 Mizar40 语料库

证明助手系统通常会与各自的证明库一同开发。这样可以对现有功能进行评估并展示。对于 Mizar [4]，

[2] https://github.com/ai4reason/ATP_Proofs

开发者很早就决定专注于其文库 MML（Mizar 数学库） [3]。为此，他们成立了专门的文库委员会，负责评估可能纳入的 Mizar文章，并维护该文库。因此，MML 成为了当今最大的形式化数学文库之一。其中包含许多其他系统推导出的结果中所没有的成果，例如格 [5] 和随机访问图灵机 [36]。

本文收集的所有数据和开展的所有评估（第6.6节中的版本迁移除外）均使用与先前的大规模评估 [32] 以及之后所有评估论文相同的 Mizar 库版本。这使我们能够严格比较各方法并评估改进效果。该版本的库 MML 1147 通过 MPTP导出 [57] 导出为一阶逻辑后，共对应 57 897 个定理，其中包括未命名的顶层引理。为了在锤击场景中进行严格评估，我们将在第6.2节中进一步把该数据集划分为若干训练集和测试集。

3 ENIGMA：自动定理证明器引导及相关技术

ENIGMA [11, 18–21、25、27–29] 是“高效学习型推理引导机器”（Efficient Learning Based Inference Guiding Machine）的缩写。它是首个学习引导的自动定理证明器，于2019年在当时最先进的饱和型自动定理证明器 [29], 之上取得了大幅改进，也是本文所报告工作的主要组成部分。本节总结此前发表的有关 ENIGMA 的研究，以及为开展本文的大规模实验（第6节）而采用的相关方法。

3.1 饱和定理证明遇上机器学习

饱和证明器： 最先进的自动定理证明器（如 E Prover [46] 和 Vampire [37], ）会搜索矛盾：首先将输入的一阶逻辑问题转换为一个反驳等价子句集。然后，证明器使用给定子句算法进行证明搜索。在该算法中，证明状态被分为两个子集： P 已处理子句集和 U 未处理子句集。 U 中的子句按照启发式评估函数排序。在主循环的每次迭代中，选取 U 中（根据启发式评估的）最佳子句。随后，这个给定子句 g 会根据 P中的所有子句进行简化。如果它不是冗余的，则用它依次简化 P中的所有子句。之后，在 g 与 P 中的其余子句之间执行所有生成推理。新生成的子句以及 P 中的简化子句随后根据 P进行完全简化，经过启发式评估后添加到 U中。该过程会持续进行，直到空子句出现（或系统耗尽资源）。

训练数据： 截至 E 1.8 [49], ，E 维护一个内部证明对象 [51] ，使其能够检查所有证明子句，并将所有已被选中处理且属于证明的子句标记为正训练示例。所有已被选中处理但未对证明作出贡献的子句，则标记为负训练示例。完全未被处理的子句既不是正例也不是负例，从而将训练示例总数通常减少到数千个已处理子句，而不是所生成的数百万个子句。E 允许用户请求实际的证明对象，或提供任意组合的正训练示例和负训练示例。示例以单独的批次提供，并且也会标注为正例或负例，以便于处理。

基于机器学习的选择： 在 E 中，选择正确的给定子句至关重要，也是应用机器学习（ML）的一个显而易见的切入点。正例和负例从以往成功的证明搜索中提取，并训练机器学习模型，为生成子句评分，或将其分类为有用（⊞）或无用（⊟）。E Prover 选择给定

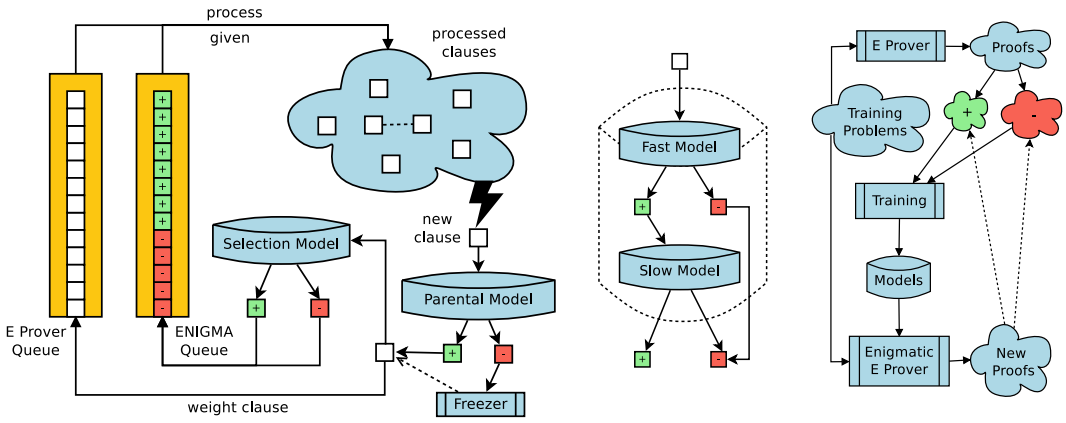

图 1 带有 ENIGMA 的 E Prover（左）、两阶段选择模型（中）以及证明-学习反馈循环（右）的示意图。

子句的方式是从优先级队列中选取，其中未处理子句按照各种启发式方法排序。ENIGMA 通过增加一个队列扩展了 E Prover，在该队列中，机器学习模型判定为正例的子句具有更高优先级。ENIGMA 队列与标准的 E 选择机制结合使用，通常采用协作方式，由 ENIGMA 选择大约一半的子句。实践证明，这种方法效率最高。

亲本引导： 后来的 ENIGMA [20] 引入了基于学习的亲本引导，解决了经典基于饱和的证明器在已处理子句之间进行所有 可能推理时所涉及的二次因子问题。相反，训练机器学习模型来阻止那些不太可能发生有意义交互的父子句之间的推理。当模型以较高置信度识别出某个推理没有用时，不会将子子句插入未处理子句集 U，而是推迟对其进行处理。为保持完备性，不能直接丢弃该子句，因为机器学习模型可能会出错。相反，会将该子句放入“冻结区”，以便在证明器耗尽未处理子句时将其取出。与上述子句选择模型不同，这种方法会影响标准的 E 选择机制，因为该子句不会被插入任何队列。ENIGMA 子句选择模型与亲本（生成）模型可以成功结合。这在图 1（左）中以示意图的形式进行了说明。

多阶段 ENIGMA： 基于 ML 的多阶段子句选择在 [20]中被引入，用于处理计算成本高昂的（缓慢）ML 模型，例如图神经网络（GNN）。在两阶段 选择模型中，使用一个更快的 模型进行初步子句筛选，只有通过筛选的子句才会由较慢的模型进行评估。快速模型应在正类上进行过近似，以便仅拒绝那些被高置信度分类为负例的子句。当加入亲本引导后，这便形成了三阶段 ENIGMA。图 1（中）以示意图说明了这一点。激进前向包含 是一种基于 E [47] 中高效包含索引的额外逻辑完备剪枝方法。我们使用它在调用计算成本更高的 ML 方法（GNN）进行子句评估之前，消除许多冗余生成子句。有关此类方法的效果，请参见第7节中的一些三阶段 ENIGMA 示例。

训练： 强大的 ENIGMA 通常通过多次证明-学习反馈循环 [60] 开发而成，具体过程如下。(1) 从（先前）成功的证明搜索中整理训练数据 $\mathcal{T}$。(2) 在数据$\mathcal{T}$上训练模型 $\mathcal{M}$，使其能够区分肯定子句和

否定子句。(3) 将模型 $\mathcal{M}$ 与自动定理证明器（E）一起运行，通常与用于获取训练数据的策略协同使用。然后利用步骤 (3) 中获得的新数据回到步骤 (1)。如图 1（右）所示，该循环可以持续重复，直到证明出新的问题。在本文中，我们运行这一循环数月。

3.2 梯度提升决策树分类器和特征

ENIGMA 支持基于梯度提升决策树（梯度提升决策树）的分类器。具体而言，我们对 XGBoost [8] 和 LightGBM [35]进行了实验。这两个框架效率都很高，在训练和评估方面也都能很好地处理大型数据。对于学习，我们用数值型 特征向量表示一阶子句。决策树是一种二叉树，其节点标记有关于特征向量值的条件。给定一个子句，沿树遍历至相应的叶节点，其中存储着该子句的评估结果。这两个框架都使用由多棵树组成的序列（集成），并以渐进的方式（提升）构建这些树。两个框架在构建决策树所采用的底层算法上有所不同。XGBoost 按层构建树，而 LightGBM 按叶节点构建树。这意味着 XGBoost 的树具有良好的平衡性。另一方面，LightGBM 可以生成深得多的树，因此树深度限制确实是一个重要的学习元参数，可以对其进行优化。

ENIGMA 从一阶子句中提取各种句法信息，并将其存储在该子句的特征向量中。给定一个有限特征集，每个特征都会在特征向量中分配一个索引，相应的特征值则存储在该索引处。例如，典型的子句特征是子句长度。ENIGMA 支持以下特征。**垂直特征** 通过遍历子句语法树并收集所有长度为 3、从上到下的符号路径构建。此外，为了抽象变量名并处理斯科伦符号可能发生的冲突，所有变量都会被替换为特殊名称 $\odot$，所有斯科伦符号都会被替换为 $\circledast$。**水平特征**会针对每个项 $f(t_1, \ldots\ldots, t_n)$引入一个新特征 $f(s_1, \ldots\ldots, s_n)$，其中 s_i是 t_i的顶层符号。**计数特征**包括子句长度、文字计数以及类似的统计信息。**猜想特征** 将待证明的猜想嵌入特征向量中。因此，ENIGMA 能够提供目标特定预测。**父特征** 通过其父项的特征（连接或求和）表示一个子句。**特征哈希** 是 ENIGMA [11]处理大规模数据的重要步骤。它显著减小了特征向量大小，从而能够处理更大规模的数据。每个特征都由唯一的字符串标识符表示。该字符串会经过哈希函数处理，并将哈希值对所选哈希基数 取模，所得结果用作特征索引。**符号匿名化** 允许抽象出具体的符号名称 [25]。在提取子句特征的过程中，所有符号名称都会被符号元数替换，只保留该符号是函数还是谓词的信息。这样一来，决策树分类器便不依赖符号名称，但代价是可能产生符号冲突；不过，通过收集更长的路径作为特征，可以在经验上缓解这一问题。

3.3 图神经网络（GNN）分类器

匿名化图神经网络为摆脱特定术语的抽象提供了一种替代方法。ENIGMA 使用 [25] 一种最初为指导语义表搜索而开发的、与符号无关的 GNN架构 [39] ，该架构在 TensorFlow 中实现 [1]。一组子句可直接表示为超图，其中包含子句、子项/文字和符号这三类节点。对象之间的关系通过各种图边表示，使网络能够区分不同符号，同时抽象掉它们的名称。

GNN 层通过边执行消息传递，因此每个节点上的信息都能传递给其邻居。这使网络能够了解符号的使用方式，而无需知道它们的名称。我们始终将新子句与初始子句一起进行分类，后者为理解匿名化符号的含义提供上下文。在 ATP 评估期间，会在更大的批次中一次性计算数百个生成子句的预测，上下文同时由初始子句和已处理子句提供。上下文可以是固定的，包含初始子句和已处理子句的起始片段；也可以是使用具有最佳 GNN 评估值的子句窗口的上下文移位。

3.4 其他相关技术

GPU 服务器模式 支持使用 GPU 进行实时评估 [20]。为了减少模型加载带来的 GPU 开销，我们开发了一个 Python GPU 服务器，其中预先加载了模型，可以将评估任务分配到多个 GPU 上。E Prover 客户端通过网络套接字与服务器通信。当我们并行运行 160 个 E prover 实例时，可以充分利用物理服务器[3] 。在同一台机器上运行服务器和客户端可以减少网络通信开销。

交替迭代 解决了选择新的给定子句时上下文不断变化的问题 [10]。我们使用给定的抽象限制运行 ENIGMA，并生成更大的子句集。然后，我们对这些生成子句进行前提选择（例如，仅选择已处理子句），提取其中的优质子句，并将其作为新一轮 ENIGMA 运行的输入。相关的 拆分/合并方法会反复将生成子句拆分为多个组件，分别运行后再通过前提选择进行合并。这一方法的灵感来自这样的观点：较难的问题由多个组件构成，而分治方法有助于处理这些组件。

Deepire 是 Vampire [37] 的扩展版本 [52, 53] ，采用机器学习的子句选择引导，整体遵循 ENIGMA 风格的方法。其特点是使用递归神经网络，仅根据生成子句的推导历史对其进行分类。因而，Deepire 并不试图读取“子句表达了什么”，而只根据“子句来自何处”做出决策。这使得子句评估可以特别快速，同时仍能识别并提升有用的子句，尤其是在包含特殊公理的领域中；这些公理会在许多问题中反复出现。

4 从 MML 中学习前提选择

在大型 ITP 库中使用自动定理证明器时，通常只有一小部分事实与证明新猜想相关。由于向自动定理证明器提供过多冗余前提会大幅降低证明该猜想的可能性，因此前提选择是一项关键任务。最高效的前提选择方法采用数据驱动 或机器学习 方法。如果 T 是一个包含定理及其证明的集合，而 C 是一个不含证明的猜想集合，那么任务就是从 T 中学习一个（统计）模型，该模型会针对每个猜想$c \in C$ ，根据可用前提与生成该猜想的自动定理证明器证明相关的程度，对这些前提进行排序（或选取其子集）。有两种主要的机器学习设置可供采用。在**多标签分类**中，证明中使用的前提被视为不透明标签，并训练机器学习模型根据猜想的特征为猜想添加标签。**二元分类** 旨在识别（猜想，前提）对之间的相关性，也就是根据猜想和前提双方的特征，估计某个前提与证明该猜想相关的可能性。

[3] 36 超线程 Intel(R) Xeon(R) Gold 6140 中央处理器 @ 2.30 GHz 核心、755 GB 内存，以及 4 个 NVIDIA GeForce GTX 1080 Ti GPU。

第一种设置适用于较简单、快速的机器学习方法，例如 k-NN 或朴素贝叶斯——这些方法将在第4.1节中介绍。第二种设置（第4.2节）支持使用更强大的机器学习架构，例如梯度提升决策树和图神经网络（第3.3节和第3.2节）。不过，这种设置还需要为训练 [42]，选择负例，这会增加其复杂性。

4.1 多标签前提选择（$\mathcal{K}$、$\mathcal{N}$、$\mathcal{R}$）

朴素贝叶斯和 k 近邻是 Mizar40 评估 [32] 中最强的选择方法。在本工作中，我们对它们进行了改进，并将其与较新的方法结合使用。

***k*-近邻（$\mathcal{K}$）：** 将 k 近邻算法应用于前提选择时，会在特征空间中选择与猜想最接近的 k 事实，并选取它们的依赖项。标准 k-近邻算法已有的改进包括考虑事实的依赖项数量（依赖项更多的证明更长，因此重要性较低），以及使用词频-逆文档频率（罕见特征更为重要） [30]。此外，我们发现无需固定 k。相反，我们采用一个较小的 k，如果评分依赖项的数量过少，就增加 k 并更新依赖项。该过程会重复进行，直到获得所请求数量的预测结果。基于 k-近邻且使用固定k 的预测结果将记作，例如 $\mathcal{K}_{512}$,；而使用可变 k 时则记作 $\mathcal{K}^{\text{fea}}_{\text{var}}$，其中 fea 指定所使用的特征。

朴素贝叶斯（$\mathcal{N}$）： 稀疏朴素贝叶斯算法通过以下条件概率来估计事实F 的相关性：在猜想中存在这些特征的条件下，F 有用的概率（根据以往的证明统计数据估计）。我们还考虑 F的扩展特征，即 F 的特征，以及使用 F所证明的事实的特征。结合前提选择专用权重后，该方法的效果优于基本的朴素贝叶斯，并且已经应用于 HolyHammer，之后又应用于 Sledgehammer。算法的完整推导见 [6]。朴素贝叶斯预测将表示为$\mathcal{N}_{\text{fea}}$。

这些算法可以通过更复杂的特征进行参数化。我们考虑了： `cp` 用于项图中的常量和路径（第 3.2 节）， `sub` 用于子项， `au` 用于反统一特征 [33]， `eni` 用于第 3.2 节讨论的在线 ENIGMA 特征，以及 `uni` 用于上述所有特征的并集。最后，这些算法还支持按时间顺序模式：在学习阶段，根据 Mizar规范顺序（`MML.LAR`），丢弃使用了当前猜想之后引入的事实的证明。这会使算法的效果略有减弱，但与之前的 Mizar40 前提选择评估 [32]兼容。这些算法将通过 ^{chrono}进行标记。

使用循环神经网络进行依赖选择（$\mathcal{R}$）： 前提选择方法最初主要是根据事实相对于猜想的相关性，独立地对其进行排序。随后，将排名最高的事实作为公理，与猜想一起提供给自动定理证明系统。这类方法（也用于梯度提升决策树）虽然实用且效果良好，但没有考虑到前提之间并非相互独立。在证明特定猜想时，有些前提能够更好地互相补充，而一些排名很高的前提可能只是彼此的细微变体。循环神经网络（RNN）编码器-解码器模型 [9] 和 Transformer [61] （语言模型）事实证明是用于建模这类隐式依赖关系的合适机器学习架构。这类模型传统上用于自然语言处理，不过近年来也越来越多地用于符号推理任务 [12,16,38,44,62],，包括前提选择 [43]。

4.2 将前提选择作为二元分类（$\mathcal{L}$、$\mathcal{G}$）

梯度提升决策树（$\mathcal{L}$）：我们也使用梯度提升决策树（LightGBM）以二元模式进行前提选择。与深度学习方法相比，它们训练速度更快，在不平衡训练集上表现良好，并且能够很好地处理稀疏特征。根据我们之前将梯度提升决策树应用于前提选择的实验 [42]，我们在此固定 LightGBM 的超参数。在二元设置下，梯度提升决策树对猜想与候选前提的成对相关性进行评分。由于可能的候选项数量庞大（大型 ITP 库中所有先前的事实），我们首先使用成本更低的 k近邻算法对可用前提进行预筛选。LightGBM 的预测结果将在下文记为$\mathcal{L}$ 。

使用图神经网络进行依赖选择（$\mathcal{G}$）：第 3.3 节所述的消息传递图神经网络架构也可以应用于前提选择。与循环神经网络一样，它也可以考虑前提之间的依赖关系。由于图神经网络相对较慢，我们将把它与更简单的前提选择方法结合使用，例如 k-NN，预先选择 512 个事实。下文将用 $\mathcal{G}$ 表示图神经网络的预测。$\mathcal{L}$ 和$\mathcal{G}$都可以根据分数的阈值建立索引（如 $\mathcal{L}_{0.1}$ 或$\mathcal{G}_{-1}$），该阈值用于区分有用和无用的子句。

4.3 用于前提选择的集成方法（$\mathcal{E}$）

我们可以通过多种方式组合前几小节中讨论的前提选择方法。自然地，针对不同策略使用不同方法效果良好；不过，我们还发现，将通过多种方法获得的预测组合起来，并将其用于一次证明器运行，可以得到良好且互补的结果。由于不同算法产生的预测分数通常无法比较 [40],，我们只使用各种方法生成的排名，并据此创建一个组合排名。在之前的工作中 [30] ，我们比较了多种组合排名的方法，发现几种平均数效果良好：算术平均数、最小值和几何平均数；其中，调和平均数在实验中取得了最佳结果。此外，我们还为不同的组合方法添加权重。权重会优先考虑更强的预测方法，但也允许它总体上受益于较简单的方法（通过找回一些遗漏的事实）。给定来自 n 种不同方法的预测以及方法权重 w_1、.....、w_n，假设某个事实在第一种方法中的排名为 r_1，在最后一种方法中的排名为 r_n。那么，集成方法会为该事实赋予 $1/\sum_{i=1}^{n} \frac{w_i}{r_i}$ 的分数。将以此方式获得的事实分数进行排序，即可得到所有事实的排名。集成预测将记为$\mathcal{E}$，其上标和下标中标示方法及其权重，例如$\mathcal{E}_{0.25}^{\mathcal{K}\,\mathcal{N}\,\mathcal{G}}$、、、、0。25、0。5。

4.4 基于子问题的前提最小化（$\mathcal{M}$）

通过成功的自动定理证明器运行所获得的证明依赖，通常比人工编写的 ITP 证明中的依赖项更适合作为前提选择的数据 [7,31]。然而，有些 Mizar 证明长达数百行，因此目前还不现实地将在丛状设置下取得的 75% 自动定理证明器性能提升到接近 100% 的水平。这意味着，如果我们只使用基于自动定理证明器的前提数据，那么在前提选择训练中，目前将遗漏 MML 中现有证明依赖信息的 25%。

为了解决这个问题，我们在此首次采用了基于子问题的前提。其背后的思路是，Mizar证明较长的定理由一系列通常需要论证的自然演绎步骤组成。一旦某个顶层定理的所有此类步骤（我们称之为子问题）都有了 ATP 证明，就可以用它们来精简该定理的（过度近似的）人工编写前提集合。这种精简也会提高直接证明该定理的可能性

。具体而言，我们考虑以下方法：（1）仅使用由 ATP证明的子问题所提供的前提，忽略未证明的子问题。（2）在（1）的基础上，加入该定理所有明确列出的 Mizar前提（可能忽略部分背景事实）。（3）在（2）的基础上，再加入自然演绎模块检测到的（半显式）定义展开。（4）在（3）的基础上，再加入部分背景前提，通常是经过训练的前提选择器排序靠前的那些。采用（1）和（3）时，我们成功证明了 1000 多个困难定理（参见第6节 6.1 中的表 1）。在训练前提选择器时，我们还将（3）用作 ATP 未能证明的定理的额外证明依赖（第6节 6.2）。

5 策略与方法组合

策略： E、ENIGMA、吸血鬼和 Deepire 由 ATP策略及其组合进行参数化。ENIGMA 风格的引导通常涉及将规模较大（神经、基于树等）且可能较慢的统计模型应用于子句，而标准 ATP策略通常由速度快得多的子句评估函数以及使用证明器提供的领域特定语言编写的程序组成。对于特定的问题类别，这类程序同样可以通过各种方式发明和学习。在这里的实验中，我们使用了由 BliStr/Tune 系统自动发明的许多 ATP策略 [26, 27, 58]。这些系统通过反馈回路，将使用 ParamILS [23], 等引擎在问题簇上进行有针对性的 参数搜索 ，与对所发明策略进行大规模评估交错进行，并利用评估结果不断演化问题聚类。从少量策略开始，BliStr/Tune 通常会在每个策略表现最佳的问题上演化该策略。在使用这些系统进行实验期间，我们开发了数千种 E Prover 策略，其中许多针对 Mizar 问题。这些策略中的一部分将在第6节的实验中提及。

稳健组合： 更大型的人工智能/定理证明系统和元系统依赖于由互补策略构成的方法组合 [54] ，这些策略利用全局时间限制，以串行或并行方式攻克问题。在存在前提选择和多个自动定理证明器的情况下，这类方法组合可能由数十种到数百种不同方法组成。方法空间越大，在针对特定问题集构建方法组合时发生过拟合的风险也越大。例如，使用 SAT求解器解决集合覆盖问题（其中每种策略覆盖解空间的一部分）来朴素地构建“最优”方法组合，往往会导致方法组合高度专门化于特定的问题集。更稳健的方法（例如贪心覆盖）可以缓解这一问题，但其中的过拟合仍可能相当严重。例如，在 Mizar40实验中构建的包含 14 种策略的贪心覆盖，解决了用于构建它的随机子集的 44.1%，但在整个 MML 中仅解决了 40.6%，即少了 8%。

为改进这一点，我们在此提出一种更稳健的组合构建方法，同样基于通过控制过拟合来改进机器学习的思路。我们不再只是在整个开发集 D 上构建一个贪心覆盖 C （并设定一定的时间预算），然后在留出集 H上评估它，而是先将 D 随机划分为两个大小相等的部分 D_1和 D_2。接着，我们仅在 D_1上构建贪心覆盖 C_1 ，并在 D_2 和完整数据集 D上评估其性能。此过程重复 n 次（我们使用 $n = 1000$）；当 n 足够大时，这通常能保证在某些随机划分中，贪心覆盖 C_1^i D_1^i的过拟合程度很低（甚至出现欠拟合）。此外，还可以在许多其他随机划分上评估最佳（实力最强且过拟合最少的）覆盖，并选出最稳健的方案，从而进一步改进。我们在第6节中采用了这一方法来构建一个方法组合：它在（未见过的）留出集上的表现仅比用于构建它的开发集差 3.5%。

6 实验与结果

6.1 丛状实验与时间线

我们在评估中由自动定理证明器证明的全部 43 717 个 Mizar 问题的最终列表可在我们的网页上获取。[4] 这些方法及新增解答的大致时间线如表 1 所示。我们持续在网页上记录这一进程，[5] 这也展示了实验的推进情况，以及难度日益增加的问题是如何得到证明的。

大型评估于 2020 年 4 月启动，是我们在 ENIGMA Anonymous [25] 上工作的后续。在结合此前开发的方法并采用更长时间限制后，ENIGMA 在丛状设置下证明的问题比例于 2020 年 6 月达到 65。65%。此后，我们通过在大型 Malarea式反馈循环中反复进行训练和证明，持续推进这项工作。不断积累的证明被持续用于训练图神经网络和梯度提升引导，并将它们用于进一步的证明尝试，同时结合不同的搜索参数；后来，这些证明也被用于训练前提选择。

这包括针对一小部分随机抽取的问题进行大量网格搜索，搜索对象是数千个经过不同训练的 GNN 和 GBDT，它们分别对应不同的训练轮次；随后，使用这些经过不同训练的 GNN 和 GBDT，在全部问题或仅困难（迄今尚未被 ATP 证明）的问题上评估实力最强且互补性最佳的 ENIGMA。到 2021 年 9 月实验结束时，已保存的、对应不同训练轮次且可用于网格搜索和完整评估的 GNN 快照总数达到 15 920。[6] 我们进行的最长一次 GNN 训练包含 964 个训练轮次，在高端 NVIDIA V100 GPU 卡上耗时 12 天。[7] GNN 训练偶尔（但很少）会在数百个训练轮次后发散，我们通过重启来处理这种情况。

我们用于训练 ENIGMA 指导模型的证明总数最终超过了三百万，这些证明以序列化压缩形式存储时，占用空间超过 200 GB。由于完整数据甚至无法装入配备高效基于GPU的神经网络训练环境的大型服务器的主存中，我们编写了定制流水线，持续加载、混合并卸载用于 ENIGMA 训练的较小数据块。对于许多问题，我们获得了数百个不同的证明，而对于另一些问题，我们可能只有单个证明。这促使我们进一步开展实验，研究不同证明应以何种方式及多高频率呈现在训练数据中。这是更大规模的训练数据规范化任务的一部分，其中包括移除或删减训练数据中非常庞大的证明搜索，因为这些搜索会导致基于内存的GPU崩溃。

75%的里程碑于7月 26 2021[9] 通过使用新开发的二阶段和三阶段 ENIGMA，以及采用不同参数的交替迭代（第3.4节）运行而达成。最强的单一三阶段 ENIGMA 策略在严格的训练/开发/留出集设置下进行训练和评估时，在分支繁茂问题上于30秒内达到了56.4%的性能 [20]。该最佳 ENIGMA 使用0.01的亲本阈值、0.1的二阶段阈值，以及大小分别为768和256的上下文和查询。其（基于服务器的）图神经网络包含10层，在训练集中的每个问题上最多使用三个证明进行训练。另请参阅第6.6节，其中介绍了它在MML较新版本的242篇新文章中的13 370个全新问题集上的评估。

---

[4] http://grid01.ciirc.cvut.cz/~mptp/00proved_20210902_6 [5] https://github.com/ai4reason/ATP_Proofs 对于网格搜索而言，ENIGMA 和 E 策略的其他参数进一步加剧了这一问题。[7] 除层数 外，我们通常使用与 [25, 39] 相同的 GNN 超参数；这里层数在 5 到 12 之间变化，从而在图神经网络的速度和精度之间进行权衡。
[9] https://github.com/ai4reason/ATP_Proofs/blob/master/75percent_announce.md

已解决	[%]	*date*	前提	方法/注释
38k	65.65	2020年6月	$\mathcal{B}$	ENIGMA，于 IJCAR'20[8] 7月2日报道
40 268	69.57	2020年10月	$\mathcal{B}$	ENIGMA
40 994	70.83	11月12日	$\mathcal{M}$	ENIGMA，启发式前提最小化
41 169	71.13	11月12日	$\mathcal{M}$	吸血鬼，采用300秒限制，增加175
41 792	72.20	11月27日	$\mathcal{M}$	E/ENIGMA/Vampire，采用更多前提最小化
42 206	72.92	12月7日	$\mathcal{M}$	E/ENIGMA/Vampire配合更多前提最小化
42 471	73.38	1月6日	$\mathcal{G}$, $\mathcal{E}$	E 使用 BliStr/Tune策略处理 $\mathcal{G}$、$\mathcal{E}$ 前提
42 519	73.46	1月10日	many	ENIGMA在所有训练预测上运行
42 826	73.99	5月14日	$\mathcal{G}$, $\mathcal{L}$, $\mathcal{K}$	吸血鬼/Deepire 运行 – FroCoS'21 [53]
43 414	75.01	7月26日	$\mathcal{M}$,$\mathcal{B}$	2、三阶段 ENIGMA、交替迭代
43 524	75.20	8月21日	$\mathcal{M}$	三阶段 ENIGMA、上下文移位、蛙跳、前向包含。
43 599	75.33	8月26日	$\mathcal{L}$	三阶段 ENIGMA、交替迭代、前向包含
43 717	75.53	9月2日	$\mathcal{M}$	主要是 Vampire/Deepire

表 1 实验时间线。$\mathcal{B}$ 表示标准的分支繁茂的前提，$\mathcal{M}$ 表示经过子问题最小化的前提，$\mathcal{G}$、$\mathcal{L}$和 $\mathcal{K}$ 表示基于图神经网络/LightGDB/k近邻的前提，而$\mathcal{E}$ 表示它们的集成。

6.2 前提选择的训练数据

在以各种方式对这些问题运行学习/证明循环数月后，我们使用收集的数据训练前提选择方法。具体而言，当时有41 504个由自动定理证明器证明的问题，对于这些问题，我们通常拥有许多替代证明和前提集，从而产生了621 642个唯一的 ATP证明依赖。由于在锤击场景中我们还可以分析人工编写的证明并从中学习，因此，对于每个 ATP未证明的问题 P，我们通过取 P的所有子问题的 ATP依赖的并集，添加了该问题的前提。换言之，对于其余的困难问题，我们采用基于子问题的前提最小化（第4.4节）。这为前提选择数据集增加了16 651个示例。随后，我们以各种方式使用这个包含638 293个唯一证明依赖的数据集，在 MML上训练和评估前提选择方法。与 Mizar40实验相比，这大约多出六倍的证明数据。与机器学习实验中的惯例一样，我们还按照90 : 5 : 5的比例，将整个 Mizar 问题集划分为训练、开发和留出子集。这得到训练集中的52 125个问题、开发集中的2896个问题，以及留出集中的2896个问题。

6.3 训练前提选择器

我们首先使用不同的特征（第4.1节）及其组合，在训练子集上以多种方式训练 k近邻和朴素贝叶斯。训练图神经网络和 LightGBM 时，我们先使用基于 k近邻的预选择，为每个问题选出最相关的 512 个前提。训练时，我们将每个样例的正例（实际依赖）加入其中，并从 k近邻预选出的 512 个前提中移除这些正例，从而构成该样例的负例集。因此，图神经网络和 LightGBM 的训练目标是纠正 k近邻所犯的错误（这是一种提升形式）。预测时也采用相同的方式，即先用训练好的 k近邻预选 512 个前提，再由图神经网络/LightGBM 对其排序。我们既使用分数阈值（例如，纳入所有分数优于 0, -1 或 -3 的前提），也使用与其他前提选择方法相同的固定大小切片。使用 ENIGMA 的同一最佳版本时，最强的基于图神经网络的预测器（$\mathcal{G}_{-1}$）解决了 1089 个问题，而

模型	100-Cover		100-Prec		召回率		AUC		平均排名	
	D	H	D	H	D	H	D	H	D	H
$\mathcal{K}^{\mathrm{cp}}_{\mathrm{var}}$	83.3	82.3	8.837	8.713	386.8	401.9	92.03	91.27	90.17	97.98
$\mathcal{K}^{\mathrm{au}}_{\mathrm{var}}$	83.5	82.7	8.855	8.754	383.32	401.54	92.13	91.36	89.21	97.19
$\mathcal{K}^{\mathrm{mi}}_{\mathrm{var}}$	82.6	81.8	8.700	8.596	401.89	418.40	91.32	90.59	97.30	104.88
$\mathcal{K}^{\mathrm{s0}}_{\mathrm{var}}$	83.6	82.9	8.851	8.785	382.31	399.10	92.19	91.39	88.53	96.84
$\mathcal{N}_{\mathrm{cp}}$	87.8	87.0	9.739	9.665	300.49	310.72	94.77	94.32	62.64	67.51
$\mathcal{N}_{\mathrm{au}}$	88.0	87.5	9.748	9.714	298.66	307.82	94.84	94.44	61.99	66.31
$\mathcal{N}_{\mathrm{mi}}$	83.3	83.5	9.358	9.367	382.87	379.24	92.53	92.41	85.39	86.88
$\mathcal{N}_{\mathrm{s0}}$	88.3	87.5	9.776	9.720	299.10	308.67	94.85	94.41	61.85	66.60
$\mathcal{N}_{\mathrm{mi,时间}}$	83.9	82.7	9.151	9.010	384.68	393.37	92.33	91.75	87.23	93.27
$\mathcal{L}$	82.9	83.1	9.077	9.090	410.06	408.06	91.53	91.26	95.11	97.74
$\mathcal{G}$	87.4	86.3	9.408	9.282	241.22	249.32	88.45	87.42	66.69	71.87
$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5.5.\text{平均值}}$	87.8	87.1	9.606	9.522	291.27	304.43	95.06	94.53	59.81	65.47
$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5.5.\text{几何平均值}}$	89.4	88.7	9.806	9.733	277.75	288.53	95.53	95.04	55.13	60.27
$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5.5.\text{调和平均值}}$	89.4	88.9	9.822	9.780	276.34	286.23	95.53	95.08	55.06	59.94
$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5.5.\text{最小值}}$	89.0	88.4	9.753	9.707	279.88	289.41	95.37	94.95	56.70	61.19
$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.33..33..33}$	92.1	91.1	10.237	10.160	228.79	248.16	96.64	96.20	44.06	48.76
$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.25..25..25..25}$	92.5	91.5	10.297	10.219	210.31	227.74	96.93	96.56	41.20	45.17
$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.2..2..1}$	91.3	90.4	10.091	10.014	261.10	272.85	96.20	95.71	48.51	53.64

表2 在**D**evelopment和**H**oldout数据集上对机器学习前提选择模型的评估。请注意，此处仅为完整性而展示图神经网络的评估；在实践中，我们使用基于分数的阈值，并采用更少的前提。

基线 k近邻解决了 870 个问题，因此提升幅度很大（25.2 %）。图神经网络的表现也优于 LightGBM，后者似乎更容易在训练数据上过拟合。表2详细展示了评估中使用的主要方法在开发集和留出集上的性能。

6.4 ENIGMA 在前提选择数据上的实验

首先，为了在前提选择问题上训练 ENIGMA，我们使用 ENIGMA/GBDT 在前提切片上进行多轮证明/学习迭代。在循环 (1) 中，我们从三个选定的切片$\mathcal{G}_{-1}$、$\mathcal{L}_{0.1}$和$\mathcal{K}_{64}$，开始；实验发现这三者具有互补性。我们在这三个切片上评估策略$\mathcal{S}_1$ (bls0f17)，得到 20 604 个已证明的训练问题。我们使用不同的学习超参数（树叶节点数量、树深度、所用的 ENIGMA 特征）训练多个决策树（梯度提升决策树）模型，并使用所有可用的训练证明。在循环 (2) 中，我们评估多个在$\mathcal{B}$ （分支繁茂问题）上训练的 ENIGMA 模型，以获取额外的训练数据。经过几轮训练/评估迭代后，训练数据可能开始为某些（较容易的）问题积累大量由多种策略找到的证明。因此，从循环 (2) 开始，我们每个问题只使用有限数量的证明。我们可以为每个问题随机选取最多 6 个证明，也可以只选取特定的证明（例如最短、最长以及一个中等长度证明）。在循环 (3) 中，通过在前提切片上运行 ENIGMA/GNN，并使用 GBDT 运行结果训练 GNN，来增加额外的训练数据。在循环 (4) 中，我们纳入来自 7 个额外切片（$\mathcal{G}$、$\mathcal{L}$、$\mathcal{K}$ 的变体）的训练数据；这些数据是通过运行在分支繁茂问题上训练的 ENIGMA 模型获得的。在循环 (5) 中，我们扩充训练数据

loop	训练	开发（并集）	开发覆盖率（用时 420 秒）	[%]
init	20 604	1215	-	-
(1)	25 240	1601	1516	52.33
(2)	25 725	1669	1555	53.69
(3)	25 887	1679	1560	53.88
(4)	29 266	1716	1591	54.94
(5)	37 053	1735	1610	55.59

证明器（420 秒）	覆盖	对	[%]
E 2.6（自动调度）	1430	14	49.38
Vampire 4.0（CASC）	1536	14	53.03
BliStr/Tune	1582	210	54.62
ENIGMA/GBDT	1610	42	55.59
ENIGMA/GNN	1670	84	57.66

表3 ENIGMA/GBDT模型的训练（左），以及开发集的最佳覆盖率（右）。

加入通过我们此前各种尝试为未解决的训练问题找到的分支繁茂证明。

从1215个已解决的开发问题出发，经过第五次迭代后，最终解决了1735个问题。虽然我们仅在少数选定的切片上训练GBDT模型，但会在更多的切片上评估这些模型，最多涵盖56个开发切片，覆盖所有类别$\mathcal{G}$、$\mathcal{L}$、$\mathcal{K}$、$\mathcal{E}$和$\mathcal{N}$。我们报告训练问题数量（训练）的增长情况，以及所有经过评估的策略/切片对所解决的开发问题总数（开发 并集）。由于每个策略/切片对的评估时间均为10秒，我们构建了最佳42个策略/切片对的贪心覆盖，以近似在420秒内可获得的最佳结果（见“开发覆盖率”列）。由于开发集未以任何方式用于训练GBDT模型，因此可以将此结果视为留出集上最佳可能结果的近似。

仅使用 ENIGMA/GBDT模型，我们在 420 秒内可解的问题中达到 55.59%。为了将这一结果与其他方法进行比较，我们分别为处于自动调度模式的 E Prover 和处于*CASC*模式的 Vampire 构建兼容的贪心覆盖，也就是说，采用它们各自最强的默认设置。我们在全部 56 个开发切片上评估这两个证明器，每个问题的时间限制为 30 秒。对于每个证明器，我们构建由最佳 14 个切片组成的贪心覆盖，再次近似于在 420 秒内所能获得的最佳结果。BliStr/Tune 是我们此前为 Mizar 分支繁茂问题设计的、包含 15 个 E 策略的方法组合。我们在全部 56 个切片上评估这 15 个策略，每个问题使用 2 秒。类似地，我们构建长度为 210 的贪心覆盖。对 列指定了各情况下所考虑的贪心覆盖长度。每个策略/切片对的时间限制为 420/。

在五个循环中获得的训练数据最终用于训练面向前提选择切片的新 ENIGMA/GNN模型。我们训练并评估了各种 GNN模型（层数不同，以及来自不同轮次的网络），时间限制为 5 秒。与之前一样，我们构建长度为 84 的贪心覆盖，以模拟在 420 秒内所能获得的最佳运行结果。ENIGMA/GNN 的表现甚至优于 ENIGMA/GBDT，可解决 57.66% 的问题。两个 ENIGMA/*策略组合在 840 秒内合计覆盖了 1701 个开发问题，表明这些方法具有良好的互补性。请注意，只有 ENIGMA/GBDT 策略最多能够覆盖 1735 个问题（见左侧开发 列），这占开发集的 59.9%。

我们的大多数 ENIGMA 模型都与基准策略 bls0f17 结合使用。bls0f17 和 bls05fc 是 BliStr/Tune [24] 提出的两种策略，它们在前提选择数据上表现出色。我们还使用了另外两种较早的 BliStr [58]策略 mzr02 和 mzr03，它们在分支繁茂问题上表现出色。通常，我们只会合并来自项排序和文字选择设置兼容的策略的训练数据。不过，研究发现，少量使用来自排序方式不兼容的策略的数据也有帮助。我们还使用其他几种 BliStr 和吸血鬼策略，以及 E 的自动 模式，来收集更多已解决的开发问题。借助我们的所有方法（ENIGMA

$\mathcal{G}$	$\mathcal{E}_{5221}$	$\mathcal{L}$	$\mathcal{E}_{5221}$	$\mathcal{N}_{\text{uni}}$	$\mathcal{E}_{5221}$	$\mathcal{N}_{\text{eni}}$	$\mathcal{N}_{\text{eni}}$	$\mathcal{E}_{5221}$	$\mathcal{G}$	$\mathcal{E}_{55}$	$\mathcal{E}_{533}$	$\mathcal{E}_{533}$
图神经网络	图神经网络	图神经网络	图神经网络 V		V	图神经网络	图神经网络	图神经网络	图神经网络 V		V	GNN
984	1142	1215	1263	1297	1325	1346	1370	1381	1393	1405	1419	1444
1013	1157	1240	1275	1305	1321	1346	1364	1378	1386	1398	1407	1436

表4 最终组合的95个切片中的13个切片的前缀。每一列依次给出前提选择方法、自动定理证明方法，以及截至该切片在开发集和留出集上累计解决的问题数量。“V”代表吸血鬼，“图神经网络”是基于 bls0f17 的 ENIGMA/图神经网络模型。此外，$\mathcal{E}_{5221} = \mathcal{E}^{\mathcal{N}\ \mathcal{K}\ \mathcal{L}\ \mathcal{G}}_{.5\ .2\ .2\ .1}$,,,,,,以及 $\mathcal{E}_{55} = \mathcal{E}^{\mathcal{N}\ \mathcal{K}}_{.5\ .5}$,,,平均值 和 $\mathcal{E}_{533} = \mathcal{E}^{\mathcal{N}\ \mathcal{G}\ \mathcal{K}}_{.5\ .25}$,,,,.25.

和 BliStr/Tune），以及针对所选策略运行的额外吸血鬼程序，我们已解决超过 62.7% 的开发问题。这些结果为构建最终的留出组合提供了训练数据，具体内容将在下一节中介绍。

6.5 最终 HammerPortfolio

借助在前提切片上运行的系统所产生的大型开发结果数据库，我们最终构建出终极自动证明组合。为此，我们采用第5节所述的稳健组合构建 方法。具体而言，我们将开发集随机划分为两个大小相等的部分，并利用第一部分上的完整结果数据库计算420秒的贪心覆盖。随后在第二部分上评估该贪心覆盖，从而衡量过拟合程度。这个随机化过程重复一千次。然后，我们（手动）选出20个最强且最不易过拟合的方法组合，并在另外80次随机划分上分别评估它们，从而衡量其平均不平衡性。通常，它们在整个开发集上的性能最高可达60.5%，因此我们在160个随机半集上选择59.5%的阈值来衡量不平衡性。其中最平衡的方法组合胜出：160个随机半集中有135个通过了该阈值。

这个最终的420秒组合包含95个切片，解决了开发问题中的1749个（60.4 %）和留出问题中的1690个（58.36 %）。表4展示了该方法组合由13个切片构成的初始部分，以及解决的问题数量。完整的方法组合见表5。第一个数字t 表示运行该切片的秒数。基数 列指定所使用的自动定理证明策略，而*ENIGMA* 列描述所使用的ENIGMA模型类型（如有）。我们可以看到，GNN模型凭借快速运行占据了调度表的主导地位。调度表最后是运行时间较长的项目，其中尤其包括GBDT模型；尽管是在单CPU环境中进行评估，这些模型仍需要数秒来加载。这意味着得益于预加载图神经网络服务器，我们优先采用了GNN ENIGMA；如果我们也预加载GBDT模型，预计还能进一步改进。我们最强的单个基于图神经网络的策略使用$\mathcal{G}_{-1}$ 预测，在30 s内解决了1178个留出问题。这一比例为39.5 %，仅比Mizar40实验中构建的完整420 s方法组合所解决的40.6 %低1.1 %。

6.6 迁移到 MML1382

在最后一项实验中，我们使用训练效果最佳的 ENIGMA（三阶段，见第6.1节）对来自新版 Mizar（1382）的分支繁茂问题运行 120 秒；该版本包含 242 篇新文章，其中有 13 370 个定理。ENIGMA 不仅从未在这些文章中的任何一篇上进行训练，也从未接触过其中引入的新术语。我们还在该新版上运行标准的 E 自动调度 120 秒。ENIGMA 证明了新文库中 70 396 个问题里的 37 094 个（52.7 %），而 E 自动调度证明了其中的 24 158 个（34.32 %）。因此，ENIGMA

t	*base*	*ENIGMA*	切片
2	bls0f17	GNN	$\mathcal{G}$
2	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
2	bls0f17	GNN	$\mathcal{L}_{0.1}$
2	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
2	吸血鬼	-	$\mathcal{N}_{\text{uni}}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
2	bls0f17	GNN	$\mathcal{N}_{\text{eni}}$
2	bls0f17	GNN	$\mathcal{N}_{\text{eni}}$
2	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
2	bls0f17	GNN	$\mathcal{G}_{-3}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.25}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25}$
5	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
2	吸血鬼	-	$\mathcal{K}^{\text{au}}_{\text{var}}$
2	吸血鬼	-	$\mathcal{K}^{\text{cp}}_{\text{var}}$
2	mzr02	-	$\mathcal{G}_0$
2	bls0f17	GNN	$\mathcal{G}_{-5}$
2	vampire-16	-	$\mathcal{N}_{\text{uni}}$
2	吸血鬼	-	$\mathcal{N}_{\text{eni}}$
2	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
5	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
5	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25}$
2	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
2	vampire-16	-	$L_{0.05}$
5	bls0f17	GNN	$\mathcal{G}$
2	vampire-18	-	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L}}_{.5,.2,.3}$
2	mzr22	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5,geo}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
2	吸血鬼	-	$K^{\text{cp}}_{\text{var}}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L}}_{.5,.2,.3}$
2	BliStr-edc9	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5,chrono}$
2	吸血鬼	-	$\mathcal{N}_{\text{uni}}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25}$
5	bls0f17	GNN	$\mathcal{L}_0$
10	mzr02	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
5	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
5	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
2	bls0f17	GNN	$\mathcal{N}_{\text{eni}}$
2	vampire-2	-	$\mathcal{G}_{16}$
2	vampire-16	-	$\mathcal{N}_{\text{au}}$
2	吸血鬼	-	$\mathcal{N}_{\text{au}}$
2	E-auto	-	$\mathcal{L}_{96}$
10	bls05fc	GBDT	$L_{0.25}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5,min}$
2	vampire-16	-	$\mathcal{N}_{\text{sub}}$
2	vampire-16	-	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25,.25}$

t	*base*	*ENIGMA*	切片
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25}$
2	Blistr-5fce	-	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L}}_{.5,.2,3}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
2	E-auto	-	$\mathcal{G}_{-2}$
2	吸血鬼	-	$\mathcal{K}^{\text{uni}}_{\text{var}}$
2	vampire-16	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5,geo}$
2	吸血鬼	-	$K^{\text{uni}}_{\text{var}}$
2	vampire-21	-	$\mathcal{G}$
2	吸血鬼	-	$\mathcal{N}_{\text{sub}}$
2	吸血鬼	-	$\mathcal{N}_{\text{uni}}$
2	E-auto	-	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25}$
2	吸血鬼	-	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5,min}$
2	吸血鬼	-	$\mathcal{N}_{\text{cp}}$
2	bls0f17	GNN	$\mathcal{K}^{\text{eni}}_{\text{var}}$
5	bls0f17	GNN	$L_{0.01}$
10	bls05fc	GBDT	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.33,.33}$
5	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
5	bls0f17	GNN	$\mathcal{N}_{\text{eni}}$
5	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25}$
10	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
10	mzr03	GBDT	$\mathcal{G}_{64}$
10	bls05fc	GBDT	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.33,.33,.33}$
10	mzr02	GBDT	$\mathcal{K}_{\text{短}}$
10	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.5,.25}$
10	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
10	bls0f17	GBDT	$L_{0.01}$
5	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
5	bls0f17	GNN	$L_{0.005}$
10	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.5,.2,.2,.1}$
5	bls0f17	GNN	$L_{0.01}$
5	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
5	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
5	mzr03	-	$\mathcal{E}^{\mathcal{N},\mathcal{K},\mathcal{L},\mathcal{G}}_{.25,.25,.25,.25}$
10	bls0f17	GNN	$\mathcal{G}_{-1}$
5	bls0f17	GNN	$\mathcal{G}_0$
10	bls05fc	GBDT	$\mathcal{K}_{\text{短}}$
5	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
5	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
10	bls0f17	GNN	$\mathcal{G}_{0.5}$
10	bls0f17	GNN	$\mathcal{N}_{\text{eni}}$
10	bls0f17	GBDT	$L_{0.01}$
10	bls0f17	GBDT	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5,\text{&最小值}}$
10	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
10	bls0f17	GNN	$\mathcal{E}^{\mathcal{N},\mathcal{K}}_{.5,.5}$
10	bls0f17	GNN	$\mathcal{N}_{\text{au}}$
10	mzr03	GBDT	$\mathcal{E}^{\mathcal{N},\mathcal{G},\mathcal{K}}_{.33,.33}$

表5 420 秒的最终 Mizar 锤子方法组合。

在新文库上比 E 提高了 53.55%。我们将其与旧 MML 进行比较：在旧 MML 中，经过训练的 ENIGMA 能解决 57 880 个问题中的 34 528 个（59.65%），而 E 能解决 22 119 个（38.22%），也就是说，相对提升幅度为 56.10%。

令人惊讶的是，仅在新的 13 370 个定理中，其中一半以上包含新术语，ENIGMA 证明的问题与 E 证明的问题之比为 5934 比 3751，也就是说，在这里 ENIGMA 比 E 高出 58.20%。这些数字表明，我们的$anonymous$ [25] 逻辑感知机器学习方法仅从数学问题的结构中学习，其性能几乎不受迁移到包含许多新概念和引理的新环境的影响。这在当今的机器学习领域相当不寻常，因为该领域似乎由大型语言模型主导，而大型语言模型通常难以应对新术语。

7 个证明

随着主要实验从2020年春季推进到2021年夏季，我们收集了自动发现的证明中的有趣实例，并在我们的网页上发布了它们的摘要描述。[10] 截至2021年9月，此类示例证明已超过200个，最初的自动定理证明器长度为几十个子句步骤，后来逐渐达到数百个子句步骤。最初，这些证明是在丛状设置中找到的；后来又加入了在链状（前提选择）设置中完成的证明，通常用于展示替代前提的效果。

我们发布到网页上的最早证明之一[11] 是 `NEWTON:72` [12] 证明对于每个自然数，都存在一个更大的素数：

```
for l being Nat ex p being Prime st p is prime & p > l
```

ENIGMA证明[13] 从328个预先选定的Mizar事实开始，这些事实被转换为549个初始子句。当时（2020年4月）运行在中央处理器上的特定版本图神经网络引导着搜索。由于其成本相对较高，证明搜索仅在6秒内生成了2856个非平凡子句，执行了734次非平凡的给定子句循环。最终证明包含83个子句步骤，并使用了最初提供的328个步骤中的38个。其中许多步骤重现了Mizar中完成的算术论证。一个有趣之处在于，引导式证明器能够在这里利用所提供的事实综合出一个非平凡见证（$n! + 1$），此后，鉴于文库中的知识，该证明很可能会变得相当直接（关于此实例的更详细讨论，见附录）。总的来说，将所提供的事实与训练好的学习器结合起来，用于引导式综合非平凡见证，似乎是ENIGMA引导带来的主要改进之一，并促成了相较于Mizar40评估的新证明。这促使我们开始研究面向人工智能/定理证明设置的见证和猜想的神经综合 [13, 16, 17, 59]。

算术推理以及其他通常被称为“常规计算”的领域，后来都成为 ENIGMA 经常通过解决越来越难的 Mizar 问题并从中学习而逐步改进的领域。这类问题包括关于三角函数、积分、导数、矩阵运算等的推理。在三阶段 ENIGMA 完成的更高级成果中，例如，有一个 619 行的证明 `SINCOS10:86` [14] 在 60 秒内找到，其中对反正割函数的定义域和值域进行了大量计算，[15] 以及一个 326 行的证明

[10] https://github.com/ai4reason/ATP_Proofs
[11] https://bit.ly/3Spmf26
[12] https://bit.ly/3ILEkEp
[13] https://bit.ly/3Z2iXo3
[14] https://bit.ly/3StOHzV
[15] https://bit.ly/2YZOOgX

FDIFF_8:14，[16] 在 31 秒内找到，内容涉及 tan（自然对数 x）的导数。[17]

```
for x being set st x in [.(- (sqrt 2)),(- 1).] holds arcsec2 . x in [.((3 / 4) * PI),PI.]

for Z being open Subset of REAL st Z c= dom (tan * ln) holds tan * ln is_differentiable_on Z
& for x being Real st x in Z holds ((tan * ln) `| Z) . x = 1 / (x * (cos . (ln . x))^2)
```

第一个证明使用了 83 条 Mizar事实，起始于 1025 条预先选定的事实。其证明搜索耗用了 5344 个非平凡给定子句，并总共生成了超过 10 万个非平凡子句，因此三阶段过滤和 GPU 服务器的使用对于高效找到该证明至关重要。第二个证明使用了 55 条 Mizar事实、3136 个给定子句循环，并生成了 2.66 万个非平凡子句。读者可以在我们的网页上看到，还有许多此类“主要是计算型”的问题已经得到解决，这表明此类学习方法可能适合自动掌握日常计算任务中的能力，而无需像在 SMT求解器中那样手动编写程序。这促使我们开展了关于学习推理组件的研究 [10]。由三阶段 ENIGMA 找到的两个不那么“计算型”但仍然很长的自动定理证明器证明分别是 BORSUK_5:31 [18] 说明有理数在 (a,b) 上的闭包为 [a,b],[19] 以及 IDEAL_1:22 [20] 说明交换环是域当且仅当理想是平凡的：[21]

```
for A being Subset of R^1 for a, b being real number
st a < b & A = RAT (a,b) holds Cl A = [.a,b.]

for R being non degenerated comRing holds R is Field iff
for I being Ideal of R holds I = {(0. R)} or I = the carrier of R
```

BORSUK_5:31 的 Mizar证明需要 80 行。ENIGMA 从 38 条 Mizar事实中找到了一份证明，其中包含 359 个子句步骤，经过 4883 次给定子句循环。在生成的 40 万个子句中，多阶段 ENIGMA 的机制如下：亲本引导冻结了 133 869 个子句，随后激进吸收筛选掉 83 871 个，第一阶段的 LightGBM 模型又筛选掉 64 364 个。剩余的 125 489 个“优质”子句由 GNN服务器逐步评估（分 176 批调用），使用包含 1536 个已处理子句的上下文。ENIGMA 对IDEAL_1:22 的证明使用了 48 条 Mizar事实，包含 493 个子句步骤，经过 4481 次给定子句循环。

一个借助前提选择器发现文库中的替代引理而得以完成的自动定理证明器证明示例是 FIB_NUM2:69。[22] 这一在 MML 中称为“Carmichael素因子定理”的定理表明：如果 m 整除第 n个斐波那契数（Fib n），那么 m 不整除任何更小的斐波那契数，前提是 m，n 是素数。[23] Mizar证明有 122 行，使用了归纳法，而到目前为止我们还无法使用自动定理证明器重放该证明。然而，前提选择器找到了一个先前的库引理 FIB_NUM:5 [24]，其表明 (Fib m) gcd (Fib n) = Fib (m gcd n)，由此使用 159 个子句步骤、4214 次给定子句循环和 32 个 Mizar事实即可得到证明。最后，一个使用较高时间限制的较长 Deepire证明示例是 [25] ORDINAL5:36，即 $\epsilon_0 = \omega^{\omega^{\omega^{\cdots}}}$ 零阶 epsilon序数的公式：[27]

[16] https://bit.ly/3IuYHV0
[17] https://bit.ly/3SdZjTq
[18] https://bit.ly/3KzuPJY
[19] https://bit.ly/3C0Lwa8
[20] https://bit.ly/3Z7UPQC
[21] https://bit.ly/3BWqR6K
[22] https://bit.ly/3YWIfE6
[23] https://bit.ly/3oGBdRz
[24] https://bit.ly/3ExtvmS
[25] https://bit.ly/3klDrJr
[26] https://bit.ly/3SrPyRN
[27] https://bit.ly/3SozGPM

```
first_epsilon_greater_than 0 = omega |^|^ omega
```

搜索耗用了 38 065 次给定子句循环和 504 秒。该证明包含 1193 个子句步骤，使用了 49 个 Mizar事实。在这里，Deepire 极其高效的神经引导仅耗用了总时间中的 18 秒。

8 结论：人工智能/定理证明赌约已完成

2014年，Kaliszyk 和 Urban 在 Flyspeck 和 Mizar 语料库上均取得了 40 % 的成绩后，最后一位作者在亨利·庞加莱研究所的一场演讲中公开宣布了三项人工智能/定理证明赌约[28]，并提出愿为此下注，金额最高可达 10 000 欧元。第二项赌约的一部分内容是：到 2024年，在采用与 2014年相同的设置时，MML 和 Flyspeck 的顶层定理中将有 60 % 可由自动方式证明。在高阶逻辑环境下，TacticToe 系统早在 2017/18 年就实现了这一目标：它在 60 秒内证明了 HOL 库中 66.4 % 的定理，在 120 秒内则达到 69 % [14,15]。不过，也有人可能会认为，TacticToe 引入了一种新型的机器学习引导的战术证明器，它从针对语料库定制的专家编写的过程中特别受益。这一点尤其体现在需要归纳法的 HOL 问题上：TacticToe 的表现大幅提升，而标准的高阶自动定理证明器在处理这类问题时历来表现不佳。

在这项工作中，我们基本完成了第二个人工智能/定理证明赌约的这一部分，也将其应用于 Mizar 文库。主要需要注意的是，我们使用了较新的硬件，尤其是许多 ENIGMA 实验使用 GPU 服务器进行子句评估。不过，显然（无论从 LightGBM 实验，还是从高效且基于 CPU 的 Deepire 实验来看），这并不是一个主要问题。尽管如今在基于机器学习的实验中，使用专用硬件通常更容易，但人们也越来越多地研究如何从在 GPU 上训练的模型中提取出速度更快的预测器，使其能在标准硬件上更高效地运行。

9 致谢

ENIGMA、前提选择以及本文使用的其他方法的开发，以及大规模实验，都得益于许多非正式讨论，参与者包括（至少）Lasse Blaauwbroek、Chad Brown、Thibault Gauthier、Mikolas Janota、Jelle Piepenbrock、Stanislaw Purgal、Bob Veroff 和 Jiri Vyskocil。方法的多年开发和实验所需资金，部分来自 ERC 协同资助项目 $AI4REASON$ 编号 649043（ZG、JJ、BP、MS 和 JU）；欧洲区域发展基金资助的捷克项目 AI&Reasoning，编号 CZ.02.1.01/0.0/0.0/15_003/0000466（KC、ZG、JJ、MO 和 JU）；ERC 初创资助项目 $SMART$ 编号 714034（JJ、CK 和 MO）；捷克科学基金会项目 20-06390Y 及欧盟 H2020 计划下的 RICAIP 项目编号 857306（MS）；ERC-CZ 项目 POSTMAN 编号 LL1902（JJ、BP）；亚马逊研究奖（JU）；以及欧盟 ICT-48 2020 项目 TAILOR 编号 952215（JU）。

参考资料

1 Martín Abadi、Ashish Agarwal、Paul Barham、Eugene Brevdo、Zhifeng Chen、Craig Citro、Greg S. Corrado、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Ian Goodf- ell、Andrew Harp、Geoffrey Irving、Michael Isard、Yangqing Jia、Rafal Jozefowicz、Lukasz Kaiser、Manjunath Kudlur、Josh Levenberg、Dandelion Mané、Rajat Monga、Sherry Moore、Derek Murray、Chris Olah、Mike Schuster、Jonathon Shlens、Benoit Steiner、IlyaSutskever、Kunal Talwar、Paul Tucker、Vincent Vanhoucke、Vijay Vasudevan、Fernanda Viégas、Oriol Vinyals、Pete Warden、Martin Wattenberg、Martin Wicke、Yuan Yu 和 Xiaoqiang Zheng。

[28] http://ai4reason.org/aichallenges.html

TensorFlow：异构系统上的大规模机器学习，2015年。软件可从 tensorflow.org 获取。URL：https://www.tensorflow.org/。2 Jesse Alama、Tom Heskes、Daniel Kühlwein、Evgeni Tsivtsivadze 和 Josef Urban。通过语料库分析和核方法进行数学前提选择。《自动推理杂志》，52(2)：191–213，2014年。doi:10.1007/s10817-013-9286-5。3 Grzegorz Bancerek、Czeslaw Bylinski、Adam Grabowski、Artur Kornilowicz、Roman Matuszewski、Adam Naumowicz 和 Karol Pak。Mizar 数学库在 Mizar 交互式证明开发中的作用。《自动推理杂志》，61(1-4)：9–32，2018年。URL：https://doi.org/10.1007/s10817-017-9440-6、doi:10.1007/s10817-017-9440-6。4 Grzegorz Bancerek、Czeslaw Bylinski、Adam Grabowski、Artur Kornilowicz、Roman Matuszewski、Adam Naumowicz、Karol Pak 和 Josef Urban。Mizar：现状与展望。载于 Manfred Kerber、Jacques Carette、Cezary Kaliszyk、Florian Rabe 和 Volker Sorge 编，《智能计算机数学——CICM 2015 国际会议，2015年7月13日至17日，美国华盛顿特区，会议录》，《计算机科学讲义》第9150卷，第261–279页。Springer，2015年。URL：https://doi.org/10.1007/978-3-319-20615-8_17、doi:10.1007/978-3-319-20615-8_17。5 Grzegorz Bancerek 和 Piotr Rudnicki。MIZAR 中的连续格纲要。《自动推理杂志》，29(3-4)：189–224，2002年。6 Jasmin Christian Blanchette、David Greenaway、Cezary Kaliszyk、Daniel Kühlwein 和 Josef Urban。基于学习的事实选择器，用于 Isabelle/HOL。《自动推理杂志》，57(3)：219–244，2016年。URL：http://dx.doi.org/10.1007/s10817-016-9362-8、doi:10.1007/s10817-016-9362-8。7 Jasmin Christian Blanchette、Cezary Kaliszyk、Lawrence C. Paulson 和 Josef Urban。迈向 QED 的锤击法。《形式化推理杂志》，9(1)：101–148，2016年。URL：http://dx.doi.org/10.6092/issn.1972-5787/4593、doi:10.6092/issn.1972-5787/4593。8 Tianqi Chen 和 Carlos Guestrin。XGBoost：一种可扩展的树提升系统。载于第22届 ACM SIGKDD知识发现与数据挖掘国际会议论文集，KDD '16，第785–794页，美国纽约州纽约市，2016年。ACM。URL：http://doi.acm.org/10.1145/2939672.2939785、doi:10.1145/2939672.2939785。9 Kyunghyun Cho、Bart van Merrienboer、Çaglar Gülçehre、Dzmitry Bahdanau、Fethi Bougares、Holger Schwenk 和 Yoshua Bengio。使用 RNN 编码器-解码器学习用于统计机器翻译的短语表示。载于 Alessandro Moschitti、Bo Pang 和 Walter Daelemans 编，第2014年自然语言处理经验方法会议论文集，EMNLP 2014，2014年10月25日至29日，多哈，卡塔尔，ACL 的特殊兴趣小组 SIGDAT 会议，第1724–1734页。ACL，2014年。URL：http://aclweb.org/anthology/D/D14/D14-1179.pdf、doi:10.3115/v1/d14-1179。10 Karel Chvalovský、Jan Jakubuv、Miroslav Olsák 和 Josef Urban。学习定理证明组件。载于 Anupam Das 和 Sara Negri 编，《基于解析表及相关方法的自动推理——第30届国际会议，TABLEAUX 2021，英国伯明翰，2021年9月6日至9日，会议录》，《计算机科学讲义》第12842卷，第266–278页。Springer，2021年。URL：https://doi.org/10.1007/978-3-030-86059-2_16、doi:10.1007/978-3-030-86059-2_16。11 Karel Chvalovský、Jan Jakubuv、Martin Suda 和 Josef Urban。ENIGMA-NG：面向 E 的高效神经与梯度提升推理指导。载于 Pascal Fontaine 编，《自动演绎——CADE 27——第27届国际自动演绎会议》，巴西纳塔尔，2019年8月27日至30日，会议录，《计算机科学讲义》第11716卷，第197–215页。Springer，2019年。URL：https://doi.org/10.1007/978-3-030-29436-6_12、doi:10.1007/978-3-030-29436-6_12。12 Richard Evans、David Saxton、David Amos、Pushmeet Kohli 和 Edward Grefenstette。神经网络能理解逻辑蕴涵吗？载于学习表征国际会议，2018年。URL：https://openreview.net/forum?id=SkZxCk-0Z。

20 MizAR 60适用于Mizar 50

13 Thibault Gauthier。高阶逻辑中的函数综合深度强化学习。载于 LPAR，《EPiC 计算系列》第73卷，第230–248页。EasyChair，2020年。14 Thibault Gauthier、Cezary Kaliszyk 和 Josef Urban。TacticToe：学习使用 HOL4 策略进行推理。载于 Thomas Eiter 和 David Sands 编，第21届逻辑编程、人工智能与推理国际会议 LPAR-21，博茨瓦纳毛努，2017年5月7日至12日，《EPiC》第46卷，第125–143页。EasyChair，2017年。URL： https://doi.org/10.29007/ntlb, doi:10.29007/ntlb。15 Thibault Gauthier、Cezary Kaliszyk、Josef Urban、Ramana Kumar 和 Michael Norrish。Tactictoe：学习使用策略进行证明。《自动推理杂志》，65(2)：257–286，2021年。URL：https://doi.org/10.1007/s10817-020-09580-x, doi:10.1007/s10817-020-09580-x。16 Thibault Gauthier、Miroslav Olsák 和 Josef Urban。Alien coding。CoRR，abs/2301.11479，2023年。17 Thibault Gauthier 和 Josef Urban。从零开始学习整数序列的程序综合。CoRR，abs/2202.11908，2022年。18 Zarathustra Goertzel、Jan Jakubův 和 Josef Urban。ENIGMAWatch：ProofWatch 遇见 ENIGMA。载于 Serenella Cerrito 和 Andrei Popescu 编，《基于解析表及相关方法的自动推理》，第374–388页，沙姆，2019年。Springer International Publishing。19 Zarathustra Amadeus Goertzel。让 E 再次变得智能（短文）。载于 IJCAR（第2部分），《计算机科学讲义》第12167卷，第408–415页。Springer，2020年。20 Zarathustra Amadeus Goertzel、Karel Chvalovský、Jan Jakubuv、Miroslav Olsák 和 Josef Urban。快速与慢速Enigma以及亲本引导。载于 FroCoS，《计算机科学讲义》第12941卷，第173–191页。Springer，2021年。21 Zarathustra Amadeus Goertzel、Jan Jakubuv、Cezary Kaliszyk、Miroslav Olsák、Jelle Piepen- brock 和 Josef Urban。Isabelle ENIGMA。载于 ITP，《LIPIcs》第237卷，第16:1–16:21页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2022年。22 Adam Grabowski、Artur Korniłowicz 和 Adam Naumowicz。Mizar概览。《形式化推理杂志》，3(2)：153–245，2010年。23 Frank Hutter、Holger H. Hoos、Kevin Leyton-Brown 和 Thomas Stützle。ParamILS：一种自动算法配置框架。《人工智能研究杂志》，36：267–306，2009年10月。24 Jan Jakubův 和 Josef Urban。定理证明策略的层次化发明。《人工智能通讯》，31(3)：237–250，2018年。URL： https://doi.org/10.3233/AIC-180761， doi:10.3233/AIC-180761。25 Jan Jakubuv、Karel Chvalovský、Miroslav Olsák、Bartosz Piotrowski、Martin Suda 和 Josef Urban。ENIGMA Anonymous：符号无关推理引导机（系统描述）。载于 IJCAR（第2部分），《计算机科学讲义》第12167卷，第448–463页。Springer，2020年。26 Jan Jakubuv、Martin Suda 和 Josef Urban。为 vampire 自动发明策略和项序。载于 GCAI，《 EPiC 计算系列》第 50 卷，第 121–133 页。EasyChair，2017年。27 Jan Jakubuv 和 Josef Urban。BliStrTune：定理证明策略的层次化发明。载于 Yves Bertot 和 Viktor Vafeiadis 编，《第六届 ACM SIGPLAN 经认证的程序与证明会议论文集》，CPP 2017，法国巴黎，2017年1月16日至17日，第43–52页。ACM，2017年。URL： http://doi.acm.org/10.1145/3018610.3018619， doi:10.1145/3018610.3018619。28 Jan Jakubuv 和 Josef Urban。增强 ENIGMA 给定子句引导。载于 Florian Rabe、William M. Farmer、Grant O. Passmore 和 Abdou Youssef 编，《智能计算机数学——第十一届国际会议》，CICM 2018，奥地利哈根贝格，2018年8月13日至17日，论文集，《计算机科学讲义》第11006卷，第118–124页。Springer，2018年。URL： https://doi.org/10.1007/978-3-319-96812-4_11, doi:10.1007/978-3-319-96812-4_11。

29 Jan Jakubuv 和 Josef Urban。通过学习子句引导来锤炼 Mizar。载于 John Harrison、John O'Leary 和 Andrew Tolmach 编，《第十届交互式定理证明国际会议》，ITP 2019，2019年9月9日至12日，美国俄勒冈州波特兰，《LIPIcs》第141卷，第34:1–34:8页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2019年。URL：https://doi.org/10.4230/LIPIcs.ITP.2019.34，doi:10.4230/LIPIcs.ITP.2019.34。30 Cezary Kaliszyk 和 Josef Urban。通过特征加权和策略演化增强 Flyspeck 的自动化。载于 Jasmin Christian Blanchette 和 Josef Urban 编，PxTP 2013，《EPiC Series》第14卷，第87–95页。EasyChair，2013年。31 Cezary Kaliszyk 和 Josef Urban。使用 Flyspeck 进行学习辅助自动推理。《自动推理杂志》，53(2)：173–213，2014年。URL：http://dx.doi.org/10.1007/s10817-014-9303-3，doi:10.1007/s10817-014-9303-3。32 Cezary Kaliszyk 和 Josef Urban。MizAR 40 for Mizar 40。《自动推理杂志》，55(3)：245–256，2015年。URL：http://dx.doi.org/10.1007/s10817-015-9330-8，doi:10.1007/s10817-015-9330-8。33 Cezary Kaliszyk、Josef Urban 和 Jirí Vyskocil。大型理论自动推理的高效语义特征。载于 IJCAI，第3084–3090页。AAAI Press，2015年。34 Cezary Kaliszyk、Josef Urban 和 Jirí Vyskocil。通过数学的统计解析和语义解析实现形式化自动化。载于 ITP，《计算机科学讲义》第10499卷，第12–27页。Springer，2017年。35 Guolin Ke、Qi Meng、Thomas Finley、Taifeng Wang、Wei Chen、Weidong Ma、Qiwei Ye 和 Tie-Yan Liu。Lightgbm：一种高效的梯度提升决策树。载于 NIPS，第3146–3154页，2017年。36 Artur Korniłowicz 和 Christoph Schwarzweller。Mizar 中的计算机与算法。《机械化数学及其应用》，4(1)：43–50，2005年。37 Laura Kovács 和 Andrei Voronkov。一阶定理证明与 Vampire。载于 Natasha Sharygina 和 Helmut Veith 编，CAV，《LNCS》第8044卷，第1–35页。Springer，2013年。38 Guillaume Lample 和 François Charton。符号数学的深度学习。载于第八届国际学习表征会议，ICLR 2020，埃塞俄比亚亚的斯亚贝巴，2020年4月26日至30日。OpenReview.net，2020年。URL：https://openreview.net/forum?id=S1eZYeHFDS。39 Miroslav Olsák、Cezary Kaliszyk 和 Josef Urban。用于自动推理的性质不变嵌入。载于 Giuseppe De Giacomo、Alejandro Catalá、Bistra Dilkina、Michela Milano、Senén Barro、Alberto Bugarín 和 Jérôme Lang 编，ECAI 2020——第二十四届欧洲人工智能会议，《人工智能与应用前沿》第325卷，第1395–1402页。IOS Press，2020年。URL：https://doi.org/10.3233/FAIA200244，doi:10.3233/FAIA200244。40 David W. Opitz 和 Richard Maclin。流行的集成方法：一项实证研究。《人工智能研究杂志》，11：169–198，1999年。URL：https://doi.org/10.1613/jair.614，doi10.1613/jair.614。41 Jelle Piepenbrock、Josef Urban、Konstantin Korovin、Miroslav Olsák、Tom Heskes 和 Mikolas Janota。机器学习遇见 Herbrand 宇宙。CoRR，abs/2210.03590，2022年。42 Bartosz Piotrowski 和 Josef Urban。ATPboost：在 ATP 反馈下学习二元设置中的前提选择。载于 Didier Galmiche、Stephan Schulz 和 Roberto Sebastiani 编，《自动推理——第九届国际联合会议》，IJCAR 2018，作为联合逻辑会议 FloC 2018 的一部分举行，英国牛津，2018年7月14日至17日，论文集，《计算机科学讲义》第10900卷，第566–574页。Springer，2018年。URL：https://doi.org/10.1007/978-3-319-94205-6_37，doi:10.1007/978-3-319-94205-6_37。43 Bartosz Piotrowski 和 Josef Urban。通过循环神经网络进行有状态前提选择。载于 LPAR，《EPiC 计算系列》第73卷，第409–422页。EasyChair，2020年。44 Bartosz Piotrowski、Josef Urban、Chad E. Brown 和 Cezary Kaliszyk。神经网络能学习符号重写吗？CoRR，abs/1911.04873，2019年。45 John Alan Robinson 和 Andrei Voronkov 编。《自动推理手册》（共2卷）。爱思唯尔和麻省理工学院出版社，2001年。

46 Stephan Schulz。《E——一款天才定理证明器》。《人工智能通讯》，15(2-3):111–126，2002年。47 Stephan Schulz。利用特征向量索引实现简单高效的子句包含。在《自动推理与数学》中，《计算机科学讲义》第7788卷，第45–67页。Springer，2013年。48 Stephan Schulz。系统描述：E 1.8。在 Kenneth L. McMillan、Aart Mid- deldorp 和 Andrei Voronkov（编）编著的《LPAR》中，《LNCS》第8312卷，第735–743页。Springer，2013年。URL: http://dx.doi.org/10.1007/978-3-642-45221-5_49, doi:10.1007/978-3-642-45221-5 49。_ 49 Stephan Schulz。系统描述：E 1.8。在 Ken McMillan、Aart Middeldorp 和 Andrei Voronkov（编）编著的《第19届LPAR会议论文集》，斯泰伦博斯，《LNCS》第8312卷，第735–743页。Springer，2013年。50 Stephan Schulz、Simon Cruanes 和 Petar Vukmirović。更快、更高、更强：E 2.3。在 Pascal Fontaine（编）编著的《第27届CADE会议论文集》中，纳塔尔，巴西，《LNAI》第11716号，第495–507页。Springer，2019年。51 Stephan Schulz 和 Geoff Sutcliffe。饱和一阶定理证明器的证明生成。在 David Delahaye 和 Bruno Woltzenlogel Paleo（编）编著的《关于证明，证明惠及所有人》中，《数理逻辑与基础》第55卷，第45–61页。College Publications，英国伦敦，2015年1月。52 Martin Suda。在从历史中学习的同时改进ENIGMA风格的子句选择。在 André Platzer 和 Geoff Sutcliffe（编）编著的《自动演绎——CADE 28——第28届国际自动演绎会议，虚拟活动，2021年7月12日至15日，论文集》中，《计算机科学讲义》第12699卷，第543–561页。Springer，2021年。URL: https://doi.org/10.1007/978-3-030-79876-5_31, doi:10.1007/978-3-030-79876-5_31。53 Martin Suda。拥有大脑的吸血鬼是一个优秀的ITP工具。在 Boris Konev 和 Giles Reger（编）编著的《组合系统前沿——第13届国际研讨会，FroCoS 2021，英国伯明翰，2021年9月8日至10日，论文集》中，《计算机科学讲义》第12941卷，第192–209页。Springer，2021年。URL: https://doi.org/10.1007/978-3-030-86205-3_11, doi:10.1007/978-3-030-86205-3_11。54 Tanel Tammet。迈向高效的包含。在CADE中，《计算机科学讲义》第1421卷，第427–441页。Springer，1998年。55 J. Urban。为一阶定理证明器翻译Mizar。在 A. Asperti、B. Buchberger 和 J.H. Davenport（编）编著的《第2届数学知识管理国际会议论文集》中，《LNCS》第2594号，第203–215页。Springer，2003年。56 Josef Urban。MPTP——动机、实现与首次实验。《自动推理杂志》，33(3-4):319–339，2004年。 doi:10.1007/s10817-004-6245-1。57 Josef Urban。MPTP 0.2：设计、实现与初步实验。《自动推理杂志》，37(1-2):21–43，2006年。58 Josef Urban。BliStr：盲策略生成器。在 Georg Gottlob、Geoff Sutcliffe 和 Andrei Voronkov（编）编著的《全球人工智能大会，GCAI 2015，格鲁吉亚第比利斯，2015年10月16日至19日》中，《EPiC 计算系列》第36卷，第312–319页。Easy- Chair，2015年。URL: http://www.easychair.org/publications/paper/BliStr_The_Blind_Strategymaker。59 J osef Urban 和 Jan Jakubuv。首批神经网络猜想生成数据集与实验。在 Christoph Benzmüller 和 Bruce R. Miller（编）编著的《智能计算机数学——第13届国际会议，CICM 2020，意大利贝尔蒂诺罗，2020年7月26日至31日，论文集》中，《计算机科学讲义》第12236卷，第315–323页。Springer，2020年。URL: https://doi.org/10.1007/978-3-030-53518-6_24, doi:10.1007/978-3-030-53518-6_24。60 Josef Urban、Geoff Sutcliffe、Petr Pudlák 和 Jiří Vyskočil。MaLARea SG1——带语义引导的自动推理机器学习器。在IJCAR中，第441–456页，2008年。61 Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。注意力机制就是你所需要的一切。CoRR，abs/1706.03762，2017年。URL: http://arxiv.org/abs/1706.03762, arXiv:1706.03762。

62 Qingxiang Wang、Cezary Kaliszyk 和 Josef Urban。非正式数学到形式化数学的神经翻译首次实验。在 Florian Rabe、William M. Farmer、Grant O. Passmore 和 Abdou Youssef（编）编著的 第*11*届智能计算机数学国际会议（*CICM 2018*）中，第11006卷，*LNCS*，第255–270页。Springer，2018年。URL： https://doi.org/10.1007/978-3-319-96812-4_22， doi:10.1007/978-3-319-96812-4_22。

A 进一步的证明细节与讨论

A.1 NEWTON:72 的进一步证明细节

我们展示 NEWTON:72 –对于每个自然数，都存在一个更大的素数– 的完整证明，但不包括图 2 中的子句化步骤。逐字的 ENIGMA 证明可在线获取。[29] 原始猜想如下。请注意，两次陈述素性是 Mizar 编码造成的结果。

```
fof(t72_newton,conjecture,(
! [X1] : ( natural(X1)=> ? [X2] :
( natural(X2)& prime(X2)
& prime(X2)& ~ le(X2,X1) ) ) ),
file(t72_newton,t72_newton)).
```

然后对该猜想取否定。并非对于每个（自然数）X1，都存在一个满足 $X2 > X1$ 的素数（自然数）X2。

```
fof(c_0_54,negated_conjecture,(~ ( ! [X1] : ( natural(X1)
=> ? [X2] : ( natural(X2)& prime(X2)& prime(X2)
& ~ le(X2,X1) ) ) ) ),
inference(assume_negation,[status(cth)],[t72_newton])).
```

之后，X1（存在一个更大的素数所针对的那个数）被斯科伦化为 esk1_0。由于现在它是否定的，我们得到的不是“存在一个 X6”，而是“对于所有 X6”，其中 X6 不是素数，或者 X6 是 $\leqslant$ 那个数。

```
fof(c_0_64,negated_conjecture,( ! [X6] : ( natural(esk1_0)& ( ~ natural(X6)| ~ prime(X6)
| ~ prime(X6)| le(X6,esk1_0) ) ) ),
inference(shift_quantors,[status(thm)],[inference(skolemize,[status(esa)],...)]).
```

在下面的推理中，证明器首次合成了项 $x1! + 1$“plus(fact1(X1),n1)”（从现在起称为见证 ）。

[29] https://bit.ly/3Z2iXo3

```
cnf(c_0_96,plain,( esk60_1(plus(fact1(X1),n1)) = n0
| esk60_1(plus(fact1(X1),n1)) = n1| ~ le(esk60_1(plus(fact1(X1),n1)),X1)
| ~ le(n2,plus(fact1(X1),n1))| ~ natural(plus(fact1(X1),n1))| ~ natural(X1) ),
inference(csr,[status(thm)],[inference(csr,[status(thm)],[inference(...)])]).
```

接下来便得到这里的猜想：

```
cnf(c_0_106,negated_conjecture,( esk60_1(plus(fact1(esk1_0),n1)) = n1
| esk60_1(plus(fact1(esk1_0),n1)) = n0
| ~ le(n2,plus(fact1(esk1_0),n1))| ~ natural(plus(fact1(esk1_0),n1)) ),
inference(csr,[status(thm)],[inference(cn,[status(thm)]...)]).
```

其核心意思是：

```
    witness=1
 or witness=0
 or witness>2 ("not witness<=2")
 or witness is not a natural number
```

确定这一点后，证明中便以相对直接的方式处理了这四种情况。

A.2 NEWTON:72：机器学习引导演绎综合的一个示例

上述非平凡见证 X1!+1 在证明中的综合，其背后一个简化的核心是以下两个 Mizar 定理 NEWTON:39 和 NEWTON:41的组合：

```
theorem Th39: :: NEWTON:39for m,
 n being Nat st m <> 1 & m divides n holds not m divides n + 1

theorem Th41: :: NEWTON:41for j,
 l being Nat st j <= l & j <> 0 holds j divides l !
```

进行归结时，合一实际上会导致在 NEWTON:39中用 l! 替换 n，从而综合出实例 l!+1 ，这正是所需的见证（j 整除 l!，因此它确实不能整除 l!+1）。

这说明了（A）（学习引导的）基于归结的综合与（B）纯粹的学习引导综合或枚举有何不同。在（A）中，学习器会推荐应进行归结的子句（亲本引导），或者推荐与一组已选子句相关的单个（给定）子句（标准的梯度提升决策树/图神经网络引导）。这种推荐未必完全理解归结会产生什么结果。它很可能只是一种“直觉”，认为组合这两个事实或许会很有意思；而实际通过演绎综合出的见证，则会让人感到意外，因为它是在这里由逻辑演算“计算”出来的。

而在（B）中，我们要求学习者（例如语言模型 [59], GNN2RNN [41], 或蒙特卡洛树搜索引导的综合框架 [17]）“通过观察情境，自行想出该项”。在数学家已经见过并识别出这种模式或技巧，或进行类比、如同在 GNN2RNN 中那样迁移术语的情境下，这种做法是合理的。模式识别和类比的数据库及能力越完善、越强大，数学家就越有把握直接想出见证。

因此，方法（A）似乎更具探索性，也更适用于新的问题求解情境：通过演绎或计算来强化较弱的直觉，从而可能带来新颖且出人意料的发现和值。相比之下，方法（B）似乎在已知情境中更有用，更多依赖于对可能值的直接且足够确信的召回率和记忆。方法（A）更多是对应该采用何种方法或方向的直觉。例如，我们可能觉得应该以某种方式结合结果 X、Y 和 Z，计算某个特定的导数/积分，求解一组方程/约束，寻找一个归纳假设，等等，但并不知道结果。而方法（B）更多是对解本身的直接直觉或回忆，而不是对导出该解的过程的直觉或回忆。

Type-A 与 Type-B 机器学习引导的问题求解方法之间可能再次存在反馈回路。Type-A 更为间接，其工作方式是指向并调用进一步的不确定过程，并利用这些过程进行搜索。如果这种方法在某类特定情形下取得了相当程度的成功，置信度就会提高，并可能最终促成更直接的 Type-B 猜测。而 Type-B 猜测一旦成功，可能会缩短 Type-A 搜索的部分过程，从而也使这些搜索更加成功，如此循环往复。

A.3 更多示例证明图

为了让读者了解第7节所述更高级 ENIGMA 证明的复杂性，我们在此给出其中三个证明的图形表示（图3、4、5）。我们再次省略其中的子句化部分。由于这些图规模很大，包含数百个节点（这些节点可能还由相当复杂的子句构成），我们也建议感兴趣的读者访问我们的网页，在那里可以以交互方式查看这些图的 SVG图像。[1,2,3,4,5]

[1] http://grid01.ciirc.cvut.cz/~mptp/enigma_prf_graph/t72_newton_nice.proof1.svg
[2] http://grid01.ciirc.cvut.cz/~mptp/enigma_prf_graph/t31_borsuk_5.proof1.svg
[3] http://grid01.ciirc.cvut.cz/~mptp/enigma_prf_graph/t14_fdiff_8.proof1.svg
[4] http://grid01.ciirc.cvut.cz/~mptp/enigma_prf_graph/t86_sincos10.proof1.svg
[5] http://grid01.ciirc.cvut.cz/~mptp/enigma_prf_graph/t22_ideal_1.proof1.svg

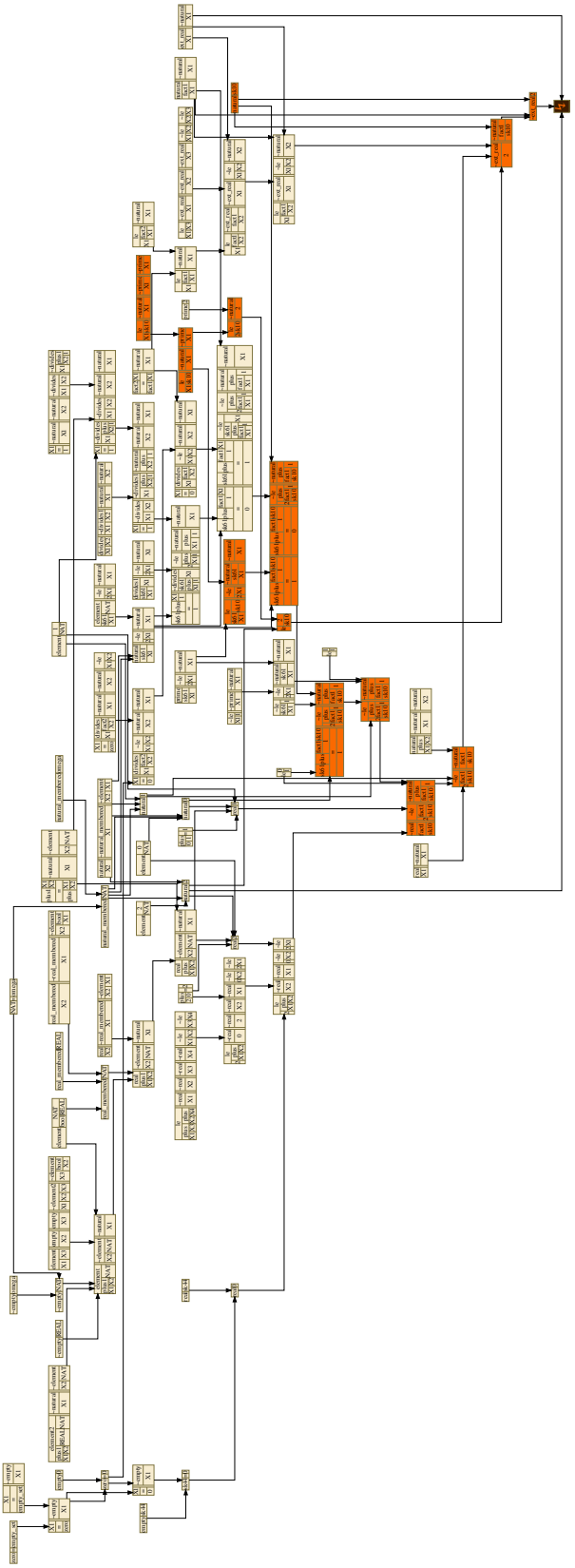

Figure 2 ENIGMA's proof of NEWTON:72.

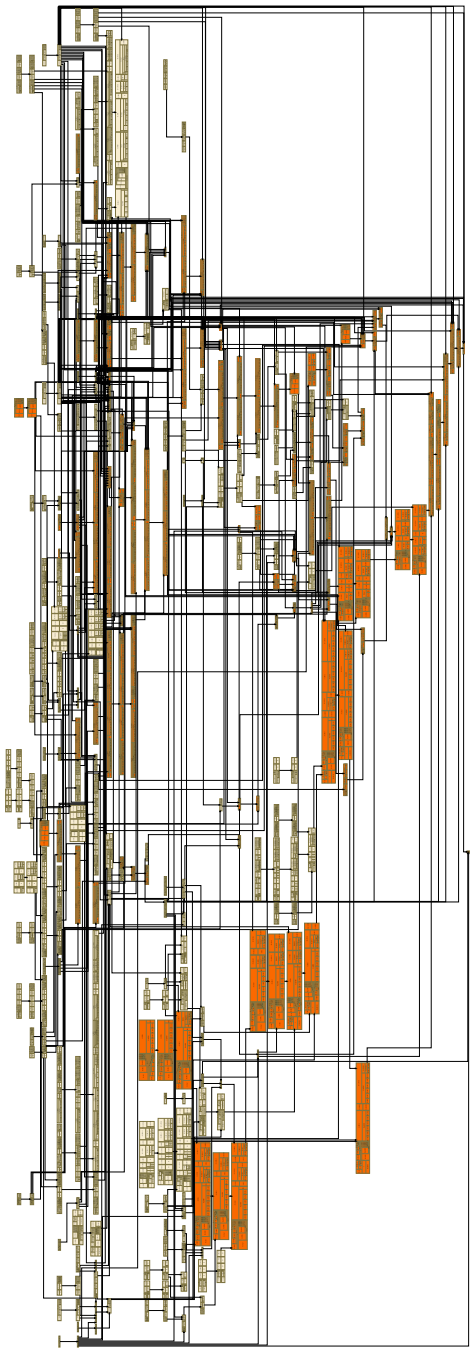

Figure 3 ENIGMA's proof of `FDIFF_8:14`.

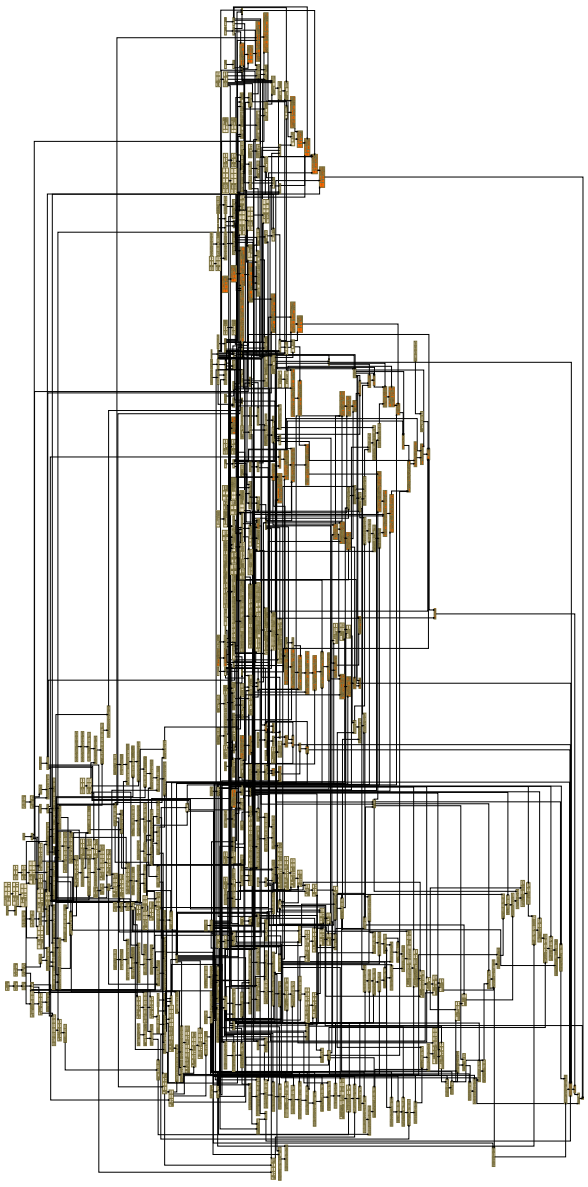

Figure 4 ENIGMA's proof of `SINCOS10:86`.

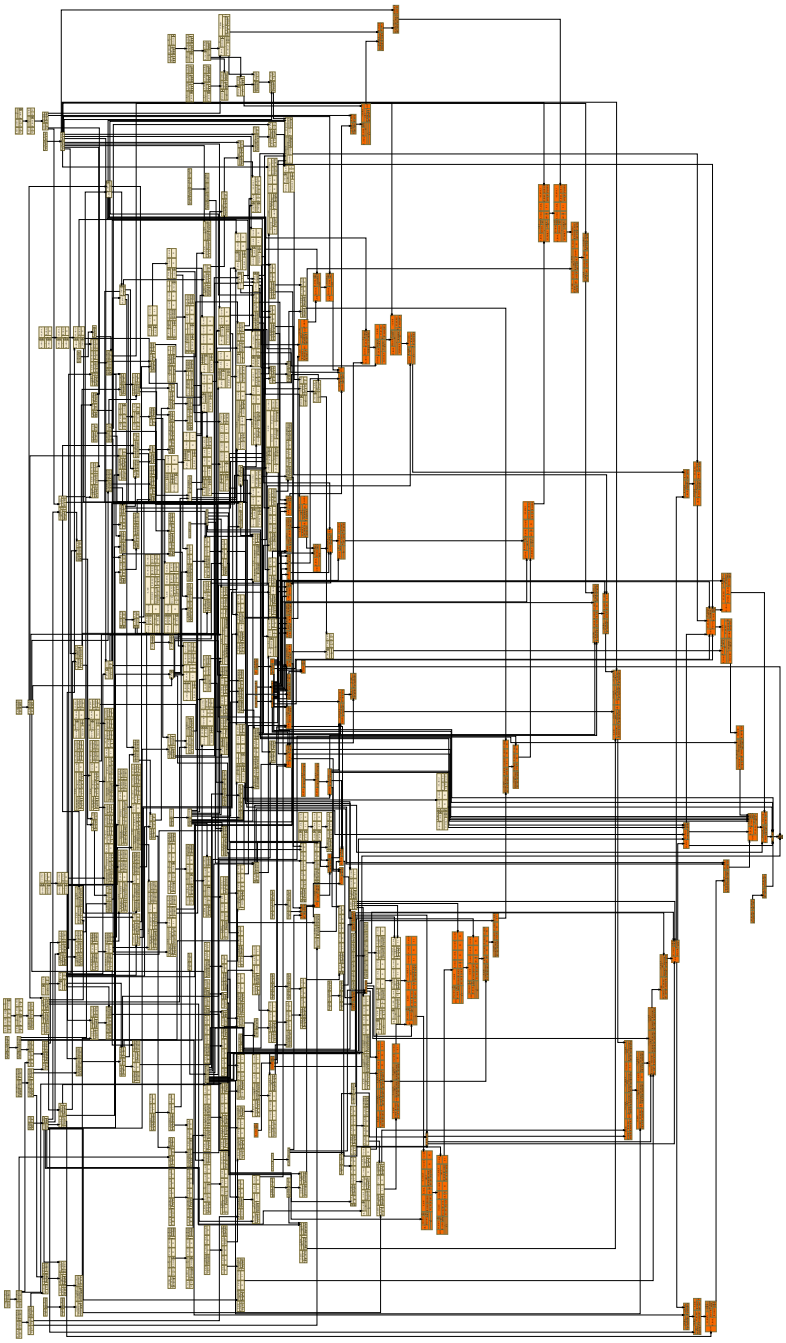

Figure 5 ENIGMA's proof of `BORSUK_5:31`.