清華大學

综 合 论 文 训 练

平面几何自然语言的形式化、自动构图和定理集构建

系　　别：　致理书院

专　　业：　数学与应用数学

姓　　名：　宋 晨 东

指导教师：　包 承 龙　　副教授

二〇二五年六月

关于论文使用授权的说明

本人完全了解清华大学有关保留、使用综合论文训练论文的规定，即：学校有权保留论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

作者签名：宋晨东　　导师签名：包承龙

日　　期：2025.6.12　　日　　期：2025.6.12

摘　要

本研究围绕平面几何的自动形式化展开，探索了自然语言描述、限制性形式化语言与构造性形式化语言之间的相互转化方法，并实现了从自然语言几何题自动生成形式化表示和图形绘制的流程。论文首先梳理了 AlphaGeometry 等现有形式系统的特点，提出了一种新的限制性到构造性语言的自动转换算法，搭建了自动绘图系统。随后，利用大语言模型（LLM）进行自然语言到形式语言的自动翻译，并设计了严格的语法、数值、语义三重检验流程来保证翻译的正确性。本研究还搭建了验证模型形式化能力的基准测试 IMO_100 和新的几何形式化数据集 Numina_Geometry，并通过 AlphaGeometry 等推理引擎在数据集上进行测试，验证了数据集的质量和模型在形式化任务中的能力。通过对成功与失败案例的系统分析，进一步总结了大模型在几何形式化任务中存在的挑战和改进空间。

关键词：平面几何；形式化语言；自动形式化；大语言模型；AlphaGeometry

Abstract

This thesis focuses on the autoformalization of plane geometry, exploring methods for automatic conversion method between natural language descriptions, restrictive formal language, and constructive formal language. It implements a pipeline for automatically generating formal representations and geometric drawings from natural language geometry problems. The paper first reviews the characteristics of existing formal systems such as AlphaGeometry, then proposes a novel autoformalization algorithm from restrictive to constructive geometry formal language and establishes an automatic drawing system. Subsequently, large language models (LLMs) are employed for automatic translation from natural language to formal language, accompanied by a rigorous verification process — syntactic, numeric, and semantic—to ensure the correctness of translation. We also construct a new benchmark, IMO_100 to evaluate the capability of autoformalization, and a geometry exercise dataset, Numina_Geometry, and conducts benchmarking using AlphaGeometry to validate the dataset quality and the models' capabilities on formalization tasks. Through systematic analysis of both successful and failed cases, the thesis further summarizes challenges and potential improvements for large models in geometric formalization tasks.

Keywords: plane geometry; formal language; autoformalization; large language model; AlphaGeometry

目 录

插图和附表清单

第 1 章　引　言

1.1　研究背景

自人工智能诞生之初，数学家和 AI 研究者们就梦想着构建能够自动进行严谨数学推理的人工智能系统。由此诞生一个新的数学统计人工智能领域——AI4Math，这一领域在大语言模型（LLM）诞生后取得了蓬勃的发展。[1] 对于数学研究者而言，AI 的自动化推理能够更广泛的探索未知的数学定理库。而对于 AI 研究者而言，做数学题和推导数学定理是增强 AI 推理能力的重要一环。

AI4math 目前主要分成两种研究路径——形式化的和非形式化的。非形式化方法法是在数学数据（例如 arXiv 论文和 MathOverflow 的网页）上对 LLM 进行预训练，然后在数学题数据集上对模型进行微调，这种方法已经在广泛使用的基准测试（如 GSM8K 和 MATH）以及 AIMO Progress Price 中取得了显著的成功。然而，由于高质量数学数据缺失和大模型本身的幻觉，非形式化方法只能局限于不超过 AIME 水平的高中数学。

与之相对应的，形式化数学语言是一种通过精确的符号和规则来表达数学概念和推理的语言。形式化语言不依赖于模糊的词义，而是依赖严格的符号、定义和公理系统来进行推理，并能够利用机器自动检验定理推导的正确性。形式化的数学定理证明有两个关键步骤：自动定理证明和自动形式化。自动定理证明旨在利用计算机程序自动验证数学命题的正确性，而自动形式化是指将自然语言自动转换成某种形式化语言的任务。[2] 自动形式化可以为训练自动定理证明的 AI 代理提供大量数据，减少人工数据标注的成本，能够快速规模化以提高 AI 的性能，是目前合成数据的主流策略。

不同数学领域的形式化语言和难度都不同。本文将聚焦于研究一个细分领域———平面几何定理证明的自动形式化。选取平面几何作为研究对象，是因为平面几何有封闭严谨的公理推导体系，且平面几何的自然语言叙述相对固定。

1.2　平面几何形式化与公理体系进展

从两千多年的欧几里得《几何原本》开始，几何学一直作为严密推理的基石，深刻影响了数学与逻辑的发展。自 20 世纪以来，计算机程序和机器自动定理证明的出现，极大地推动了欧几里得几何形式化与自动推理的发展，如 LeanEuclid[3] 将《几何原本》定理和图形推导相结合，在几何公理系统 E[4] 的基础上用 Lean 编程

语言和可满足性模理论 (Satisfiability Modulo Theories) 搭建了几何自动验证形式系统。

一般的几何的推理系统通常分为两大类：综合推理（synthetic deduction）和代数计算（algebraic computation）。

在综合推理方法中，演绎数据库方法（deductive database approach）[5] 使用前向推理系统地枚举几何规则，并在数据库中推导新事实，直至达到定理闭包。尽管这种方法在推导几何性质方面十分有效，但在处理需要代数运算的各种问题时表现不佳。另一方面，代数方法，如 Gröbner-基方法[6] 和吴方法（Wu’s method）[7]，则将几何性质转化为基于点坐标的多项式方程组，并通过代数手段求解。虽然代数计算在证明几何等式非常强大，但它们通常生成人类难以理解的复杂证明过程。

最近，结合深度学习的几何系 AlphaGeometry[8] 成为几何定理证明的一个重要的里程碑，它将大语言模型与一种新的逻辑推导引擎 DDAR 结合，利用大预言模型预测辅助点，通过树搜索的 DDAR 求解几何题，该系统成为首个在国际数学奥林匹克（IMO）题目上达到银牌水平的系统。DDAR 结合了演绎数据库与角度、比例、距离追踪的基础代数技术，实现了高级几何推理。

除了 AlphaGeometry，TongGeometry[9] 采用树搜索算法和强化学习策略，在几何定理证明基准测试 IMO_30 上的成绩超越了 AlphaGeometry。Newclid[10] 改进了 AlphaGeometry 的 DD+AR 推理系统，并使输入输出对用户更友好。TrustGeoGen[11] 构建了用于生成数学几何问题的可扩展数据引擎，确保了多模态信息的一致性和逻辑连贯性。而基于 AlphaGeometry 构建的 AlphaGeometry2[12] 也对多个技术细节做了改进，大幅提升了其表达性、搜索速度和生成辅助点的能力。在附录的文献翻译部分将详细介绍 AlphaGeometry2。

AlphaGeometry 是非常强的自动证明器，其输入为格式规范的构造性形式化语言，而在其自动定理合成和证明推理的环节，使用的均是限制性形式化语言，这对于人类使用、理解其运作规律和规模化数据集构成了挑战。在本研究中，我们将基于 AlphaGeometry 的形式系统，研究上述两种形式化语言和自然语言的关系，讨论其相互转化的方法，并利用自然语言平面几何数据集合成形式化语言的数据集。本研究的主要贡献有：

- 基于两种形式化语言的特点，使用“删点”策略，提出全新的限制性形式化语言到构造性形式语言的转换算法, 在机器合成的 65K 限制性形式化语言的数据集中达到 80.8% 的翻译成功正确率。
- 基于联立坐标求解和搜索回溯，搭建平面几何自动构图系统，在 200 道复杂形式化语言叙述的构图任务中达到 61.5% 的准确率。

- 搭建基于语法检验、数值检验、语义检验的平面几何自动形式化正确性验证管线，构造形式化任务基准测试 100 题，探索提示词中样例数和细微指令对翻译效果的影响。
- 将开源数据集 Numina_Math 通过自动形式化管线获得 2720 道高质量形式化几何习题，在 AlphaGeometry 上进行测试，比较不同几何数据集的特点。

第 2 章　平面几何形式化语言相互转换及绘图

2.1　平面几何形式化语言的公理系统

本研究所采用的形式系统为基于 AlphaGeometry 采用的形式语言，其可以分为“限制性”和“构造性”两种类型。

2.1.1　限制性形式化语言

限制性形式化语言用定理的形式描述点的位置关系，所有的定理由点和点满足的条件谓词构成，形如 `<premise> <arg1> <arg2> <arg3> …`。我们在B.1给出了所有的谓词和其详细的表述。这套形式系统具有如下特点：

- 该系统中基本对象仅有点，直线用由两个点确定，圆由圆心和圆上的一个点来确定，所有几何前提的参数只能是点。
- 该系统是一个“无向”(unordered) 的形式系统[5]。在该系统中没有射线、方向的概念，也不会区分点在直线上的顺序、点在直线的同侧异侧等关系，而仅仅关心垂直、平行、相等、共圆等定量关系。
- 该系统中角度使用“全角”(full angle) 来刻画[5]。对于两条直线 l, m, 定义其全角 $\angle[l, m]$，若存在一个旋转 K ，使得 $K(l)$ 与 u 平行，且 $K(m)$ 与 v 平行，则两个全角 $\angle[l, m]$ 和 $\angle[u, v]$ 相等。若 A, B 和 C, D 分别为 l 和 m 上的不同点，则 $\angle[l, m]$ 亦可表示为 $\angle[AB, CD]$, $\angle[BA, CD]$, $\angle[AB, DC]$ 和 $\angle[BA, DC]$ 。在“全角”形式系统里角的描述没有了对角射线指定方向的困难，并使大量定理的表述变得简洁，如”`cyclic A B P Q` $\Longleftrightarrow$ `eqangle P A P B Q A Q B` 一个命题表现了共边同侧对角等或对角互补等价于四点共圆的两层含义，而不用考虑区分点在圆上的坐标位置关系。作为代价，“全角”的系统由于角相等和互补的模糊性，会让部分定理，如三角形全等的 SAS 判定出现困难。

2.1.2　基于限制性形式化语言的演绎库和自动定理合成

演绎数据库是指能够利用数据库本身的数据进行逻辑推导的数据库。在平面几何中，演绎数据库能够利用给定的几何定理，从当前图形命题中推导出新的命题。定理库中的每一个“定理”都具有如下形式：

$P_1(x), \cdots, P_k(x) \Rightarrow Q(x)$　表示

$$\forall x \left[\left(P_1(x) \wedge \cdots \wedge P_k(x)\right) \Rightarrow Q(x)\right]$$

自动证明的机器可以在已知结论中对每一个已知的定理进行模式匹配并不断进行搜索，增添已知结论，最终达到“推理闭包”，即在不添加辅助点的情况下，所有图上已知的位置关系都已经发掘完成的状态。

AlphaGeometry 中应用了 43 条几何定理，在附录中给出（见B.2）。其中部分谓词（如”ncoll”) 和定理推导（如三角形 SAS 的判定）并不单纯依赖于限制性形式化语言本身，而需要借助几何图形的数值坐标等语义来判断。

有了演绎数据库，我们可以将随机生成的前提条件作为演绎数据库的输入，通过演绎数据库进行自动推理，得到相应的几何结论。从而提取习题。同项目组的汪思然同学曾利用该方法合成了拥有 69047 条几何习题数据的数据库 sr_geometry.

2.1.3 构造性形式化语言

称一个几何命题是“线性构造的”，如果一个几何命题中的点可以按顺序列出，使得序列中的每个点都可以由序列中前面的点唯一确定。“线性构造”的命题作有以下几类基本操作：

1. 取一个自由点。
2. 取直线或圆上的任意点。
3. 取两条直线的交点。
4. 已知直线和圆或圆与圆的一个交点，取其另一个交点。

AlphaGeometry[8]给出了共 68 种构造性前提的例子，每一个构造性形式化前提都由一个描述点，一个或两个谓词和若干个参数点构成。其每一条语言都能够被限制性形式化语言来等价表述。如构造性前提 `a = on_line a b c, on_bline a d e` 中 a 是描述点，b, c, d, e 是参数点而 on_line 和 on_bline 是谓词，表示作点 A 使得 A 是 DE 的垂直平分线和直线 BC 的交点。注意到每一个构造性形式化语言都能够被限制性形式化语言表述，如上述命题可以表达成 `coll a b c; cong a d a e`, 即用“A，B，C”共线且 $AD = AE$ 来表述。具体的构造性形式化前提见B.2。由构造性形式化前提连接成的平面几何命题称为构造性命题。我们通过如下定义来描述几何形式化语言的规范性：

定义：称一个构造性命题是“符合语法规范的”，当且仅当其满足：

- 该命题的结构为 $p_1; p_2; \cdots p_n ? c$, 其中 $p_1; p_2; \cdots p_n$ 为符合语法规范的构造性前提，c 是符合语法规范的限制性形式化语言。每个 p_i 包含一个描述点，至多 2 个谓词和至多 2 个参数集。。
- 对于任意 $i = 1, 2 \cdots, n$, 第 i 条构造性命题的前提条件 p_i 的所有参数点包含于 $p_1, p_2, \cdots, p_i$ 描述点构成的集合。

例如，下面的句子是“符合语法规范的”构造性形式化语言：

```
a b c = triangle a b c; o = circle o a b c; d = on_circle d o a
   ; p = on_aline p b c a b d, on_aline p d c a d b ? cong a p
   c p
```

相比较于限制性形式化语言，构造性的形式化语言能够更准确地反映几何图形的信息，也更贴合人类自然语言的表述习惯。构造性形式化语言的构建过程和尺规作图的过程非常类似，其表达性和人类作图的范畴也基本一致。

2.2 形式语言的自动转化

为了测试合成数据集 sr_geometry 的质量，我们需要将数据集当中的形式化几何题从限制性的形式化命题转化成构造性的命题，以便后续画图定性观察和通过 AlphaGeometry 定量观察。

我们将该问题进行数学建模：给定点集 $A = \{a_1, a_2, \cdots, a_k\}$ 和 n 条限制性命题 $S = \{s_1, s_2, \cdots, s_n\}$, $s_i = \{p_i, a_{i1}, a_{i2}, \cdots, a_{in_i}\}$，其中 p_i 是第 i 个条件的谓词，n_i 是第 i 个条件的参数数量 ($n_i \geqslant 3$)，$a_{ij} \in A$ 代表第 i 个定理条件的第 j 个参数。我们需要在 A 的置换里找一个元素 $\tau = (a_{\sigma(1)}, a_{\sigma(2)}, \cdots, a_{\sigma(k)}) \in \sigma(A)$，以确定构造性语言在图上绘图点的顺序, 对于每个时刻 j, 记此时在纸上的前 j 个点构成的点集为 $A(\tau, j) = \{a_{\sigma(1)}, a_{\sigma(2)}, \cdots, t_{\sigma(j)}\}$

我们记 $P(\tau, j) = |\{i | a_{i1}, a_{i2}, \cdots, a_{in_i} \in A(\tau, j)\}|$, 即此时能够被所有点表达的点的命题的数量，称一个置换 τ 是“合理”的, 如果对于 $\forall j \in \{2, 3, \cdots, k\}$, 都有 $P(\tau, j) - P(\tau, j-1) \leqslant 2$, 即画一个新的点至多由两个限制条件确定。称一个限制性形式化语言 (A, S) 是“可翻译的”，如果该限制性形式化语言的点集 A 存在一个“合理”的置换 τ。

称 A 中的点 a_j 对于命题集 $S_0 \subset S$ 中是“可删”的，如果 t_j 在 S_0 当中出现不超过两次。设 $t_j \in S_j \subset S_0$, 此时我们可以将 a_j 作为最后一个翻译的点，其必能够被画出。因此包含 t_j 的命题不会影响最终命题的可翻译性，从而 (A, S) 是可翻译的当且仅当 $(A - \{t_j\}, S - S_j)$ 是可翻译的。称上述删去 t_j 的一次操作为“删点”。我们有如下断言：

命题：一个几何命题是可翻译的，当且仅当存在一个点的置换 σ, 使得在该置换规定的顺序下，删点操作能够不断进行，直到 $A = S = \varnothing$。

证明 假设删点操作能够不断进行直至空集，则依照删点的顺序构造出的置换是“合理”的，因此能够给出一条合法的构造性形式化语言，于是该命题可以被翻译。反之，假设存在 $A' \subset A, S' \subset S$ 使得 (A', S') 没有任何可删的点。如果此时有一个合理的置换 σ, 则在 σ 给出的顺序下 A' 中最后一个点有至少 3 个限制性条件，否则该点在 (A', S') 下可删，因此该置换不符合“合理”的条件，矛盾！从而假设不

成立，即 (A, S) 不可翻。 ∎

需要注意的是一个限制性几何命题“可翻译”仅仅代表语法上能用两个条件限制一个点的位置来描述该命题，而并不代表该能通过尺规作图作出，比如假设点 A, B, C, D 已知，需要作点 X 使得 $\angle AXB = \angle CXD$, 这在尺规作图的范畴下是不可能的。因此，一个限制性几何命题能否翻译成构造性几何命题, 不仅仅需要上述形式逻辑的检验，还需要通过语义来具体判断。

自然地，上述“删点”过程给出了一个由限制性形式化语言到构造性形式化语言转换的算法:

算法 2.1 限制性形式化语言转化成构造性形式化语言

Require: 点集 A，限制性命题集 S
Ensure: 构造性命题 Q
  while $A \neq \varnothing$ **do**
    for $t \in A$ **do**
      $S_0 = \{s \in S | \text{t 是命题 s 的参数}\}$
      if $|S_0| \leqslant 2$ **then**
        $A \leftarrow A - \{t\}$
        $S \leftarrow S - S_0$
        $Q \leftarrow Q +$ s 对应的构造性形式化语言
      end if
    end for
  end while

我们使用该算法在 sr_geometry 的 69047 条几何数据上进行测试，结果有 86.1% 的数据能够利用该方法翻译成构造性形式化语言。以下是一个翻译成功的案例, 程序大量改变了变量的顺序，先翻译复杂的点再翻译简单的点，成功得到了一个构造性形式化语言：

```
"constrained_formal_statement":cong a b b c; perp a b b c; para
    a d b c; para a b c d; perp a c c e; cong a c c e; para a c
    e f; para a f c e; coll h f d; cong h d h f; eqangle a c a
   n a n a h; eqangle h n a h c h h n; coll c k h; perp k n c h
   ; coll c a m; perp a c n m? eqangle h n c h a n k m
"constructive_formal_statement"
e = free e; a = free a; c = on_dia c a e, on_bline c a e; f =
   on_pline f e a c, on_pline f a c e; b = on_bline b a c,
   on_dia b a c; d = on_pline d a b c, on_pline d c a b; h =
   on_line h f d, on_bline h d f; n = angle_bisector n c a h,
   angle_bisector n a h c; m = on_line m c a, on_tline m n a c;
    k = on_line k c h, on_tline k n c h ? eqangle h n c h a n k
    m
```

以下是翻译过程中一个失败的样例：

```
"constrained_formal_statement":perp a b a c; coll d b a;
   eqangle d c d k d k d b; eqangle b c c k c k c d; coll d h b
   ; perp b d k h; coll c d i; perp i k c d; coll c b j; perp c
    b k j? eqangle b c h j i j h i
```

```
"constructive_formal_statement": Can't translate!
```

在上述样例中，对于条件”eqangle d c d k d k d b”，在翻译中必须先作出 d 点才能作出 c,k,b 点，因此会对上述理论算法产生额外的语义约束，目前该算法无法处理这类约束，翻译失败。

为了应对此类问题，我们将该算法筛选删去点的步骤进行随机化处理，即每次运行随机抽取一个符合要求的点删去。在随机化处理后，每道题重复 100 次实验，能通过翻译的准确率将高达 94.3%。我们对筛选出的正确的 65K 道平面几何数据去重后，得到 15584 道不同题面的平面几何问题。将这些题目作为 AlphaGeometry 的输入，13353 道题目能够通过 AlphaGeometry 的题目检验系统系统，判定为正确有效的平面几何习题。该系统总体翻译的成功准确率达到 80.8%.

2.3 利用构造性形式化语言自动绘图

根据平面几何形式化语言绘图有多种方式，Krueger 等人（2021）[13] 曾构造过平面几何自动绘图系统 GMBL，可将构造性的形式化语言转化成多项式约束条件，并把所求解的问题规约为数值优化问题，通过梯度下降获得对应点的坐标。而本节我们探讨利用构造性形式化语言，采用更直观、具体的方法来绘图。在本节，我们搭建了一个利用构造性平面几何命题自动绘图的算法，该算法的输入为一个符合语法规范的构造性命题，输出为依照该命题前提条件绘制的几何图形, 下图为构造性形式化语言和对应图的示例：

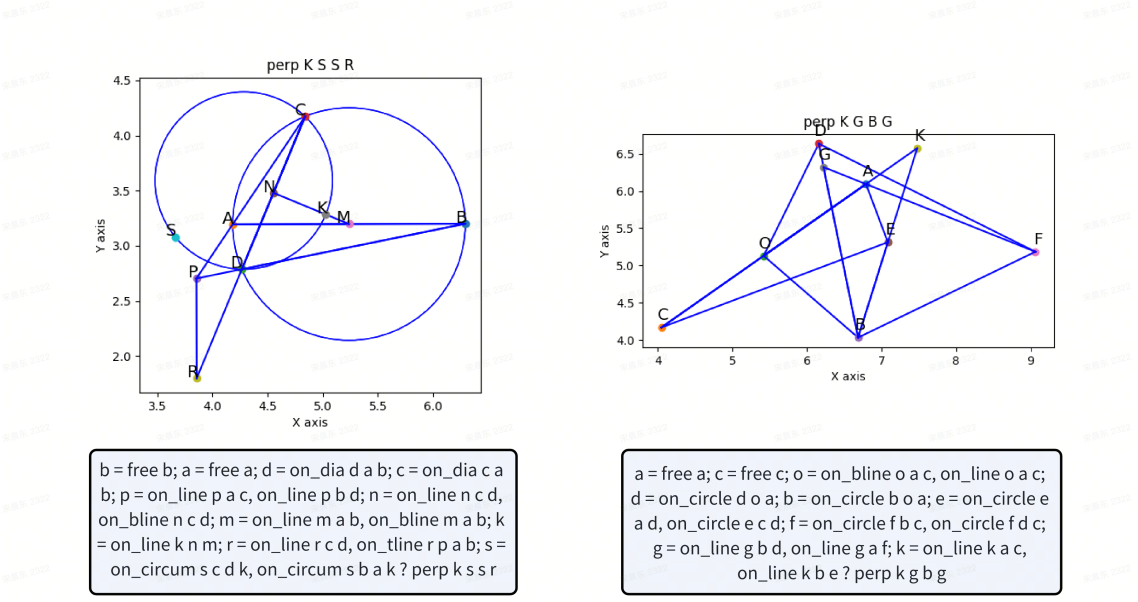


图 2.1　平面几何自动绘图样例

具体地，由于构造性形式化语言的每个步骤都是由已知点构造未知点，且每个构造性命题 P 都给出了某个新点（计为 A）的坐标的一个或两个限制条件，其中，谓词 angle_ bisector, on_ line, on_ pline, on_ tline, on_ bline, on_ aline 为“线性限制条件”，它们规定了描述点在某条直线 $Ax + By + C = 0$ 上，参数 A, B, C 由限制条件的种类和参数点的坐标决定。而 on_ circle, on_ circum, on_ dia, on_ aline2,eqdistance,eqangle3 为“非线性限制条件”，其规定了描述点在 $(x - D)^2 + (y - E)^2 = F^2$ 的圆上, 参数 D, E, F 由限制条件的种类和参数点的坐标决定。如果描述点仅有一个限制条件，则在该直线或圆的轨迹上任取一点即可作为描述点坐标，如果有两个，该点坐标可以通过联立直线和圆的交点解出。注意到直线和圆、圆与圆有可能有两个或没有交点。如果没有交点，则此次作图视为失败，进行回溯。如果有两个交点，程序将在每个节点进行二叉树搜索。当所有点都绘制完成时，程序会进行判断终止条件：

- 所作图形没有相同坐标的两个点，
- 所作图形满足题目描述的限制性结论。

当搜索到满足要求的图时，程序将终止，否则回溯至上一次分叉的节点。

如下图所示，在作出三个点 A, B, C 后，程序得到的 D 点是两个圆的交点，于是先尝试了一种可能情况，作出点 D□F，但是该情况绘制得到的图不符合结论 $DF = EF$, 因此程序回溯并找到两圆的另一个交点 D', 并在此基础上作出 F'，得到一个符合条件的图，如下所示：

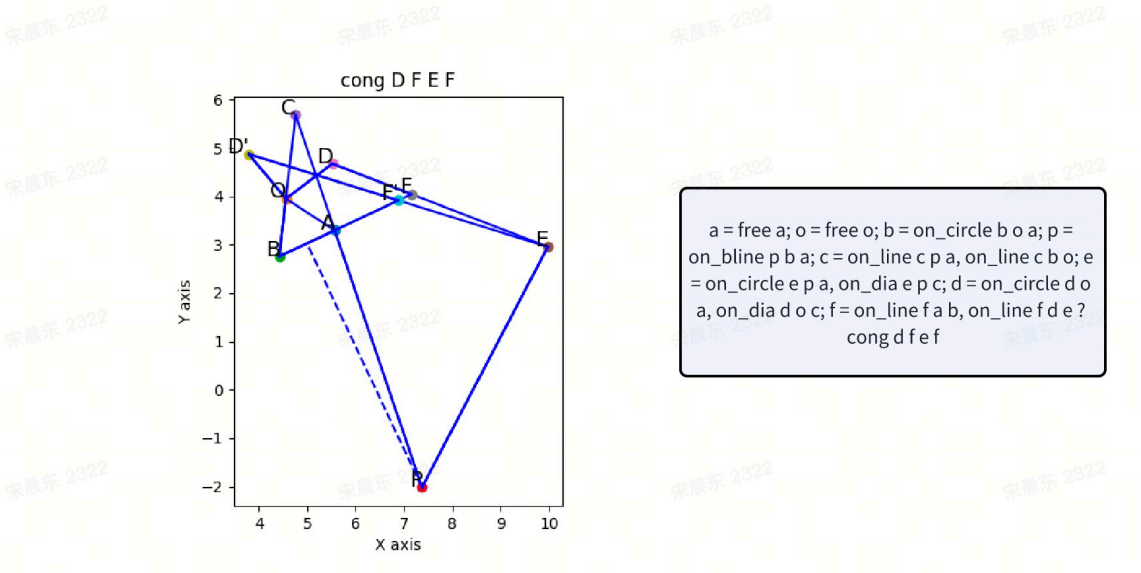


图 2.2　程序搜索绘图示例

我们从合成数据集翻译的到的结果中选取了 200 条长度最长的题目进行绘图测试。其中，123 道题目 (61.5%) 成功画出了几何图形。在绘出的图形中，经人工检查，有 72% 的图形能够较为清晰地反映图形的结构，有 28% 的图形里点和直线聚集在一起，虽然在坐标上符合题目要求约束，但是没有清晰的几何结构，如下图所示：

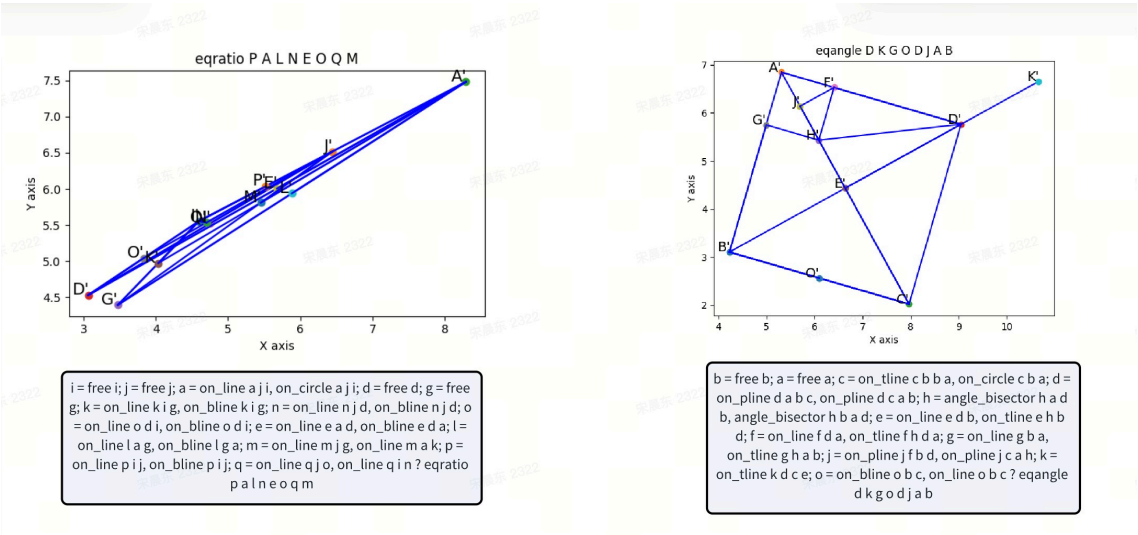


图 2.3　模糊的几何图形和清晰的几何图形比较

此外，我们也比较了自然语言绘图和人类绘图习惯的差别。下面是人类和程序对于 IMO2004 年第一题几何题的绘图[14]，由于形式化语言拆散了原本几何叙述圆与直线的叙述，程序作出的图在点的相对位置关系能和原题保持类似，但是由于形式化语言对于直线和圆的叙述规则和自然语言有所差异，因此连出的圆与直线和按自然语言要求绘出的图形有所差别，如下图所示：

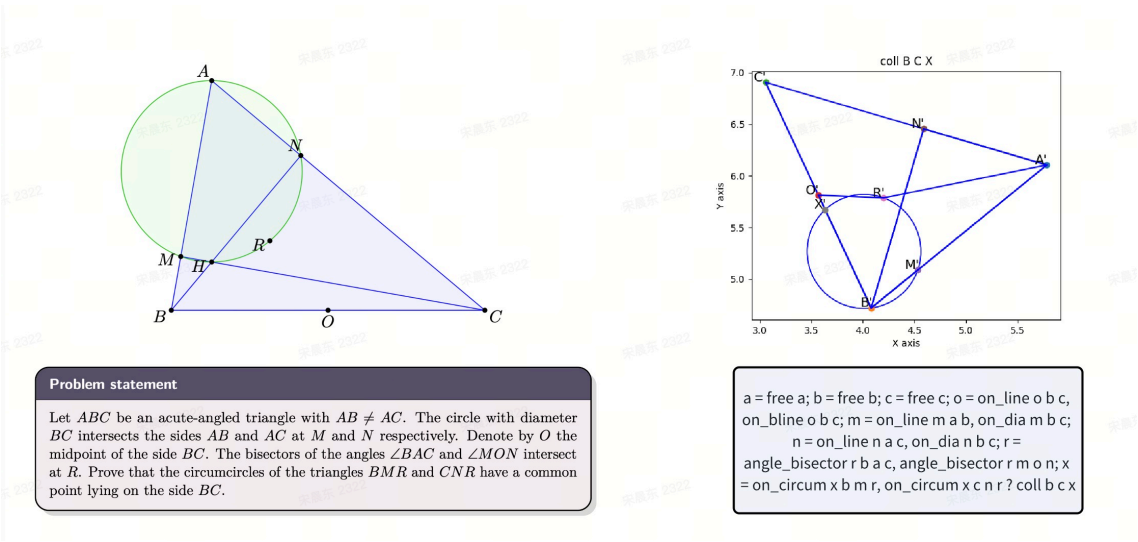


图 2.4　人类绘图和程序绘图比较[14]

第 3 章　平面几何自然语言到构造性语言的转化

自动形式化 (Autoformalization)[2] 是指利用程序或大语言模型等工具自动将自然语言数学问题转化成形式化定理的过程。传统的自动形式化策略往往基于模式匹配或语义分析，对文本的结构性要求较高。而大语言模型的出现为自动形式化提供了更普适、通用的方法。在本节我们将介绍平面几何习题基于模式匹配的形式化方法和基于大语言模型型的形式化方法。

3.1　基于模式匹配的形式化

在此节中我们讨论一种基于模式匹配的形式化策略。首先，通过对自然语言题干进行句法分析，识别出关键词和几何要素（如点、线段、相等关系等）。随后，按照“分别”“和”等连接词对语句进行第二次拆分，并借助句式匹配模板，逐一从句中提取出符合预设模式的表达，得到每句话的限制性语言，最终算法 1.1 将其翻译成连贯的构造性语言，具体流程如下所示：

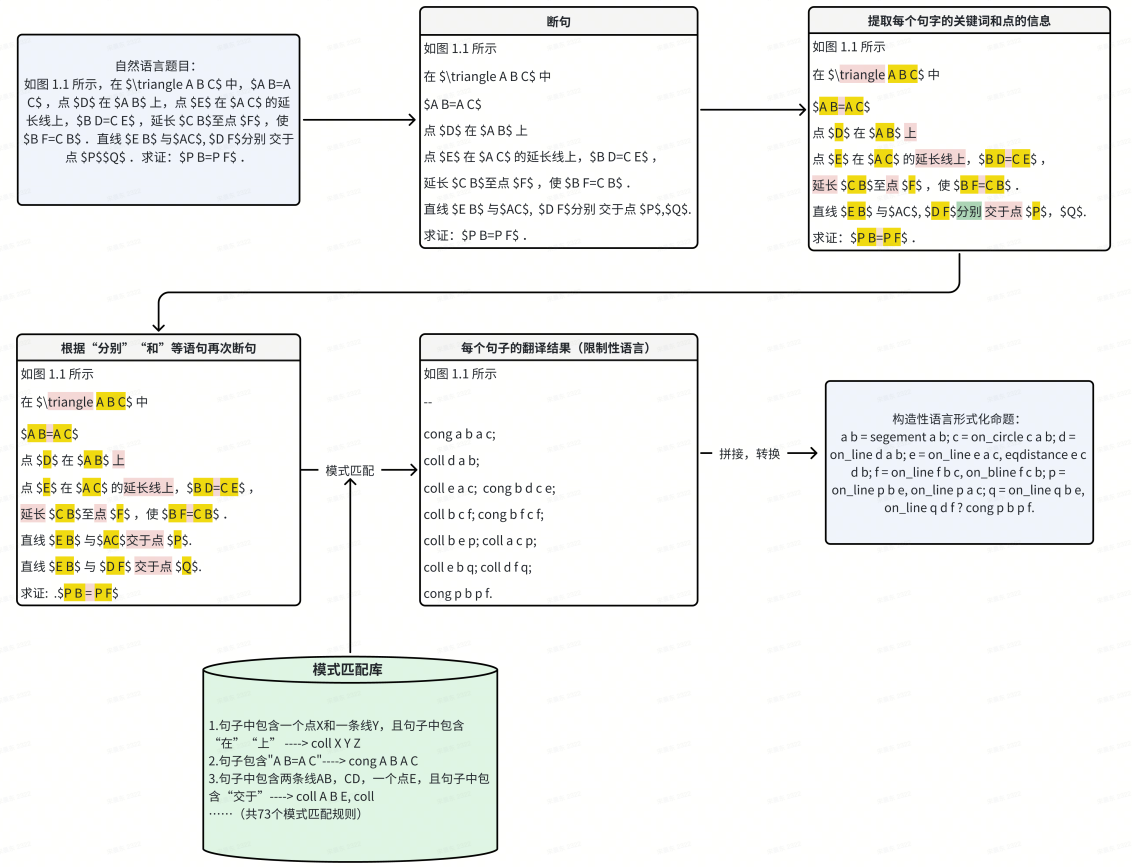


图 3.1　基于模式匹配的形式化

在这个过程中面临两个困难：首先是自然语言叙述的灵活性，如承前省略会让该系统的模式匹配缺乏关键信息。其次是自然语言的叙述中一句之中可能包含的复杂语法：如“三角形 ABC 的外接圆圆心 O 在直线 AB 上”，在新定义一个几何对象的同时也蕴含了该几何对象满足的关系。因此，基于模式匹配的形式化在实践中表现不佳。

3.2　大语言模型自动形式化

3.2.1　形式化翻译的正确性验证

大语言模型自动形式化翻译判断的正确性一直是个棘手的问题，对于平面几何题，我们搭建了一套准确判断平面几何形式化正确性的管线：首先利用大语言模型翻译自然语言定理，筛选出符合 AlphaGeometry 语法规范的题目，接着将语法正确的题目作为输入，对这些题目进行数值检验，最后再将通过数值检验的定理和对应的自然语言命题作为输入，让第二个大语言模型进行“语义检验”，即判断定理是否和原命题匹配，如果匹配则视为翻译成功，过不了语法检验、数值检验和

语义检验的题目均视为翻译失败, 具体过程如下图所示：

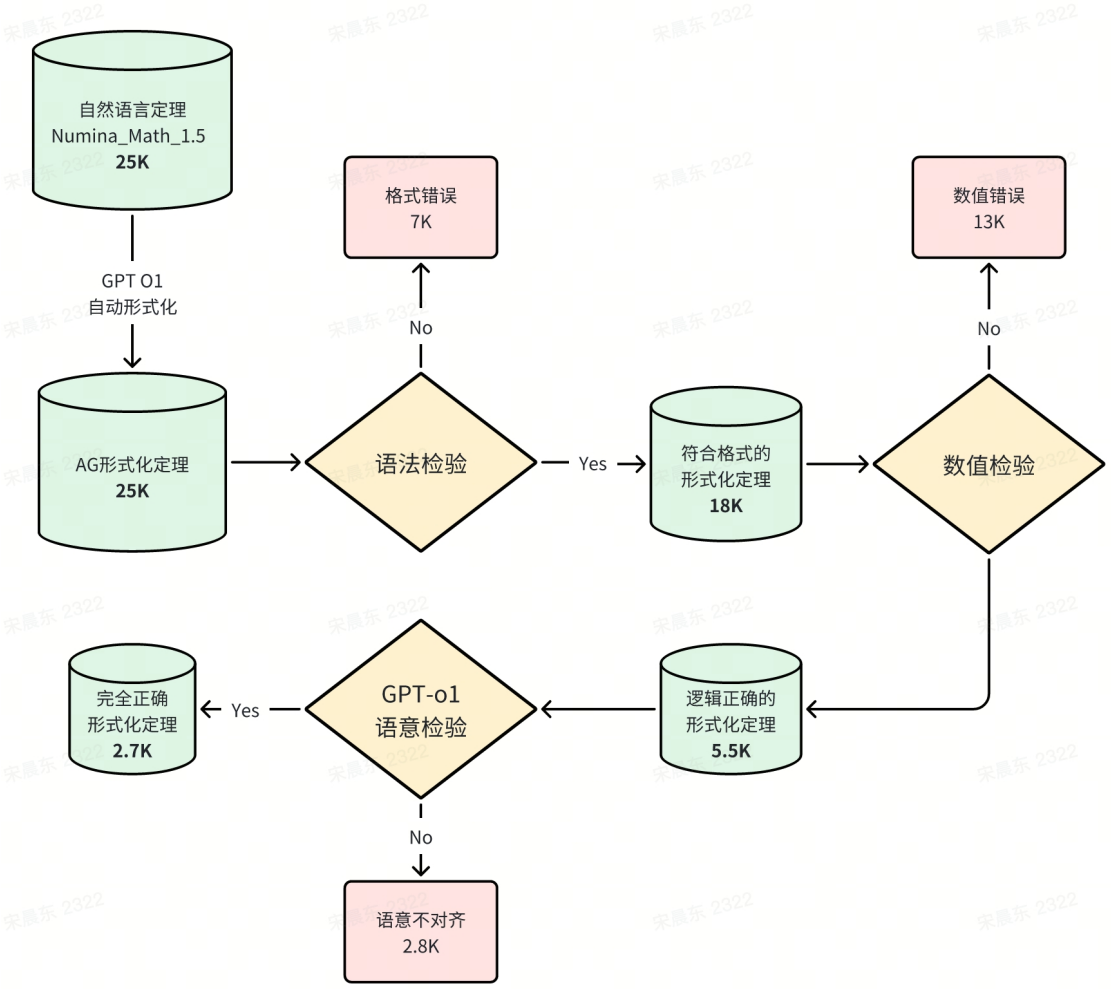


图 3.2　平面几何自动形式化验证管线

在这套管线中，我们搭建了 3 种不同类型的检验器，分别是：语法检验器、数值检验器和语义检验器。语法检验器能够筛选出符合 AlphaGeometry 语法规范的题目，数值检验能够保证命题是一个正确的几何题，而语义检验则保证了翻译出来的命题和原命题内容对齐。

其中，数值检验基于自动构图系统，其流程和自动构图系统原理类似：假设得到的构造性形式化命题是 $P \Rightarrow Q$, 我们在 $\mathbb{R}^2$ 上先随机选取自由点, 由于 P 是构造性的，P 的每一条命题都仅包含了一个未知点和该点所满足的限制条件，每个限制条件经过转化，总能得到该点在直线上或者在圆上。如果该点有两个约束条件，可以通过解二次方程来得到该点精确的坐标（如果是直线和圆、圆或圆的交点，则随机选取该点两个可能坐标的一个），如果该点仅有一个限制条件，则在直线或者圆上随机选取一个点。对构造性语言中的每一个点依次进行如上操作，如果在所

有点画出后满足 Q 的约束条件，则该命题能够“通过数值检验”。由于初始点选取和交点选取具有随机性，我们将该过程重复 100 次，其中有一次能够通过数值检验，则视为通过数值检验。

语义检验则通过提示词工程调用 GPT-o1 的 API 来实现。将自然语言命题和形式化定理的配对和形式化定理的命名规则作为输入给大语言模型，并让其判断两者在语义上是否符合。并返回“Yes”或者“No”的结果。以下是语义检验的提示词：

```
Please verify whether the following natural language statement
   of plane geometry problem matches with the formal statement.
    Some qualitative could be neglected, like the acute-
   triangle can be translated as triangle.  Please output "Yes"
    for matching and "No" for not matching.
--Examples:
<Input>
<problem1> : In a convex quadrilateral $ABCD$, the diagonal
   $BD$ bisects neither the angle $ABC$ nor the angle $CDA$.
   The point $P$ lies inside $ABCD$ and satisfies $\\angle PBC
   = \\angle DBA$ and $\\angle PDC = \\angle BDA$. Prove that
   if $ABCD$ is a cyclic quadrilateral, then $AP = CP$.
<formal_statement1> : a b c = triangle a b c; o = circle o a b
   c; d = on_circle d o a; p = on_aline p b c a b d, on_aline p
    d c a d b ? cong a p c p
<Output>
Yes
```

3.2.2 判断大模型翻译能力的基准测试

在有了这套管线之后，为了比较不同语言模型、不同翻译策略的自动形式化的效果，我们搭建了由 100 道历年 IMO 真题和预选题构成的基准测试 IMO_100。我们首先批量下载了 1987 年到 2024 年的 IMO 几何真题和预选题。接着人工滤去涉及组合几何、几何不等式等无法被形式化的题目，并将这些题目利用提示词让大语言模型自动形式化，将大语言模型返回的结果进行人工修正，使其能通过数值检验，并确保和原题目语义对齐。该基准测试能够公平比较模型在自动形式化上的能力。

3.3 提示词工程

使用 AI 自动形式化题目有两种策略：一种是使用提示词工程（prompt engineering) 和少样本提示，通过调用通用大模型的 API 来辅助定理的形式化翻译。提示词工程旨在撰写全面、完整的指令和少样本提示来指导 AI 完成特定任务。在平面几何翻译任务中，我们使用了以下多部分全面而系统的提示词：

3.3.1 先行提示词

此部分提示词目的在于明确 AI 的任务流程和所需的基础知识。

```
- Role: Expert in Formalizing Plane Geometry Theorems

- Background: The user needs to translate plane geometry
   theorems described in natural language into formal language
   to achieve more precise mathematical expression and logical
   reasoning. The user has already provided detailed
   explanations of the formal language and translation examples
   , but requires assistance to complete the specific
   translation tasks.

- Profile: You are an expert proficient in plane geometry and
   formal language, capable of accurately understanding
   geometric concepts in natural language and converting them
   into logical formal expressions. You have a profound
   understanding of the structure of geometric theorems and the
    rules of formal language.

- Skills: Proficient in translating natural language
   descriptions of plane geometry theorems into formal language
   , accurately identifying geometric elements and logical
   relationships, and converting them into formal expressions.
   Also, you possess rigorous logical thinking and a high
   attention to detail to ensure the accuracy and ambiguity-
   free nature of the translation.

- Goals: Based on the natural language plane geometry theorems
   provided by the user, accurately translate them into formal
   language, ensuring that the translation results comply with
   the rules and logical structure of formal language, avoiding
    translation errors, and providing clear explanations of the
    translation process.

- Constrains: The translation process must strictly follow the
   rules of formal language, including the correct use of
   predicates, accurate number of parameters, and ensuring that
    each point's definition does not exceed two constraints.
   The translation result must not contain undefined predicates
    or symbols, and the predicates following the question mark
   must be one of the predefined conclusion forms.

- OutputFormat: The translation result should be output in the
   specified format, including the definition of each point and
    the expression of conclusions, ensuring that the format is
   standardized and clear.

- Workflow:

  1. Carefully read the natural language description of the
     plane geometry theorem to understand the geometric
```

```
   elements and logical relationships in the problem.
2. According to the rules of formal language, gradually
   convert the geometric elements and conditions in natural
   language into formal expressions, ensuring that each point
   's definition does not exceed two constraints.
3. Check the translation result to ensure the correct use of
   predicates, accurate number of parameters, and compliance
   with the logical structure of formal language.
4. Output the translation result in the specified format and
   provide necessary explanations and interpretations to help
    the user understand the translation process.
```

3.3.2 翻译规则

此部分包含了 AlphaGeometry 两种形式化语言的规则和具体用法。

```
--Translation Rules:
- x = angle_bisector x a b c: Construct x on the angle bisector
    of angle abc. (1)
- x = angle_mirror x a b c: Construct x such that angle abc =
   angle cbx. (1)
- x = circle x a b c: Construct the circumcenter x of triangle
   abc. (1)
- x = on_circum x a b c: Construct point x such that x lies on
   the circumcircle of triangle abc. (1)
- x = eqdistance x a b c: Construct point x such that xa = bc.
   (1)
- x = foot x a b c: Construct point x such that x is the foot
   of the perpendicular from a to bc. (2)
- x = free x: x is a free point (no other constraints). (0)
- x = incenter x a b c: Construct x such that x is the incenter
    of triangle abc. (2)
- t1 t2 t3 i = incenter2 t1 t2 t3 i a b c : Construct i such
   that i is the incenter of triangle abc. And the incircle of
   triangle $ABC$ touches the sides bc, ca, and ab at t1, t2,
   and t3, respectively.
- x = excenter x a b c: Construct x such that x is the excenter
    of triangle abc opposite to a. (2)
- x = midpoint x a b: Construct point x such that x is the
   midpoint of a and b. (2)
- x = mirror x a b: Construct point x such that x is the
   reflection of a over b. (2)
- x = on_aline x a b c d e: Construct point x such that anglr
   xab = angle cde. (1)
- x = eqangle3 x a b c d e: Construct point x such that angle
   axb = angle cde. (1)
- x = on_bline x a b: Construct point x such that xa = xb. (1)
- x = on_circle x o a: Construct point x such that x lies on
   the circle with center o and radius oa(xo=oa). (1)
- x = on_line x a b: Construct point x such that x lies on line
    ab. (1)
- x = on_dia x a b: Construct point x such that angle axb = 90°
   . (1)
```

```
- x = on_pline x a b c: Construct point x such that x lies on
   the line through a and parallel to bc. (1)
- x = on_tline x a b c: Construct point x such that x lies on
   the line through a and perpendicular to bc. (1)
- x = orthocenter x a b c: Construct point x such that x is the
    orthocenter of triangle abc. (2)
- x = reflect x a b c: Construct point x such that x is the
   reflection of a over bc. (2)
- a b = segment a b: Construct line segment ab. (0)
- a b c = triangle a b c: Construct points a, b, c such that
   abc is an ordinary triangle. (0)
- x y = cc_tangent0 x y o a w b: Construct points x, y such
   that x, y are the common tangents of the circle with center
   o and radius oa and the circle with center w and radius wb.
   (2)

-- Conclusion Forms:
For the problem to be proved, use "?" to connect, and the
   conclusion to be proved follows "?". The conclusion is
   described in the following forms:
-? coll a b c indicates that a, b, c need to be proved
   collinear
-? cong a b c d indicates that the lengths of line segments ab
   and cd need to be proved equal
-? para a b c d indicates that lines ab and cd need to be
   proved parallel
-? perp a b c d indicates that line segment ab needs to be
   proved perpendicular to cd
-? cyclic a b c d indicates that points a, b, c, d need to be
   proved concyclic
-? eqangle a b c d e f g h indicates that the directed angles
   of line segments ab, cd and ef, gh need to be proved equal
-? eqangle a b b c e f f g indicates that angle abc = angle efg
    to be proved equal.
-? eqratio a b c d e f g h indicates that the ratios ab/cd and
   ef/gh need to be proved equal
```

3.3.3 少样本提示

该部分使用自然语言——形式化语言的一一对应翻译例子来具体展示翻译技巧，同时明确 AI 的输出格式。

```
-- Examples:
Below are some pairings of formal problem statements and
   problems: the problem is in English, and the formal theorem
   should be separated by semicolons, with "?" used to connect
   the problem to be proved.
<problem1> : In a convex quadrilateral $ABCD$, the diagonal
   $BD$ bisects neither the angle $ABC$ nor the angle $CDA$.
   The point $P$ lies inside $ABCD$ and satisfies $\\angle PBC
   = \\angle DBA$ and $\\angle PDC = \\angle BDA$. Prove that
   if $ABCD$ is a cyclic quadrilateral, then $AP = CP$.
```

```
<formal_statement1> : a b c = triangle a b c; o = circle o a b
   c; d = on_circle d o a; p = on_aline p b c a b d, on_aline p
    d c a d b ? cong a p c p

<problem2>: Let $D$ be an interior point of the acute triangle
   $ABC$ with $AB > AC$ so that $\\angle DAB= \\angle CAD$. The
    point $E$ on the segment $AC$ satisfies $\\angle ADE= \\
   angle BCD$, the point $F$ on the segment $AB$ satisfies $\\
   angle FDA= \\angle DBC$, and the point $X$ on the line $AC$
   satisfies $CX=BX$. Let $O_1$ and $O_2$ be the circumcentres
   of the triangles $ADC$ and $EXD$ respectively. Prove that
   the lines $BC$, $EF$, and $O_1 O_2$ are concurrent.",
<ormal_statement2>: a b c = triangle a b c; d = angle_bisector
   d b a c; e = on_aline e d a d c b, on_line a c; f = on_aline
    f d a d b c, on_line f a b; x = on_bline x b c, on_line x a
    c; o1 = circle o1 a d c; o2 = circle o2 e x d; y = on_line
   y e f, on_line y b c ? coll o1 o2 y

<problem3>: $BC$ is a diameter of a circle center $O$. $A$ is
   any point on the circle with $\angle AOC \not\le 60^\circ$.
   $EF$ is the chord which is the perpendicular bisector of
   $AO$. $D$ is the midpoint of the minor arc $AB$. The line
   through $O$ parallel to $AD$ meets $AC$ at $J$. Show that
   $J$ is the incenter of triangle $CEF$.
<formal statement3> : b c = segment b c; o = midpoint o b c; a
   = on_circle a o b; d = on_circle d o b, on_bline d a b; e =
   on_bline e o a, on_circle e o b; f = on_bline f o a,
   on_circle f o b; j = on_pline j o a d, on_line j a c ?
   eqangle e c e j e j e f
```

3.3.4 翻译注意事项

下面一部分提示词基于在 AI 翻译过程中常见的格式错误，如参数数量不正确，喜欢新造谓词等问题给出的进一步指导，旨在让翻译结果更加规范化。

```
--Tips:
Accuracy is crucial in translation, and you need to ensure that
    the translation expresses the meaning of the original text
   and avoid ambiguity. In particular, you should pay attention
    to the following in your translation:
-1. You need to check the number of parameters after the
   predicate, such as a = circle a b c must have three
   parameters
Please do not use irrelevant symbols such as parentheses in the
    translation results. Also, do not use predicates that have
   not been given to you.
-2. n your translated results, each point should have at most
   two conditions to restrict it. In the Translation Rules of
   the formal language, each item of translation has a
   parenthesis. The parenthesis indicates the number of
   restrictions brought by this predicate. Each point's
   translation should have at most 2 restrictions.
```

```
-3. The predicate after the question mark must be one of coll,
   cong, para, perp, cyclic, eqangle, eqratio, and these
   predicates should not appear before the question mark.
-4. The format for translating the theorem: "<point> = <
   predicate> <parameters>;" or "<point> = <predicate> = <
   predicate1> <parameters1>, <predicate2> <parameters2>", and
   for the conclusion, the format you translate is "? <
   predicate> <parameters>"
For each translation, such as "e = on_line e a c, on_circle e a
    d;", e is the point being translated, "on_line" and "
   on_circle" are the predicates being translated, and the
   predicates are followed by the parameters of that sentence.
   In the translation, you need to ensure that the first
   parameter is exactly the point you need to translate.
   Besides, the point you need to translate cann't appear at
   the second, third or fourth parameter.
5. For every point, you should check EVERY condition that the
   point satisfies. Don't miss any condtions.
```

3.3.5 给出翻译任务

以下提示词旨在给出 AI 具体的翻译任务，其中自然语言的部分会遍历数据库中的所有自然语言数据。

```
-- Task:
    Please translate the following problem according to
       workflow. Carefully consider the --Tips and follow the
       output format. Think carefully and step by step. Your
       formal statement output should be in TWO line(one for
       problem, one for formal statement) without any comments.
    <problem_11>
    In triangle $ABC$, point $A_1$ lies on side $BC$ and point
       $B_1$ lies on side $AC$. Let $P$ and $Q$ be points on
       segments $AA_1$ and $BB_1$, respectively, such that $PQ$
        is parallel to $AB$. Let $P_1$ be a point on line
       $PB_1$, such that $B_1$ lies strictly between $P$ and
       $P_1$, and $\\angle PP_1C=\\angle BAC$. Similarly, let
       $Q_1$ be the point on line $QA_1$, such that $A_1$ lies
       strictly between $Q$ and $Q_1$, and $\\angle CQ_1Q=\\
       angle CBA$. Prove that points $P,Q,P_1$, and $Q_1$ are
       concyclic.
```

3.3.6 不同提示词翻译比较

首先我们比较翻译样例 <example> 的数量对于翻译效果的影响。我们测试了 0 个，1 个，3 个和 10 个样例提示条件下，模型翻译的准确率，结果如下表：

表 3.1　翻译样例数对大模型翻译的正确率的影响

翻译样例数（字符数）	翻译准确率（pass@1）
0 个样例（7K）	16%
1 个样例 (8K)	21%
3 个样例 (10K)	25%
10 个样例 (16K)	26%

实验结果表明，在样例数比较少时，增加翻译样本数能够提高大模型自动形式化的能力，但是过多的样例并不能够有效地提高大模型翻译的能力。

另外，我们测试了是否将 AlphaGeometry 更细致的语法规则（Tips）的部分加入提示词的对比实验, 结果如下表：

表 3.2　是否加入 tips 对大模型翻译的正确率的影响

翻译策略（字符数）	翻译准确率（pass@1）
0 个样例（6K），没有 tips	8 %
0 个样例（7K），有 tips	16%
3 个样例 (9K)，没有 tips	24 %
3 个样例 (10K)，有 tips	25%

上述实验表明，在零样本提示时，具体的指令规则对于模型理解翻译规则和要求有很大影响。但是有少样本提示时，模型可以通过学习例子来理解翻译细化的理解要求，因此指令规则对于模型翻译正确率的提升并不显著。以下是一个模型翻译的例子, 不同的翻译策略会给模型的答案带来很大影响。其中 gpt_result_1 和 gpt_result_2 通过了数值检验，翻译成功，而其中 gpt_result_3 和 gpt_result_4 的翻译语法均有问题，故没有例子和细化翻译规则的提示词对于语法的指令遵从性远不如给了翻译例子的提示词，具体例子如下所示：

Listing 3.1　不同提示词翻译结果比较

```
"problem": "$BC$ is a diameter of a circle center $O$. $A$ is
   any point on the circle with $\\angle AOC \\not\\le 60^\\
   circ$. $EF$ is the chord which is the perpendicular bisector
    of $AO$. $D$ is the midpoint of the minor arc $AB$. The
   line through $O$ parallel to $AD$ meets $AC$ at $J$. Show
   that $J$ is the incenter of triangle $CEF$."
"gpt_result_1（3个样例，有tips）": "b c = segment b c; o =
   midpoint o b c; a = on_circle a o b; d = on_bline d a b,
```

```
   on_circle d o b; e = on_bline e a o, on_circle e o b; f =
   on_bline f a o, on_circle f o b; j = on_pline j o a d,
   on_line j a c ? eqangle e c e j e j e f",
"gpt_result_1（3个样例，无tips）": "b c = segment b c; o =
   midpoint o b c; a = on_circle a o b; d = on_circle d o b,
   on_bline d a b; e = on_bline e o a, on_circle e o b; f =
   on_bline f o a, on_circle f o b; j = on_pline j o a d,
   on_line j a c ? eqangle e c e j e j e f",
"gpt_result_3（0个样例，有tips）": "o a = segment o a; b =
   on_circle b o a; c = mirror c b o, on_circle c o a; m =
   midpoint m a o; e = on_circle e o a, on_tline e m a o; f =
   mirror f e m, on_circle f o a; d = on_circle d o a, on_bline
    d a b; j = on_pline j o a d, on_line j a c ? eqangle e c c
   j j c c f"
"gpt_result_4（0个样例，无tips）": "o a = segment o a; b =
   on_circle b o a; c = on_circle c o a; m = midpoint m a o; e
   = on_circle e o a, on_tline e m a o; f = mirror f e m a,
   on_circle f o a; d = on_circle d o a, on_bline d a b; j =
   on_pline j o a d, on_line j a c ? eqangle e c j c c f"
},
```

3.3.7 模型翻译结果与分析

我们使用包含大量非形式化数学题的 Numina_ Math_1.5[15]作为原始数据集，筛选出其中带标签“Geometry”和“proof”的证明题 25631 道。通过并行调用 GPT-o1 的 API，使用 3 个样例 +tips 的提示词进行形式化翻译。在 25631 道题中，共 18323 道题目通过了翻译的语法检验，5469 道题目通过了翻译语法和数值检验，2720 道题目通过了翻译的语义检验，最终通过管线翻译成功，我们将翻译成功的作为一个经过验证的几何形式化数据集 Numina_Geometry。

以下是在本次翻译实践中的案例分析：

Listing 3.2 成功案例

```
"problem": Triangle $A B C$ is such that $A B<A C$. The
   perpendicular bisector of side $B C$ intersects lines $A B$
   and $A C$ at points $P$ and $Q$, respectively. Let $H$ be
   the orthocentre of triangle $A B C$, and let $M$ and $N$ be
   the midpoints of segments $B C$ and $P Q$, respectively.
   Prove that lines $H M$ and $A N$ meet on the circumcircle of
    $A B C$.
"formal_statement": a b c = triangle a b c; p = on_line p a b,
   on_bline p b c; q = on_line q a c, on_bline q b c; h =
   orthocenter h a b c; m = midpoint m b c; n = midpoint n p q;
    x = on_line x h m, on_line x a n ? cyclic a b c x,
"trans_state": Successfully translated!
```

在上面的例子中，大模型对变量命名与关系指派均十分清晰，同时在翻译中创造了必要的 HM 和 AN 的交点 X，通过引入中间点过渡完成翻译。这种依赖语义信

息的翻译对于手动形式化几乎是不可能的。

Listing 3.3　翻译失败案例：数值错误

```
"problem": Let $A B C$ be an acute triangle such that $A H=H D$
   , where $H$ is the orthocenter of $A B C$ and $D \\in B C$
   is the foot of the altitude from the vertex $A$. Let $\\ell$
    denote the line through $H$ which is tangent to the
   circumcircle of the triangle $B H C$. Let $S$ and $T$ be the
    intersection points of $\\ell$ with $A B$ and $A C$,
   respectively. Denote the midpoints of $B H$ and $C H$ by $M$
    and $N$, respectively. Prove that the lines $S M$ and $T N$
    are parallel."
"formal_statement": "a b c = triangle a b c; h = orthocenter h
   a b c; d = foot d a b c; m = midpoint m b h; n = midpoint n
   c h; s = on_line s a b; t = on_line t a c ? para s m t n",
"trans_state": "Numerically Wrong!"
```

上面例子翻译错误的原因是题目本身很难用构造性语言形式化。按照题目的要求，如果任取一个三角形 ABC，则 H 同时需要满足 H 是三角形 ABC 的垂心且 $AH = HD$, 将有 3 个限制条件，因此一开始的三角形 ABC 不能够直接取，而应该满足某种性质。因此，这种描述无法一对一地用构造性形式化语言表述。在大语言模型翻译的过程中，其漏去了 $AH = HD$ 的条件，因此被判断为“数值错误”。

Listing 3.4　翻译失败案例：语法错误

```
"problem": Let $A B C$ be an acute-angled triangle with
   circumscribed circle $k$ and centre of the circumscribed
   circle $O$. A line through $O$ intersects the sides $A B$
   and $A C$ at $D$ and $E$. Denote by $B^{\\prime}$ and $C^{\\
   prime}$ the reflections of $B$ and $C$ over $O$,
   respectively. Prove that the circumscribed circles of $O D C
   ^{\\prime}$ and $O E B^{\\prime}$ concur on $k$.
"formal_statement": "a b c = triangle a b c; o = circle o a b c
   ; d = on_line d a b, on_line d o e; e = on_line e a c,
   on_line e o d; b' = mirror b' b o; c' = mirror c' c o; x =
   on_circum x o d c', on_circum x o e b' ? cyclic a b c x",
"trans_state": "Syntactically wrong at 'e'"
```

上面的例子里面，对于条件 DOE 三点共线，模型重复了两次。但是第一次出现时 E 点仍然是未知点，因此 d = on_line d o e 并不符合语法规范。模型能够理解题目的语义，但是并没有完全掌握翻译的语法。

Listing 3.5　翻译失败案例：语义错误

```
"problem": Prove that if, during a parallel projection of one
   plane onto another plane, the image of a certain square is a
    square, then the image of any figure is a figure congruent
   to it.
"formal_statement": "a = free a; b = free b ? cong a b a b",
```

```
"trans_state": "Semantically Wrong!"
```

上面的例子中，题目本身脱离了平面几何的范畴，因此无法用自然语言形式化。而模型输出了一个数值上一定正确的结果 AB=AB, 这种结果过不了第二个语言模型的语义检验，并不能认为翻译成功。

3.4 平面几何形式化数据集比较分析

为了更好地定量衡量得到数据集 Numina_Geometry 的质量，我们将得到的数据集和不同的几何数据集进行对比。其中，IMO_30 和 jgex_ag_231 是 AlphaGeometry 自带评测的基准测试集，结果如下表所示：

表 3.3 不同几何题的数据集比较

数据集	题目数量	平均题目长度	平均题目前提个数	DD+AR 解决成功率
人类转化题				
Numina_Geometry	2720	195.45	9.11	53.1%
IMO_30	30	268.27	12.50	46.7%
jgex_ag_231	231	156.21	7.02	85.7%
机器合成题				
sr_geometry	15584	184.36	9.57	92.11%

从表中可以看到，对于人类转化题，题目中题目条件和前提条件个数的增加会显著增大题目的难度——对于人类和推理引擎来说都是如此。从表中可以看出,DD+AR 解决 Numina_Geometry 中几何题的能力仅比解决 IMO_30 高了 8 个百分点。这表明 AI 和人类在解决几何题对于难度的标准判断有很大差别——面对几何题时，AI 更多的是基于语法的逻辑推理，而人类进行的是基于图形的直觉推理，直觉上显然的结论通过逻辑推导未必简单。因此尽管对于人类而言，Numina_Geometry 的几何题要远远简单于 IMO_30，但是题目对于 AI 而言仍然有不亚于 IMO_30 的挑战性。

但是对于机器合成的题目，虽然其可能有着大量的前提条件，但是可能有很多条件都是无关冗余的或在几何上处理相对简单，其复杂仅仅是通过几个简单几何结构的线性叠加，因此并没有给推理引擎造成很大的挑战。

下图是人工合成题中题目长度最长的题通过自动绘图系统构造出的两个图，可以看到，人工合成题在几何图形上易出现等腰直角三角形、正方形等较容易被

解析的结构，这些结构并不会提升题目本身的难度，也不具有几何题应该具有的训练价值，而仅仅是使题目从叙述上变得更长更复杂。因此，如果需要合成对人类有训练意义的题目，需要更改题目前提条件的配比。

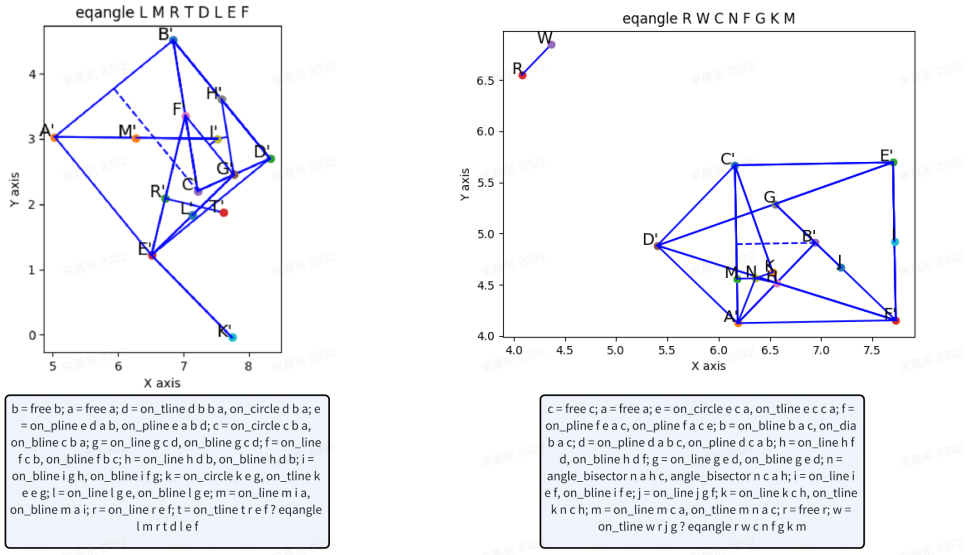


图 3.3　人工合成题示例

3.5　Numina_Geometry 数据集在 AlphaGeometry 上的测试

我们将 Numina_Geometry 的 2720 道平面几何题作为 AlphaGeometry 的输入，首先让 AlphaGeometry 进行一次 DD+AR 的测试，结果 1444 道题目经过一步 DDAR 就获得解决。剩余 1276 题目则需要借助大模型构造辅助点和树搜索来解决。我们设定模型的搜索参数 `BATCH_SIZE=8, BEAM_SIZE=32, DEPTH=8`，并且在 16 个核的 CPU 上并行，设定每道题的最长的搜索时间为 5 分钟，结果表明 AlphaGeometry 解出来的题目为 396 题，测试的成功率为 31.1%。加上通过 DDAR 的题目，AlphaGeometry 在该数据上的总体成功率仅为 67.6%。考虑到数据集 Numina_Geometry 的几何题难度远不如 IMO，这和在 AlphaGeometry 论文[8]中提到的解决 83.3% 的 IMO 几何问题有较大差距。造成差距的主要因素在于搜索搜索资源的差距。首先，AlphaGeometry 的计算消耗集中在于 DDAR 搜索的 CPU，AlphaGeometry 的测试条件是 10000 核 CPU 同时测试 15 题，测试时间为 1.5 小时，因此平均每道题拥有的计算资源是本实验的 750 倍。可以看到，AlphaGeometry 对于计算资源的规模效应非常明显，即增大搜索辅助点的层数和个数能够明显地提高模型的性能。

但是，在另一方面，我们也注意到 AlphaGeometry 构造辅助点的不稳定性。例如对于 IMO 上较简单的 2008-P1，Alphageomtry 需要使用 246 个核才能在 1.5 小时

内解出，相当于尝试了近 2000 个辅助点，故不能下断言“AlphaGeometry 通过推理或者几何能力“找到”了辅助点，而更像是通过增加算力枚举出了所有的辅助点并进行验证。这也印证了，AlphaGeometry 获得 IMO 银牌水平并不是 AI 解平面几何的终点，大语言模型的图形推理能力仍然有进一步提升的可能。

第 4 章　结论与展望

本研究围绕平面几何习题的自动形式化问题，提出了限制性形式化语言到构造性形式化语言的系统转化算法，搭建了基于构造性语言的自动绘图系统，并设计了自然语言到形式化语言的三阶段验证管线，构建了形式化几何高质量数据集。

在此基础上，本研究还得出以下观察：

- 本文提出的删点算法，在理论上保证了限制性语言到构造性语言转化的充分性和有效性，为统一描述几何命题形式化语言作出了贡献。然而，实际应用中仍存在个别无法通过构造步骤表达的复杂限制性命题，提示我们未来需要引入更丰富的构造动作或适当扩展形式化语言表达能力。
- 大语言模型虽然展示了较强的语言理解与形式化翻译能力，但在面对多层嵌套、隐式假设和复杂构造要求的题目时，翻译准确率仍有明显下降。尤其是在涉及辅助点隐式定义、复杂约束条件推导时，模型容易遗漏关键信息。这表明，单纯基于文本提示的大模型形式化仍不足以全面覆盖平面几何问题，可能需要针对此弱点专门构造数据训练。
- 通过 AlphaGeometry 对新数据集的实验表明，在有限算力资源下，推理成功率显著下降，尤其在需要搜索复杂辅助点构造时。这一现象提示，现有推理系统在“智慧推理”（主动构造辅助结构）上仍然依赖穷举和资源堆叠，缺乏对几何直觉与构造策略的有效建模，未来需要从辅助点智能生成、推理深度优化等方面进一步突破。

展望未来，平面几何自动形式化研究可以沿以下几个方向继续深入：

- 引入多模态推理：结合图像信息（如几何图形草图）与文本信息，辅助自然语言理解和辅助点定位，从而提高复杂几何题目的自动形式化与推理成功率。
- 优化辅助点生成与搜索策略：探索启发式或机器学习方法引导辅助点搜索，替代当前的暴力枚举，从而在有限资源下解决更复杂的几何推理问题。
- 开发更丰富、更具表现力的几何形式语言，支持曲线、面积、体积等更复杂概念的表达与推导，为进一步推广到更广泛的数学领域（如立体几何、解析几何）打下基础。

参考文献

[1] Yang K, Poesia G, He J, et al. Formal mathematical reasoning: A new frontier in ai[A]. 2024.

[2] Wu Y, Jiang A Q, Li W, et al. Autoformalization with large language models[J]. Advances in Neural Information Processing Systems, 2022, 35: 32353-32368.

[3] Murphy L, Yang K, Sun J, et al. Autoformalizing euclidean geometry[A]. 2024.

[4] Avigad J, Dean E, Mumma J. A formal system for euclid’s elements[J]. The Review of Symbolic Logic, 2009, 2(4): 700-768.

[5] Chou S C, Gao X S, Zhang J Z. A deductive database approach to automated geometry theorem proving and discovering[J]. Journal of Automated Reasoning, 2000, 25(3): 219-246.

[6] Kutzler B, Stifter S. On the application of buchberger’s algorithm to automated geometry theorem proving[J]. Journal of Symbolic Computation, 1986, 2(4): 389-397.

[7] Wenjun W. Basic principles of mechanical theorem proving in elementary geometries[J]. Selected Works Of Wen-Tsun Wu, 2008: 195.

[8] Trinh T H, Wu Y, Le Q V, et al. Solving olympiad geometry without human demonstrations[J]. Nature, 2024, 625(7995): 476-482.

[9] Zhang C, Song J, Li S, et al. Proposing and solving olympiad geometry with guided tree search [A]. 2024.

[10] Sicca V, Xia T, Fédérico M, et al. Newclid: A user-friendly replacement for alphageometry[A]. 2024.

[11] Fu D, Chen Z, Xia R, et al. Trustgeogen: Scalable and formal-verified data engine for trustworthy multi-modal geometric problem solving[A]. 2025.

[12] Chervonyi Y, Trinh T H, Olšák M, et al. Gold-medalist performance in solving olympiad geometry with alphageometry2[A]. 2025.

[13] Krueger R, Han J M, Selsam D. Automatically building diagrams for olympiad geometry problems.[C]//CADE. 2021: 577-588.

[14] Chen E. Imo 2004 solution notes[EB/OL]. 2025. https://web.evanchen.cc/exams/IMO-2004-notes.pdf.

[15] Li J, Beeching E, Tunstall L, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions[J]. Hugging Face repository, 2024, 13: 9.

附录 A　外文资料的书面翻译

Alphageometry2 在解决奥林匹克几何问题的金牌表现[1]

摘要

本研究主要介绍 AlphaGeometry2（AG2），这是在 Trinh 等中提出的 Alpha-Geometry[2]的显著改进版本，其在解决奥林匹克几何问题方面现已超越人类金牌得主的水平。为了实现这一目标，我们首先扩展了原始的 AlphaGeometry 语言，以解决涉及物体运动以及包含角度、比例和距离的线性方程的更难问题。这些改进显著提高了 AlphaGeometry 语言在 2000-2024 年国际数学奥林匹克竞赛（IMO）几何问题上的准确率，从 66% 提升至 88%。AG2 的搜索过程也通过使用 Gemini 架构实现自动形式化，同时结合多个搜索树，引入了一种新颖的知识共享机制。同时，我们对符号引擎和合成数据生成进一步增强，从而显著提高了 AG2 的整体解题率，使其在过去 25 年的所有 IMO 几何问题上的解题率达到了 84%，相比之前的 54% 有了显著提升。AG2 也是在 2024 年 IMO 中达到银牌标准的系统 Alphaproof 的一部分。最后, 使用 AG2 作为完全自动化系统也推动了直接从自然语言输入可靠地解决几何问题的进展。

关键词：数学；定理证明；语言模型；搜索

目录

A.1 引言

国际数学奥林匹克竞赛（IMO）是著名的全球高中生数学竞赛。IMO 问题以其难度而闻名，解决这些问题需要对数学概念有深刻的理解和创造性应用。平面几何是 IMO 的四个类别之一，由于其在问题叙述的统一性和规范性，是最容易入手的研究机器证明的题目类别，也适合用于基础推理研究。平面几何的自动定理证明主要有两种方法。一种是通过吴方法[3-4]、面积法[5-6]或格罗布纳基底等代数方法[7-8]解决几何问题；第二种方法依赖于综合技术，如演绎数据库[9]或全角法[10]。我们专注于后者，因为这是一种更接近人类思维方式的方法，并且该方法可能能够迁移到其他数学领域。在我们之前的工作中[2]，我们介绍了 AlphaGeometry（AG1），这是一个符号-神经系统，它在解决平面几何问题上取得了显著进展，在 2000-2024 年 IMO 几何问题上的解题率达到了 54%。AG1 结合了语言模型（LM）和符号引擎，有效解决了这些具有挑战性的问题。

尽管 AG1 取得了成功，但在几个关键问题上仍存在局限性。其性能受到领域特定语言范围、符号引擎效率和初始语言模型容量的限制。因此，当考虑 2000 年至今的所有近期 IMO 几何问题时，AG1 的解题率仅为 54%。

本文介绍了 AlphaGeometry2（AG2），AG1 的升级版本，解决了上述的局限性并显著提升了性能。AG2 利用了基于 Gemini[11]的语言模型，并且在更大规模和更多样化的数据集上进行了训练。我们还引入了一个更快、更强大的符号引擎，加入了优化措施，如减少规则集 (rule set) 和增强对双点情形的处理。此外，我们将领域语言扩展到涵盖更广泛的几何概念，包括轨迹定理和线性方程。为了进一步提升模型性能，我们开发了一种新颖的搜索算法，该算法探索更广泛的辅助构造策略，并采用知识共享机制来扩展和加速搜索过程。最后，我们在构建一个完全自动化且可靠的系统方面取得了进展，该系统能够直接从自然语言输入解决几何问题。为此，我们利用 Gemini 将问题从自然语言翻译成 AlphaGeometry 语言，并实现了新的自动图生成算法。这些改进最终带来了性能的显著提升：AG2 在 2000-2024 年所有 IMO 几何问题上的解题率达到了 84%，这表明 AI 在解决具有挑战性的数学推理任务方面取得了重大进步，并超越了平均 IMO 金牌得主的水平。

Alphageometry2 的关键进步有：

- 扩展的领域语言：涵盖轨迹定理、线性方程和非构造性问题陈述。
- 更强大且更快的符号引擎：优化的规则集、增加的双点处理能力以及更快的 C++ 实现。
- 先进的新型搜索算法：利用多个搜索树进行知识共享。
- 增强的语言模型：基于 Gemini 架构，训练数据集更大且多样化。

A.2 更通用的形式化语言

在 Trinh 等人（2024 年）首次介绍的 AlphaGeometry1（AG1）中，使用了一种简单的平面几何形式化语言（DSL），该语言由表 1 中列出的九个基本“谓词”组成。尽管这些谓词足以覆盖 2000-2024 年 IMO 所有几何问题的 66%，但 AG1 语言无法描述线性方程、点/线/圆的运动，以及常见的“求角度……”类型的问题。以下我们解释 AG2 如何解决这些问题以及其他挑战。

表 A.1 AG1 谓词

名称	含义
`cong a b c d`	$AB = CD$
`perp a b c d`	$AB \perp CD$
`para a b c d`	$AB \parallel CD$
`coll a b c`	A, B, C 共线
`cyclic a b c d`	点 A, B, C, D 共圆
`eqangle a b c d e f g h`	AB 和 CD 之间的有向角与 EF 和 GH 的相同
`eqratio a b c d e f g h`	$\frac{AB}{CD} = \frac{EF}{GH}$
`aconst a b c d x`	AB 和 CD 之间的角度等于 x，其中 $x \in [0, 180)$
`rconst a b c d y`	$AB : CD = y$，其中 y 是一个常数

AG2 在上述 9 个谓词表中增加了两个谓词，允许提出“求 x”类型的问题：

- `acompute a b c d` 表示“求 AB 和 CD 之间的角度”。
- `rcompute a b c d` 表示“求比例 $\frac{AB}{CD}$。

在某些几何问题中，包括 2024 年 IMO 的几何，存在几何量（角度、距离）的线性方程，而 AG1 无法捕捉这些内容。为了表达这些概念，AG2 增加了以下三个谓词：

- `distmeq a1 b1 a2 b2 ... an bn t1 t2 ... tn y` 表示 $t_1 \log(AB_1) + t_2 \log(AB_2) + \cdots + t_n \log(AB_n) + y = 0$。

- `distseq a1 b1 a2 b2 ... an bn t1 t2 ... tn` 表示 $t_1AB_1 + t_2AB_2 + \cdots + t_nAB_n = 0$。
- `angeq a1 b1 a2 b2 ... an bn t1 t2 ... tn y` 表示 $t_1\angle(AB_1) + t_2\angle(AB_2) + \cdots + t_n\angle(AB_n) + y = 0$，其中 $\angle(AB)$ 是线 AB 与水平线之间的角度。

另一个 AG1 中不支持的类别是所谓的轨迹问题，这些问题涉及点、线和圆的运动。AG2 通过新的谓词语法捕捉这些内容。表 2 列出了 11 种轨迹类型及其对应的谓词和语法。这里我们引入了一个新的标记 $*$，用作固定点占位符。

表 A.2　轨迹类型语句及其对应的谓词语法

案例名称	子案例	语法
圆通过固定点	a,b,c 的外接圆	? cyclic a b c * : X
	以 a 为圆心，bc 为半径的圆	? cong b c a * : X
线通过固定点	直线 ab	? coll a b * : X
	a b 的中垂线	? cong a * b * : X
	过 a 且与 bc 平行的直线	? para b c a * : X
	过 a 且与 bc 垂直的直线	? perp b c a * : X
点在固定线上		? coll a * * : X
点在固定圆上		? cyclic a * * * : X
固定距离		? cong a b * * : X
固定方向		? para a b * * : X
固定角度		? eqangle a b a c * * * * : X

进一步的改进

此外，在 AG2 的证明中，我们引入了显式的谓词来表示图的拓扑和非退化条件：

- `sameclock a b c d e f` 表示方向 $A \to B \to C$ 与 $D \to E \to F$ 有相同的顺时针方向。
- `noverlap a b` 表示点 A 和 B 是不同的点。
- `lessthan a b c d` 表示 $AB < CD$，这在 SSA 三角形全等定理中需要被使用。

AG2 还通过引入新的谓词 `overlap a b` 来表达点的重合性（即点 A 和 B 重合），即涉及 A 的任何谓词也可以用于 B，反之亦然。在演绎闭包 (DD) 过程

中，重合点可以通过定义为同一个圆的中心来定义；因此，我们引入了另一个谓词 `cyclic_with_center` 来捕捉这种情况。这里，`cyclic_with_center a1 a2 ... an x` 表示 $a_1 = a_2 = \cdots = a_x$ 是通过 $a_{x+1}, \cdots, a_n$ 的圆的中心（如果 $n = 0$，则等同于 `cyclic`）。

在描述问题时，AG1 最多使用 2 个谓词来定义一个点，即每个点都是通过最多两个对象（线或圆）的交点来定义的。这限制了 AG1 只能处理构造性问题 (constructive problem)——即所有点都可以通过遵循其定义顺序并取两个明确定义的对象的交点来直接构造的问题。在 AG2 中，我们放宽了这一约束，以涵盖更多问题，其中点可以通过至少三个谓词来定义，使得图的构造变得非平凡。这一过程自动化的方法将在下一节中讨论。

所有这些改进将 AG 领域的语言覆盖率从 66% 提高到 88%，在 2000-2024 年 IMO 所有几何问题上。剩余的 12% 包含 3D 几何、不等式、非线性方程和可数点问题（即问题中有 n 个点，其中 n 是任意正整数）。所有被 AG1 和 AG2 覆盖的问题（以及未被覆盖的问题）可以在图 8 中找到。未被覆盖的问题被称为“未尝试”。

A.3 自动化问题形式化和几何图形生成

自动化形式化

AlphaGeometry 及其类似神经符号系统的一个主要弱点是需要手动将输入问题从自然语言转换为领域特定语言。例如，一个简单的几何问题用自然语言表述为：“给定一个等腰三角形 ABC，其中 $AB = AC$，证明 $□B$ 和 $\angle C$ 相等”，在 AlphaGeometry 领域语言中则表示为：

```
triangle a b c; a b = a c ? eqangle b a b c c b c a
```

将这一过程自动化，称为自然语言的形式化 (formalization)，是当前研究的活跃领域[12-15]。与人类语言之间的翻译相比，这一问题要复杂得多。尽管翻译旨在保留意义，但形式化通常需要重新表述原始问题的另一种形式，并且有时需要消除原始问题陈述中的细微差别。因此，自动化形式化（auto-formalization）本身就需要相当丰富的背景知识和问题解决技能。鉴于最近的基础大语言模型开始展现出这种能力，我们使用了其中的一个模型，Gemini[11]，来为 AlphaGeometry 自动化问题形式化。我们首先手动将几十个几何问题翻译成 AG 语言，然后使用这些示例编写一个少样本提示，要求 Gemini 将给定的几何问题从自然语言翻译成 AG 语言。我们用这个提示向 Gemini 查询五次，随后再进行一次 Gemini 调用，要求将这些结果合并为一个最终答案。通过这种方法，我们能够形式化 2000-2024 年 IMO 39 个可形式化的几何问题中的 30 个。对于较简单的几何问题几乎不会出错。

自动化图形生成

我们流程中的另一个部分是图形生成。在 AG1 中，每个点都是通过最多两个表 1 所展示的基本谓词定义的，因此问题可以通过构造性方式定义，并且可以自动生成图。在 AG2 中，我们允许一个或多个点同时被任意数量的谓词定义，这使我们能够涵盖非构造性问题。考虑一个非构造性问题的表述："设 ABC 为一个三角形，其内心为 I，使得 $AI = 2BI\cdots\cdots$"，这里点 I 不仅被定义为内心，即两条内角平分线的交点，还被第三个谓词 $AI = 2BI$ 定义，而没有一般策略可以构造出这样的四个点。由于 AG2 涵盖了非构造性问题，图的构造成为流程中非平凡的部分，通常需要人工干预。类似于 Krueger 等人[16]的方法，我们提出了以下算法，用于根据非构造性问题的自动图形生成：

设 $\bar{x} \in \mathbb{R}^{2n}$ 为一个向量，表示所有点的所有坐标。我们将图中的每个约束 C_i，包括目标，编码为 $\phi_i(\bar{x}) = 0$，其中 ϕ_i 是一个非线性函数。我们通过数值方法搜索合适的 $\bar{x}$，分为两个步骤。首先，我们在均方误差损失 $\sum_{C_i \in C} \phi_i(\bar{x})^2$ 上运行 ADAM 梯度下降优化，其中 C 是所有约束的集合，同时加上一个非退化损失。对于每两个点 A 和 B，我们加上形式为 $\frac{1}{|AB|^2+\epsilon}$ 的损失，并对所有点进行 L_2 归一化，以防止其值变得过大。当 ADAM 优化中的损失达到某个阈值时，我们停止，并从梯度下降优化切换到高斯-牛顿-列文伯格方法，以寻找一个组合的欠定和超定非线性方程组的数值解。

这种方法建立在 Krueger 等人[16]提出的方法基础上。第一阶段保持不变的同时，我们引入了一个新的第二阶段。这一补充解决了在原始方法中调整梯度下降优化时遇到的实际限制，实现了的误差范围。

我们在 44 个用 AG 语言形式化的 IMO 问题上对这种方法进行了基准测试（见图 8），并能够为其中 41 个问题找到图。我们使用多个并行进程运行两阶段收敛过程，并在失败后重启并在另一个随机初始配置上运行。通过这种方式，40/44 问题在 1 小时内使用大约 40 个进程为每个问题生成了图（许多问题在第一次尝试中几秒钟内就生成了图）。对于剩下的 4 个问题，我们运行了更长时间的相同程序，并使用了更多的并行化。通过这种方式，我们也为 IMO-2011-6 生成了图，耗时 400 分钟，使用了 3333 个进程。

A.4 更强更快的符号引擎

符号引擎是 AlphaGeometry 的核心组件。我们将其称为 DDAR（Deductive Database Arithmetic Reasoning），它是一种计算演绎闭包的算法，即给定一组初始事实，推导出所有可推导的事实。DDAR 通过遵循一组固定的演绎规则，迭代

地将新事实添加到演绎闭包中，直到无法再添加新事实为止。

DDAR 既驱动了我们语言模型的训练数据生成，也在测试时的证明搜索中发挥作用。在这两种情况下，速度都至关重要。更快的数据生成可以允许更大规模和更大胆的数据过滤，而更快的证明搜索则可以进行更广泛的搜索，从而增加在给定时间预算内找到解决方案的可能性。

在 AG2 中，DDAR 的改进主要体现在三个方面：

- 处理重合点的能力。
- 更快的算法。
- 更快的实现。

4.1 处理重合点

在重新实现 DDAR 时，我们尝试保持与原始算法大致相同的定理集和逻辑强度，只是由于实现差异而略有增强（例如，用更一般的中心角定理替换了 Thales 定理）。然而，DDAR1 缺少一个关键功能，这对于解决难题至关重要：它无法接受具有不同名称但相同坐标的两个点。

例如，假设我们需要证明两条线 AB 和 CD 的交点 P 位于某个圆 ω 上。最可能的方法是通过“改写”推理来实现这一点：与其证明交点 P 位于 ω 上，我们可以通过引入一个辅助构造点 P' 作为 AB 和 ω 的交点，然后证明 P' 位于 CD 上，从而得出 $P = P'$，因此 P 位于 ω 上。

为了实现这种“改写”推理，我们通过以下四个步骤处理重合点：

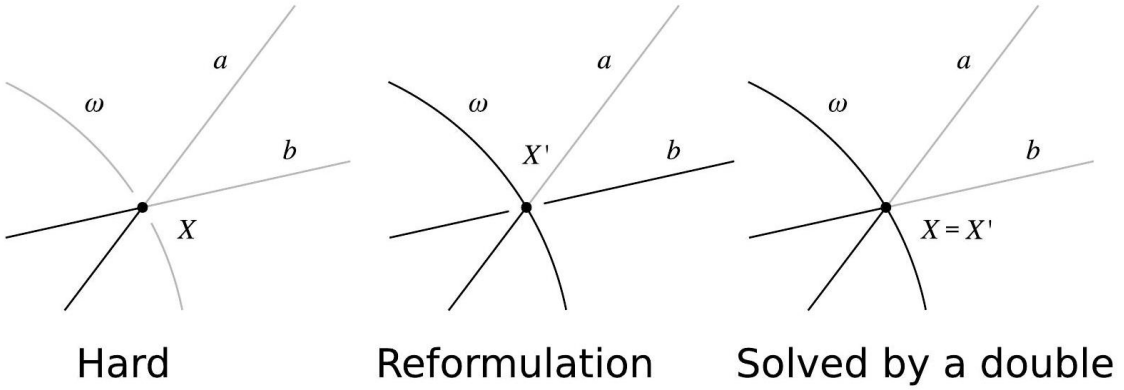
1. 构造一个新的点 P' 作为 AB 和 ω 的交点（我们还不知道 P' 是否与 P 重合）。这是一个必须由语言模型预测的辅助构造。
2. 证明 P' 位于 CD 上。
3. 由于 P 和 P' 都位于 AB 和 CD 上，我们得出 $P = P'$。
4. 因此，P 位于 ω 上。

4.2 更快的算法

DDAR1 算法处理一组规则，并尝试将每个规则应用于所有点的组合。这一过程涉及一个候选搜索步骤，其时间复杂度是点数的多项式。一个子句匹配步骤，其时间复杂度是每个前提子句数的指数。理论上，AG1 中比较耗时的步骤，如搜索相似三角形的最坏情况是 $O(n^8)$。指数级的子句匹配步骤是另一个耗时的部分。

为了使搜索更高效，我们对所有基本规则进行了硬编码 (hard-core) 搜索，将其应用的查询减少到最多三次方。此外，我们在 DD 丢弃了和角度、距离有关的规则（如平行、垂直等）——所有这些推理都自动在算术推理（AR）子模块中进行。

图 A.1　不同重合点的处理



在 AG2 中，我们设计了一种改进的 DDAR2 算法。对于相似三角形，我们遍历所有三个点的组合，对它们的“形状”进行哈希，并在形状被识别两次时检测到相似三角形对。对于圆内四边形，我们遍历所有点对（点 X 和线段 AB），并对 $(A, B, \angle(AXB))$ 的值进行哈希。如果这样的三元组重复出现，我们就得到了一个圆内四边形。这里线段 AB，$\angle AXB$ 的“值”指的是 AR 子模型计算得到的符号规范形式，这个子模型会追踪角、距离、log-距离的线性方程和代数关系，并把任何线性关系归约成标准形式。

4.3 更快的实现

在采用新算法加速 DDAR 的同时，我们通过用 C++ 实现其核心计算（高斯消元），从而进一步提高了速度。新的 C++ 库通过 pybind11[17] 导出到 Python，比 DDAR1 快 300 倍以上。

为了测试速度提升，我们选择了一组 25 个 IMO 问题，这些问题无法用 DDAR1 解决，并在一台配备 AMD EPYC 7B13 64 核 CPU 的机器上运行了 50 次测试。平均而言，DDAR1 完成计算需要 1179.57 ± 8.055 秒，而 DDAR2 仅需 3.44711 ± 0.05476 秒。

A.5　更好的合成训练数据

符号引擎的合成训练数据是 AG1 成功的关键因素之一，将解题率从 14% 提升到 25%（Trinh 等人，2024 年）。AG2 使用了类似的合成数据生成方法。

和 AG1 类似，我们生成合成数据开始于在一个随机图里取样，然后用符号引擎去推导里面所有可能的结论。对于每个推出的结论，回溯算法（Traceback algorithm）可以提取其对应的前提，辅助点以及命题的推导步骤。值得注意的是，这一数据合成过程避免了人造问题作为开始的随机图种子，严格地从随机图开始。这种设

计避免了数据污染的可能，并且让机器有探索人类定理库之外知识的可能。这种方法和 TongGeometry[18] 不同，它们依赖人类专家和已知几何图形去指导和过滤几何数据。在 AG2 中，定理生成的图形开始是完全随机的，这些随机种子持续地推动着更好的定理合成。

更大、更复杂的图和更好的数据分布

首先，我们扩大了数据生成的资源，并更加仔细地重新平衡了数据分布。如图 2 所示，与 AG1 相比，AG2：

- 探索的随机图大小是原来的两倍，允许处理更复杂的问题。
- 生成的定理复杂度提高了 2 倍，即点和前提的数量增加。
- 生成的证明复杂度提高了 10 倍，即证明步骤的数量增加。
- 在问题类型之间的数据分布更加平衡。
- 在有辅助点和没有辅助点的问题之间的数据分布更加平衡。

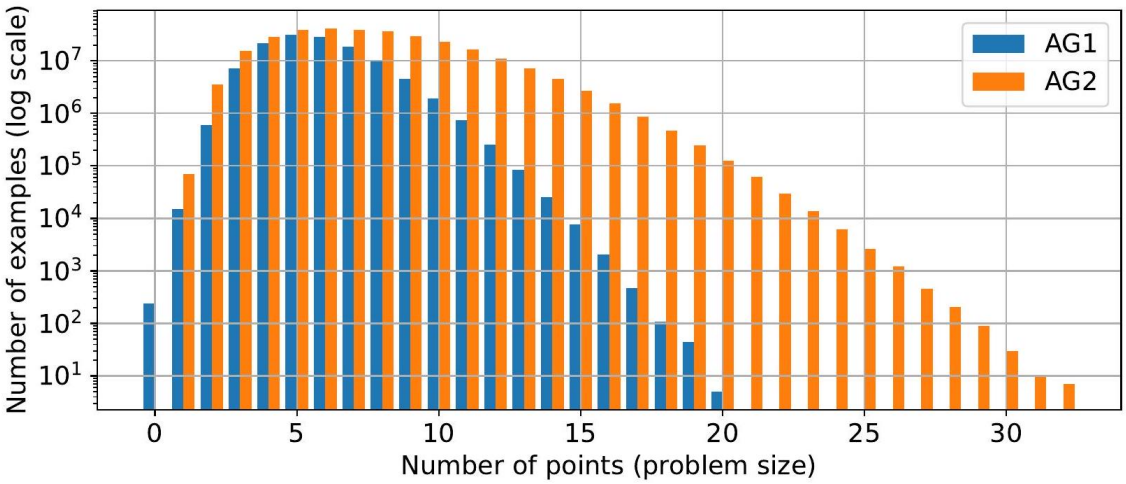


图 A.2　和 AG2 相比，AG1 能够生成更长更复杂的题目

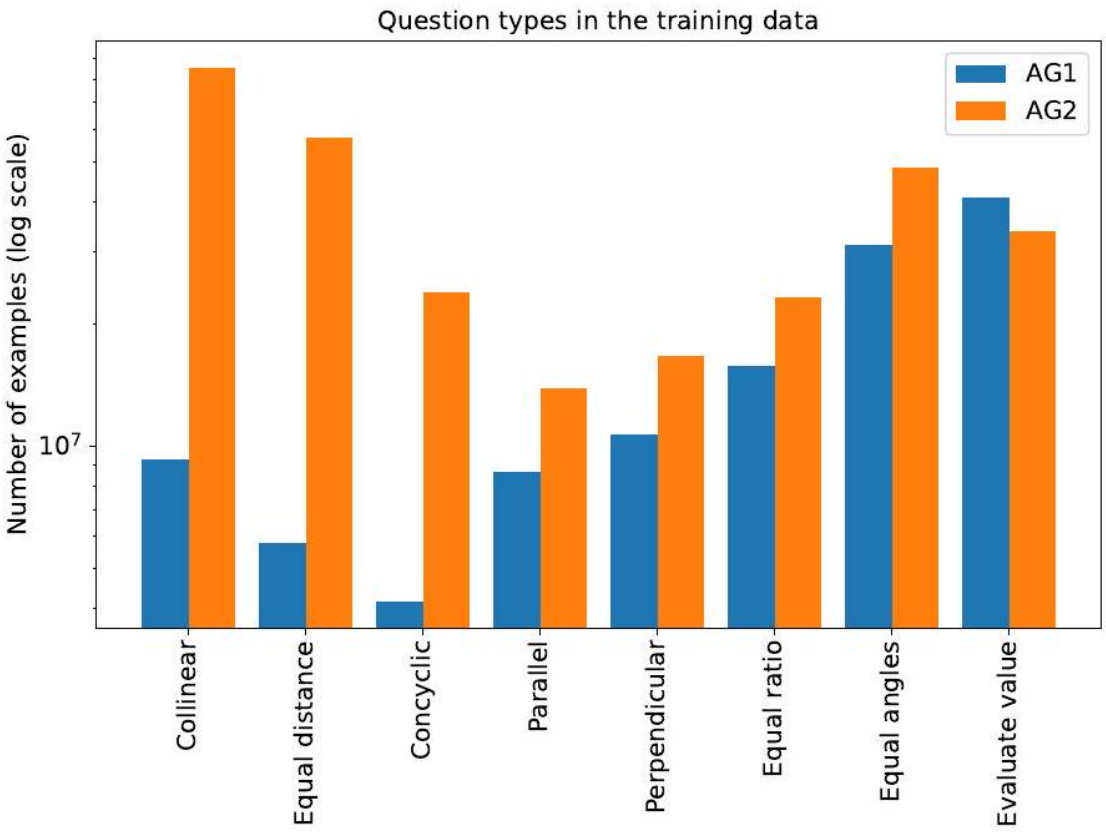


图 A.3　AG2 有更平衡的题目配比

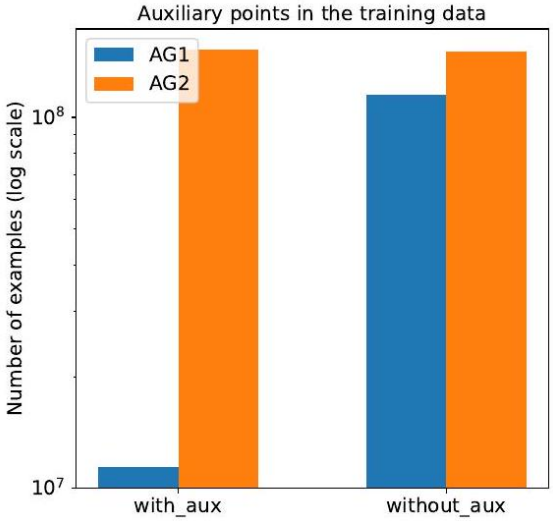


图 A.4　和 AG1 相比，AG2 有更平衡的有辅助点和无辅助点题目的配比

更多类型的定理

除了生成证明经典陈述（如“$AB = CD$”）的定理外，AG2 的数据生成算法还生成了“轨迹”类型的问题，即断言诸如“当 X 在直线/圆 Y 上移动时，Z 在固

定直线/圆 T 上移动”之类的陈述。这些陈述在 AG1 的数据生成算法中不被支持，因为 AG1 没有运动和运动依赖的形式化语言。在 AG2 中，我们在随机图生成过程中记录每个点 X 的运动依赖关系，通过一个函数 $P()$ 来表示:

$P(A)$：控制 A 移动的点集，其中 A 是构造性命题的一个点或一个点集。两个 P 有关的例子在下表中呈现：

表 A.3　两个运动点的例子

If	Then
a = midpoint bc, d = midpoint ac	$P(d) = \{b, c\}$
a = on_line bc	$P(a) = \{a, b, c\}$

上表是构造运动点的两个例子。第一行中，由于 d 唯一地定义为 a 和 c 的中点，而 a 唯一地定义为 b 和 c 的中点，因此 d 的运动来源是 b 和 c。第二行中，由于 a 可以在直线 bc 上的任意位置，因此 a 本身也是其运动来源的一部分。

更快的数据合成算法

我们还提高了数据生成算法的速度。在 AG1 中，我们首先在随机图上运行演绎闭包，然后通过“回溯”算法获得最小问题和最小证明，以证明闭包中的每个命题。为了在 AG1 中获得最小问题，我们必须穷尽地从问题中移除不同的点子集，并重新运行 DDAR 以检查可证明性。这种搜索可以找到最小基数的子集，但由于是指数级搜索，对于点数较多的情况是不可行的。因此，我们切换到图 3 中所示的贪婪丢弃算法，该算法仅使用线性数量的检查来确定一组点是否足以证明目标。贪婪算法保证找到一个最小的点集，只要检查是单调的（如果 $A \subseteq B$，则 check_provable (A) $\Rightarrow$ check_provable (B)）。实际上，我们还要求修剪后的集合在构造依赖关系下保持闭合（以便我们仍然可以运行随机构造）。如果我们将这个条件纳入 check_provable 谓词中，它就不再是单调的了。这个困难可以通过按照图 3 中的算法以逆拓扑顺序处理点来解决（首先处理不依赖于任何其他点的点，最后处理构造的初始点）。

```
def prune_points(
        points : set[Point],
        check_provable: Callable[[set[Point]], bool]):
    pruned = set(points)
    for p in reverse_topological(points):
    if check_provable(pruned - {p}):
```

```
        pruned = pruned - {p}
return
```

A.6 新颖的搜索算法

在 AG1 中，我们使用简单的束搜索（beam search）来发现证明。在 AG2 中，我们设计了一种新颖的搜索算法，其中多个不同配置的束搜索并行执行，并通过知识共享机制相互协助。为了提高系统的鲁棒性，我们为每种搜索树配置使用了多种不同的语言模型。我们称这种搜索算法为共享知识搜索树集合（Shared Knowledge Ensemble of Search Trees, SKEST）。

其工作原理如下。在每棵搜索树中，一个节点对应于一次辅助构造的尝试，随后是一次符号引擎的运行。如果尝试成功，所有搜索树都将终止。如果尝试失败，节点将把符号引擎能够证明的事实写入共享事实数据库。这些共享事实经过筛选，确保它们不是节点本身的辅助点特有的，而是与原始问题相关的。这样，这些事实也可以对同一搜索树中的其他节点以及不同搜索树中的节点有用。下面列出了我们采用的各种类型的搜索树，以确保搜索空间的不同部分得到有效探索：

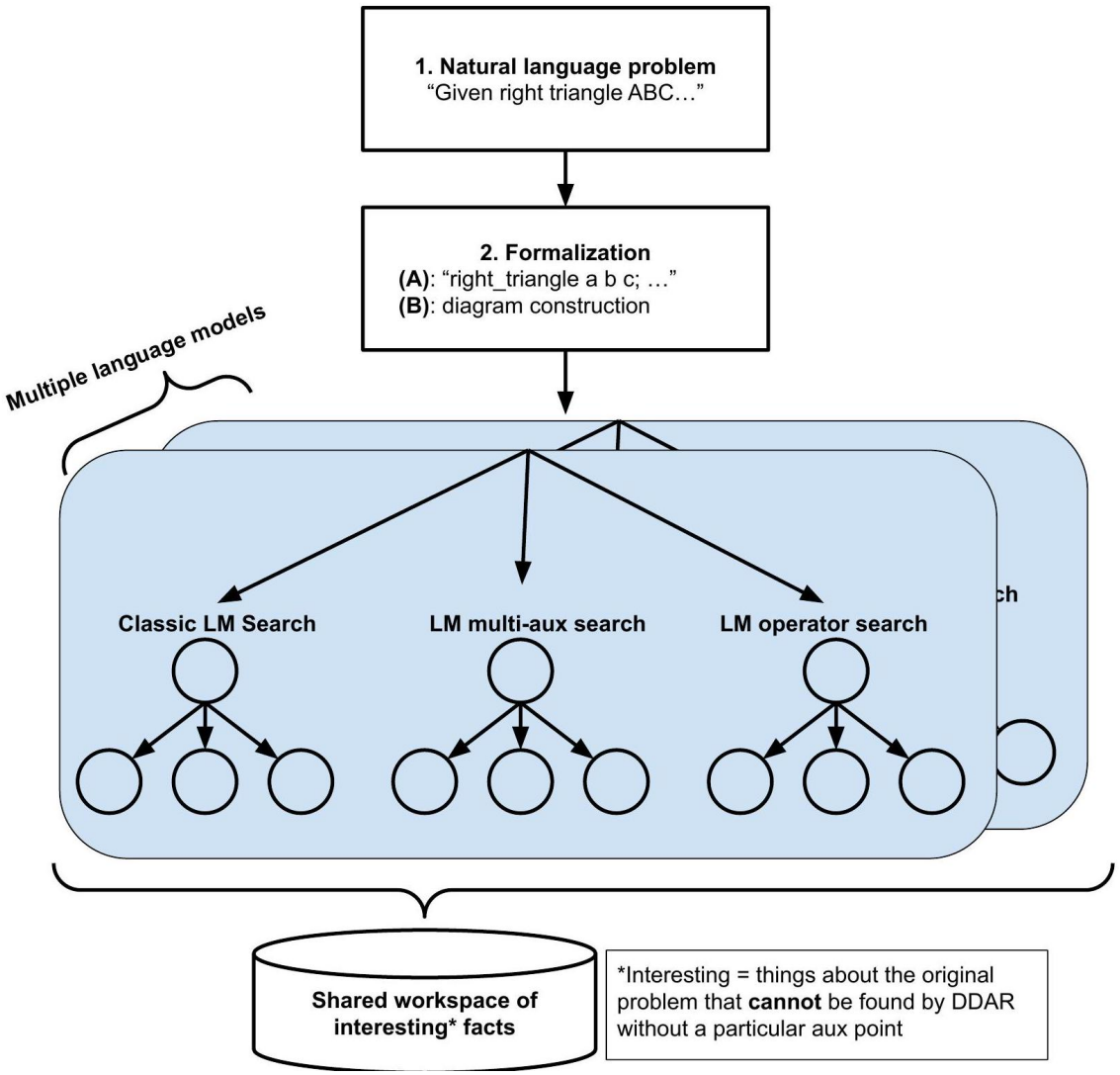


图 A.5　AlphaGeometry2 的搜索过程，我们使用了一些不同的可以共享知识库的树

- “经典”搜索树：与 AG1 中使用的束搜索相同。在每个节点，语言模型会生成一个辅助点。
- 在每个节点预测多个辅助点的树：在每个树节点，允许语言模型生成任意数量的辅助点。因为我们的语言模型被训练来生成完整的证明的过程是从辅助点开始的，所以这种方案可行。需要注意的是，尽管我们希望模型在一次查询中生成所有必要的辅助点，但在实际操作中，我们发现需要根据之前生成的辅助点多次调用模型。允许模型生成多个辅助点可以加速找到解决方案，并有效增加树搜索的深度。
- 均匀预测不同类型的辅助点的树。语言模型输出的辅助点形式如下：`x 00 a : cong a b c d (00) coll a e f (01)`，即“构造点 a，使得 $ab = cd$ 且 aef 共线”。通常，为了预测辅助点，我们提示语言模型使用

第一个标记 `x 00`，然后让它生成其余部分。在这里，我们改用 `x 00 a : cong`、`x 00 a : coll`、`x 00 a : cyclic`、`x 00 a : perp` 的提示词，使得前 4 个 token 均匀分布，并让语言模型生成剩余的部分。

- 深且窄的树 (如宽度 64 深度 10 的树)
- 浅且宽的树（如宽度 512 深度 4 的树）

系统设计细节

对于证明搜索，我们使用 TPUv4 为每个模型提供多个副本，并允许同一模型内的不同搜索树根据各自的搜索策略查询同一服务器。除了异步运行这些搜索树外，我们还异步运行语言模型工作线程和 DDAR 工作线程。语言模型工作线程将他们探索的节点内容写入数据库，DDAR 工作线程异步地获取这些节点并进行尝试。DDAR 工作线程之间相互协调，确保工作均匀分配。一个 DDAR 工作线程池在不同问题之间共享（如果同时解决多个问题），这样，较早解决的问题会释放其 DDAR 计算资源，供其他正在解决的问题使用。

A.7 更好的语言模型

AlphaGeometry2 的最后一个改进是一个新的语言模型。在本节中，我们讨论新的训练和推理设置。

7.1. 训练设置

AG1 语言模型是一个自定义的 Transformer，以无监督的方式分两个阶段进行训练：先训练包含和不包含辅助构造的问题，然后训练仅包含辅助构造的问题。对于 AG2，我们利用 Gemini 训练流程，将训练简化为一个阶段：对所有数据进行无监督学习。我们的新语言模型是一个基于 Transformer 的稀疏专家混合模型，基于 Gemini Team Gemini (2024) 并在第 5 节描述的 AG2 数据上进行训练。我们使用三种训练设置来训练不同规模的多个模型：

1. 使用 AG1 的语言在自定义分词器从头开始训练。
2. 在自然语言中微调已经预训练的自定义数学专用 Gemini 模型（更多细节见附录 A）。
3. 通过增加图像输入的多模态训练——针对给定几何问题的图形（更多细节见附录 B）。

除了大约 3 亿个定理的大型合成训练集外，我们还创建了三个评估集：

1. 包含和不包含辅助点的合成问题集，“eval”。

2. 仅包含辅助点的合成问题集，“eval_aux”。
3. 2000-2024 年国际数学奥林匹克（IMO）中之前被 AlphaGeometry 解决的几何问题的特别集合，“imo_eval”。

所有这些集合都包含完整的证明，在训练过程中我们计算它们的困惑度损失。然而，需要注意的是，这些只是代理指标，原因有两个。首先，在推理过程中（就像在 AG1 中一样），我们只使用语言模型建议的辅助点，而困惑度是针对整个证明计算的。其次，对于给定的问题可能有多种解决方法，但困惑度是针对特定解决方案计算的。与 AG1 一样，我们主要的下游指标是 IMO 问题的解决率，其中语言模型生成辅助点，然后通过第 6 节描述的束搜索进行 DDAR 运行。这些结果将在第 8 节中讨论。

我们使用 TPUv4，以硬件允许的最大批量大小训练模型。学习率调度采用线性预热，随后是余弦衰减。学习率超参数根据扩展规律确定。在图 5 中，我们展示了不同规模的 Gemini 模型的学习曲线，以参数数量表示。不出所料，增加模型规模会降低训练和评估 IMO 评估集的困惑度损失。

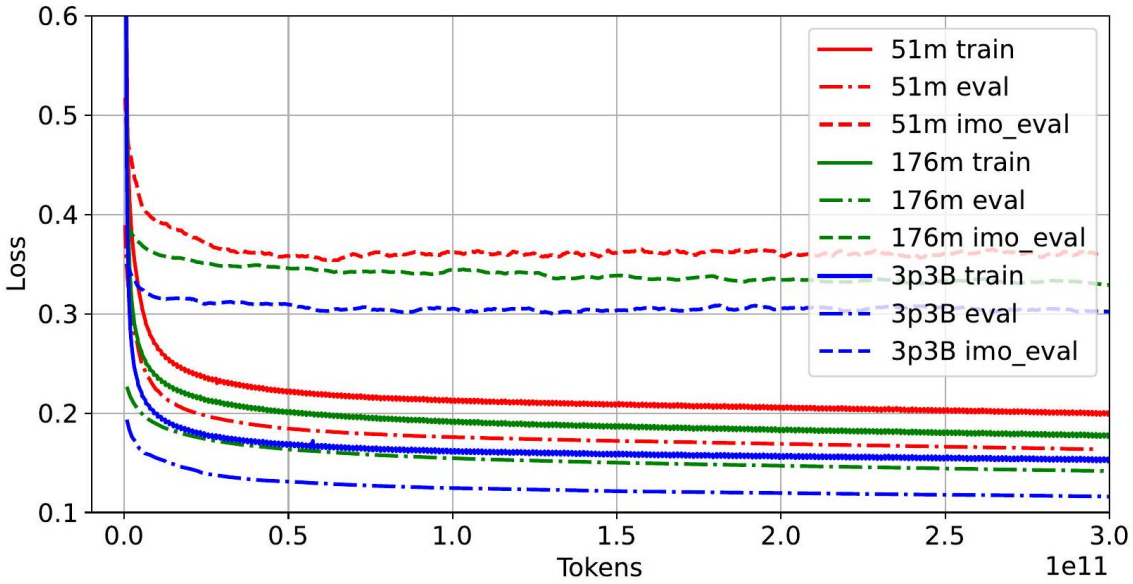


图 A.6　不同规模的 AlphaGeometry2 语言模型的学习曲线，以参数数量表示（“m”表示百万，“B”表示十亿）。增加模型规模会降低训练、评估和 IMO 评估集的损失。

7.2. 推理设置

我们使用了多个搜索树和不同规模的多个语言模型。与 AG1 相比，我们使用温度为 $t = 1.0$ 和 $k = 32$ 的 top-k 采样。需要注意的是，高温度和多个样本对于解决 IMO 问题至关重要。在贪婪解码 $t = 0.0, k = 1$ 且没有树搜索的情况下，我们的模型只能解决 26 个需要辅助构造的问题中的两个。将温度提高到 $t = 1.0$ 并使用

$k = 32$ 个样本（没有搜索树）使我们的语言模型能够解决 26 个问题中的 9 个。较低的温度 $t < 1.0$ 生成的辅助构造多样性不足（见图 6），而较高的温度则导致语言模型输出的数量增加，但语法错误。

字符串分析

在 AG1 中，语言模型（LM）与 DDAR 之间的接口非常简单：DDAR 接受 LM 提出的辅助构造，当 DDAR 成功找到解决方案时，LM 停止提出辅助构造。

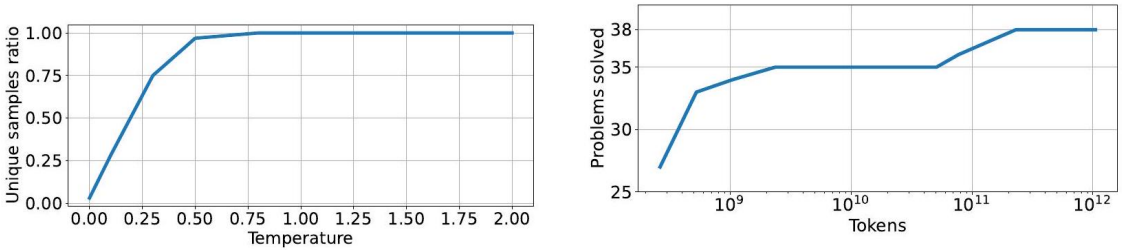


图 A.7　不同温度下 top-k 采样的唯一样本比例和 2020-2024IMO 解决数量和语言模型训练 token 数的关系

在 AG2 中，我们通过让 LM 了解 DDAR 所做的推理来丰富这种神经符号接口，然后再提出辅助构造。具体来说，我们将以下信息输入到 LM 中：

- S_1：给定原始问题前提的 DDAR 可推导的事实集。
- S_2：给定原始问题前提并假设目标谓词也为真的情况下，DDAR 可推导的事实集。
- S_3：数值上正确的事实集（通过检查图形）。

需要注意的是，根据定义，$S_1 \subset S_2 \subset S_3$。一旦计算出这三个集合，我们将它们序列化并连接成一个称为分析字符串的字符串，使用我们的领域特定语言。这个字符串与原始问题陈述一起输入到 LM 中，如下所示：

<problem_statement> serialized (S_1) serialized ($S_2 - S_1$) serialized ($S_3 - S_2$).

相比之下，AG1 的 LM 输入仅为 <problem_statement>。

A.8　结论

我们的主要下游指标是 IMO 几何问题的解决率。2000-2024 年 IMO 共有 45 个几何问题，我们将其转换为 50 个 AlphaGeometry 问题（我们称这个集合为 IMO-AG-50）。由于我们形式化的特定要求，一些问题被拆分为两个。图 8 展示了我们的主要结果：AlphaGeometry2 解决了 2000-2024 年所有 IMO 几何问题中的 42 个，这是首次超过平均金牌得主的水平。更多细节见表 4，其中比较了各种 AG2 配置

与其他系统，如 AG1 和 TongGeometry。我们还对一组新的 30 个最难的 IMO 预选题进行了额外评估，这些问题可以用 AG2 语言形式化，并且从未出现在 IMO 中。这些额外结果见附录 D。

在图 7 中，我们展示了 IMO 解决率与训练时间（训练期间看到的 token 数）之间的关系的关系。有趣的是，AlphaGeometry2 在仅 250 个训练步骤（批量大小为 256），或大约 2 亿个 token 后，已经可以解决 50 个问题中的 27 个。我们还对推理设置如何影响整体性能进行了消融研究（见图 9）。对于单个搜索树，我们发现最佳的配置是宽度 128，深度 4 且采样 32 次，更多的采样会更大的搜索不会帮助解决这个问题。

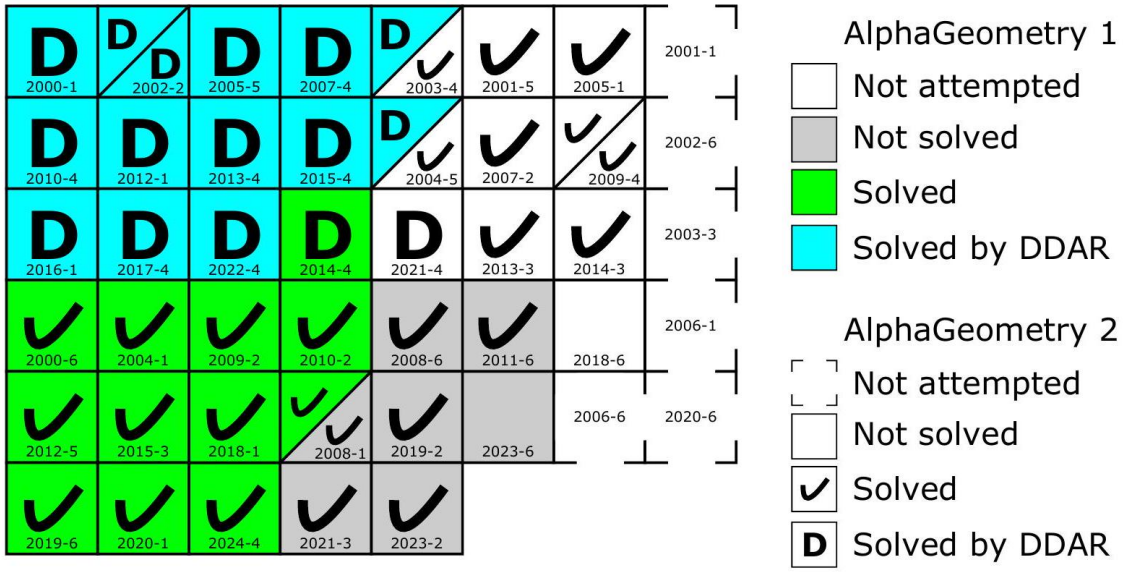


图 A.8　AlphaGeometry2 在所有 2000-2024 年 IMO 几何问题上的结果。问题根据其状态分组，并在组内按时间顺序排列。

系统	IMO-AG-50 解决数	IMO-AG-30 解决数
OpenAI o1	0	0
Gemini thinking	0	0
AG1 DDAR	14	14
AG2 DDAR	16	15
TongGeometry	-	18
平均铜牌得主	27.1	19.3
Wu with AG1	-	21
平均银牌得主	33.9	22.9
AG1	27	25
平均金牌得主	40.9	25.9
Wu + AG1	-	27
TongGeometry w/o value	-	28
AG2 with AG1 setup	38	28
TongGeometry full setting	-	30
AG2 full setting	42	30

表 4 | IMO-AG-50 基准测试的评估结果。IMO-AG-50 包含 2000-2024 年所有 IMO 几何问题，而 IMO-AG-30[8]仅包含可以用 AG1 语言形式化的一个子集。

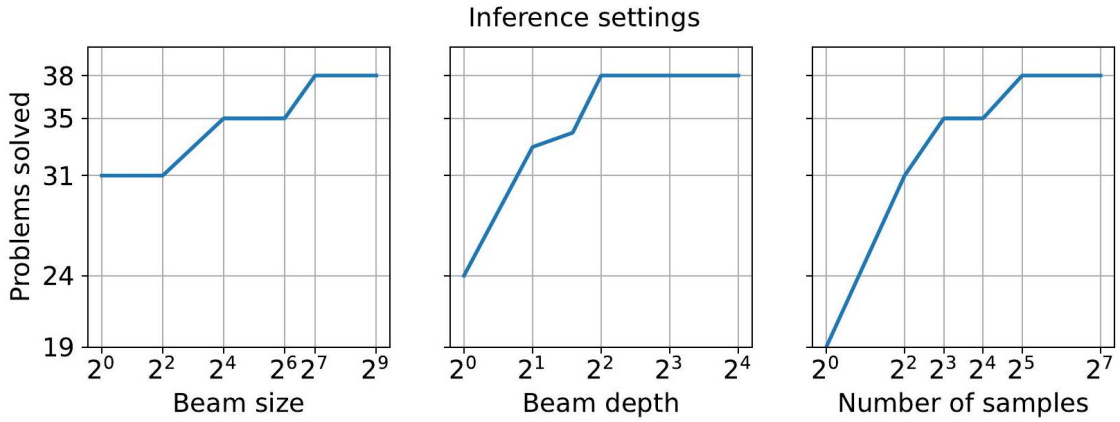


图 A.9　不同推理设置下，2000-2024 年 IMO 几何问题的解决数量，使用单个搜索树。我们从束大小为 512、束深度为 4、32 个样本开始，同时保持其他参数不变，改变其中一个参数。

我们的几何专家和 IMO 奖牌获得者认为，许多 AlphaGeometry 解决方案展现了超人的创造力。在附录 C 中，我们提供了几个这样的例子，并进行了详细分析。在未解决的 IMO 问题中，有 2 个尝试过但未解决，6 个无法形式化。未解决的问题涉及不等式和可变数量的点，这些目前不在 AlphaGeometry2 语言的覆盖范围内。

剩余未解决的两个 IMO 问题（2018 年 IMO 第 6 题和 2023 年 IMO 第 6 题）涉及高级几何问题解决技巧，如反演、射影几何或根轴，这些技巧在我们当前的 DDAR 中尚未实现。虽然从理论上讲，这些问题可以在没有这些技巧的情况下解决，但这样的解决方案需要更长的推理时间、更长的证明和更多的辅助构造，以弥补当前 DDAR 中缺乏上述工具的不足，这限制了 AlphaGeometry 当前的问题解决能力。

A.9 未来工作

本文介绍了 AlphaGeometry2，这是对 AlphaGeometry 的重大升级，解决了之前的局限性，并在几个关键领域提升了性能。AG2 采用了在更大和更多样化数据集上训练的更强大的语言模型、更快且更通用的符号引擎、扩展的领域语言和新颖的证明搜索算法。这些改进使性能有了显著提升，AG2 在 2000-2024 年 IMO 几何问题上的解决率达到 84%，显著高于其前身的 54%。

我们还提出了几项与语言建模相关的研究。首先，我们展示了我们的模型不仅能够生成辅助构造，还能生成完整的证明，这表明现代语言模型有潜力在没有外部工具（如符号引擎）的情况下运行。其次，我们发现对于 AlphaGeometry，无论是用于训练模型的分词器还是领域语言，都不是决定性的因素。我们使用小词汇表的自定义分词器和通用的大型 Gemini 分词器得到了类似的结果。在领域特定语言中训练与在自然语言中训练也得到了类似的结果。第三，我们比较了从头开始训练和微调在数学数据集上预训练的语言模型。我们发现，尽管在相同的 AlphaGeometry 数据集上训练，这些模型学到了略有不同的技能，将它们结合到我们新颖的搜索算法——共享知识搜索树集合中，可以提高整体解决率。

尽管在所有 2000-2024 年 IMO 几何问题上取得了令人印象深刻的 84% 解决率，但仍有改进的空间。首先，我们的领域语言不允许讨论可变数量的点、非线性方程和涉及不等式的问题，解决了这些挑战，才能说完全“解决几何问题”。其次，AG2 尚未解决所有 IMO 和 IMO 预选题。将问题分解为子问题并应用强化学习方法可能可以弥补这一差距。最后，在本文中，我们报告了构建一个完全自动化的几何问题解决系统的进展，该系统以自然语言输入，并可靠地输出解决方案，没有任何幻觉。尽管初步结果良好，但我们认为自动形式化可以通过更多的形式化示例和监督微调进一步改进。

参考文献

[1] Chervonyi Y, Trinh T H, Olšák M, et al. Gold-medalist performance in solving olympiad geometry with alphageometry2[A]. 2025.

[2] Trinh T H, Wu Y, Le Q V, et al. Solving olympiad geometry without human demonstrations[J]. Nature, 2024, 625(7995): 476.

[3] Chou S C. Proving and discovering geometry theorems using wu's method[M]. The University of Texas at Austin, 1985.

[4] Wu W t. On the decision problem and the mechanization of theorem-proving in elementary geometry[M]//Selected Works Of Wen-Tsun Wu. World Scientific, 2008: 117-138.

[5] Chou S C, Gao X S, Zhang J Z. Automated production of traditional proofs for constructive geometry theorems[C]//[1993] Proceedings Eighth Annual IEEE Symposium on Logic in Computer Science. IEEE, 1993: 48-56.

[6] Chou S C, Gao X, Zhang J Z. Machine proofs in geometry: Automated production of readable proofs for geometry theorems: Vol. 6[M]. World Scientific, 1994.

[7] Kapur D. Geometry theorem proving using hilbert's nullstellensatz[C]//Proceedings of the fifth ACM symposium on Symbolic and algebraic computation. 1986: 202-208.

[8] Kapur D. Using gröbner bases to reason about geometry problems[J]. Journal of Symbolic Computation, 1986, 2(4): 399-408.

[9] Chou S C, Gao X S, Zhang J Z. A deductive database approach to automated geometry theorem proving and discovering[J]. Journal of Automated Reasoning, 2000, 25(3): 219-246.

[10] Chou S C, Gao X S, Zhang J Z. Automated generation of readable proofs with geometric invariants: I. multiple and shortest proof generation[J]. Journal of Automated Reasoning, 1996, 17(3): 325-347.

[11] Team Gemini. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context[A]. 2024.

[12] Szegedy C. A promising path towards autoformalization and general artificial intelligence [C]//Intelligent Computer Mathematics: 13th International Conference, CICM 2020, Bertinoro, Italy, July 26–31, 2020, Proceedings 13. Springer, 2020: 3-20.

[13] Wu Y, Jiang A Q, Li W, et al. Autoformalization with large language models[A/OL]. 2022. arXiv: 2205.12615. https://arxiv.org/abs/2205.12615.

[14] Jiang A Q, Li W, Jamnik M. Multilingual mathematical autoformalization[A]. 2023.

[15] Poiroux A, Weiss G, Kunčak V, et al. Improving autoformalization using type checking[A]. 2024.

[16] Krueger R, Han J M, Selsam D. Automatically building diagrams for olympiad geometry problems.[C]//CADE. 2021: 577-588.

[17] Jakob W, Rhinelander J, Moldovan D. pybind11 – seamless operability between c++11 and python[Z]. 2017.

[18] Zhang C, Song J, Li S, et al. Proposing and solving olympiad geometry with guided tree search [A]. 2024.

[19] Deiseroth B, Brack M, Schramowski P, et al. T-free: Tokenizer-free generative llms via sparse representations for memory-efficient embeddings[A/OL]. 2024. arXiv: 2406.19223. https://arxiv.org/abs/2406.19223.

[20] Chae H, Yoon S, Chun C Y, et al. Decomposing complex visual comprehension into atomic visual skills for vision language models[C/OL]//The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24. 2024. https://openreview.net/forum?id=nFU4xCyoe0.

A.A 在 AG 数据上对数学专用语言模型进行微调

尽管在最初过渡到 AG2 时，我们保持了 AG1 的训练设置（使用 AG 领域特定语言中的自定义分词器从头开始训练），但是否可以通过微调已经具备问题解决能力的语言模型来提高性能？由于所使用的分词器和训练语言的差异，这种微调并不立即可行。在本节中，我们探讨了自定义分词器和领域特定语言的作用，随后讨论了在 AG 数据上对数学专用的 Gemini 模型进行微调的问题。

分词器是现代语言模型以及更广泛的基础模型的重要组成部分。普遍认为，分词器可能是模型进行数学操作的主要瓶颈[19]。我们在 AlphaGeometry 的受控环境中研究了这一假设。为此，我们使用不同的分词器训练了相同架构的模型：具有几千个词汇的自定义分词器和具有 30 万词汇的大型语言模型分词器。请注意，我们的自定义分词器是在单词级别创建的，即每个分词器都有完整的意义，而不是子词级别的分词器。在 AG 语言中，分词器有以下几种类型：

1. 点名称：“a”，“b”，“c”，...，“z”，“a1”，...，“z1”。
2. 谓词名称：coll，cong，cyclic，eqangle，eqratio，acompute，rcompute，aconst，rconst，distmeq，distseq，angeq，overlap，noverlap，sameclock，lessthan。
3. 数字和分数：$1, 2, 3, \cdots, -, /$。
4. 谓词引用分词器：(000)，(001)，(002)，... (999)。
5. 保留分词器：{Analysis}，{Numerical}，{FromGoal}，{Proof}，x00，:，;，..

令人惊讶的是，我们发现 AlphaGeometry 在 2000-2024 年 IMO 几何问题上的表现在使用不同分词器时保持相同，这表明现代 LLM 分词器可能在面对不同的数学语境时足够灵活

领域语言 除了分词器外，研究领域特定语言在自然语言解决数学问题中的作用也很有趣。自然地假设，使用领域特定语言可以简化数学操作并防止使用不太严格的语言时可能发生的明显错误。为了研究这一点，我们将所有 AlphaGeometry2 数据从 AlphaGeometry 的形式化语言翻译成自然语言并训练了一个新模型。然后，我们将其性能与在原始 AlphaGeometry 数据上训练的相同大小的模型进行比较。再次令人惊讶的是，我们在 2000-2024 年 IMO 几何问题上得到了相同的结果，这为在数学数据上微调后用自然语言预训练的大型语言模型开辟了道路。下面我们将展示一个将 AlphaGeometry 翻译成自然语言的示例。

AlphaGeometry 语言：`d e f g:coll a d g (000) coll f a b (001) coll d b c (002) coll e c a (003) cong d b d c (004) cong f a f b (005)`

自然语言：构造点 d e f g，使得 a d g 共线（000），f a b 共线

(001), d b c 共线 (002), e c a 共线 (003), |d b|= |d c| (004), |f a| =|f b| (005)

在数学数据上预训练的语言模型的微调 已经表明，自定义分词器和领域特定语言对 AlphaGeometry 并不是关键因素，我们利用在各种数学数据上预训练的语言模型。我们从一个在公共数学数据集上训练的具有 33 亿参数的 Gemini 模型开始，并以无监督的方式在 AlphaGeometry 数据上对其进行微调。在我们的 IMO-AG-50 评估集上，微调后的模型的表现与较小的模型以及从头开始训练的 33 亿参数模型相当。另一方面，我们发现，尽管所有这些模型都在相同的 AG 数据上训练，但它们确实会产生略有不同的辅助点提议，并通过第 6 节中描述的知识共享机制相互帮助，从而形成一个类似集成的系统（参见图 4）。

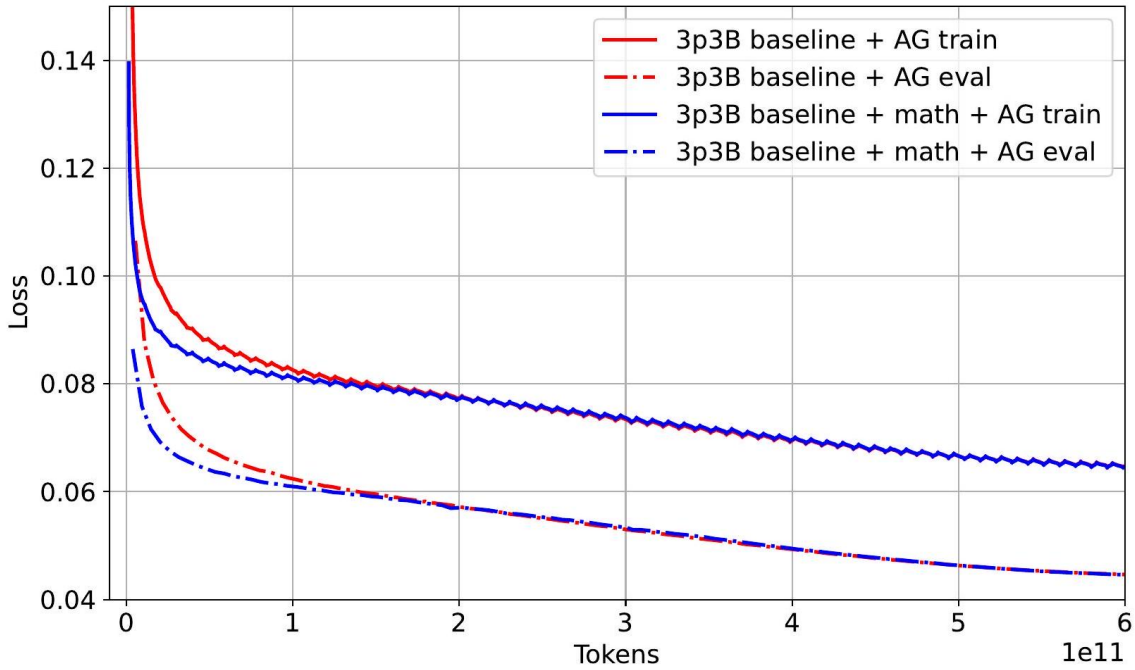


图 A.10 两个 3B 模型的学习曲线：一个是从头开始训练的，另一个是在数学数据上预训练后在 AG 数据上微调的。在数学数据上预训练的模型最初损失较低，但在训练了 200B 个标记后，两者都收敛到同一点。

A.B 多模态

到目前为止，我们讨论了 AG2 作为一个将语言模型与符号引擎结合的系统。然而，由于我们的语言模型基于 Gemini 1.5，且设计上是多模态的（见[11]），因此通过多模态推理增强 AG 模型是很自然的。为此，我们训练了一系列新的模型，这些模型在问题文本的同时，还将相应的图像作为输入。在训练和测试时，图形的构建如第 3 节所述。

尽管在训练过程中取得了良好的结果，但我们发现，当单独使用该模型时，解决下游 IMO 问题的解决率并没有改善。然而，就像在微调预训练模型的情况下（见附录 A），我们发现多模态模型生成的辅助点提案略有不同。通过知识共享机制（见第 6 节）与其他模型结合，这提升了整体性能。我们假设单独添加图像可能不会有太大帮助，因为 IMO 问题的图形非常复杂，变得非常拥挤。图像的分词过程也可能产生负面影响，因为它将图形分割成独立的顺序块，从而导致一些空间信息的丢失。还要注意的是，图形的一些细节已经通过文本提供，如第 7.2 节所述，而我们的符号引擎 DDAR 确实可以访问图形的拓扑方面，例如通过检查同顺时针谓词。此外，Chae 等人[20] 表明，视觉语言模型的原子视觉技能较差，这表明添加视觉元素可能不会有助于几何问题的解决。最后，请注意，几何问题解决的核心在于代数推理，正如[8]之前所展示的，而不是几何推理。许多 IMO 参赛者能够使用计算方法决几何问题（包括非常难的问题，如 IMO 2011 P6），使用的方法包括复数、重心坐标和三角函数，这意味着视觉信息和图形在解决几何问题时并不是关键。

A.C AlphaGeometry2 独特的解题思路

在解决的 IMO 问题中（见图 8），我们的几何专家认为许多 AlphaGeometry 解决方案展现了超人的创造力。一个这样的例子是 2024 年 IMO 第 4 题（见图 11）。

2024 年 IMO 第 4 题：设三角形 ABC 的内心为 I，且满足 $AB < AC < BC$。设 X 是直线 BC 上的一个点，且不同于 C，使得通过 X 且平行于 AC 的直线与内切圆相切。类似地，设 Y 是直线 BC 上的一个点，且不同于 B，使得通过 Y 且平行于 AB 的直线与内切圆相切。直线 AI 与三角形 ABC 的外接圆再次相交于点 P。设 K 和 L 分别为 AC 和 AB 的中点。证明 $\angle KIL + \angle YPX = 180^\circ$。

该问题询问 $\angle KIL$ 和 $\angle YPX$ 之间的关系。前者是由一个中点和内心形成的角，通常这两者并不容易结合，且无法通过三角形 ABC 的角度计算。通常，参赛者会依赖三角函数、复数或其他计算方法来找到解决方案。对于 AlphaGeometry，其 DDAR 系统仅依赖简单的角度追踪和比例追踪，因此需要一些辅助点构造。为此，AlphaGeometry 构造了点 E，使得 $\angle AEB = 90^\circ$，这优雅地将这些看似不相关的几何元素结合在一起，形成了相似三角形对 ABE 和 YBI，以及 ALE 和 IPC。这些相似三角形对产生了新的等角和等边长比。也就是说，点 E 为 AB 的中点 L 提供了目的。为了完成证明，我们需要证明 $\angle AIK = \angle BYP$ 和 $\angle AIL = \angle CPX$。为此，我们需要证明三角形 AKI 与三角形 BPY 相似，三角形 ALI 与三角形 CPX 相似，这可以通过边长比追踪来完成，边长比是从上述相似三角形对中获得的。完整的解决方

案已发布在 https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/imo-2024-solutions/P4/index.html。该解决方案在仅仅使用了 30 秒，并由两次获得 IMO 金牌的选手、2024 年 IMO 问题选择委员会主席 Joseph Myers 给予了满分七分的评价。

除了 2024 年 IMO 第 4 题外，AlphaGeometry 还可以用仅 1 个辅助点解决许多具有挑战性的问题，其中一些涉及相当非传统的构造。一个这样的问题是 2013 年 IMO 第 3 题。

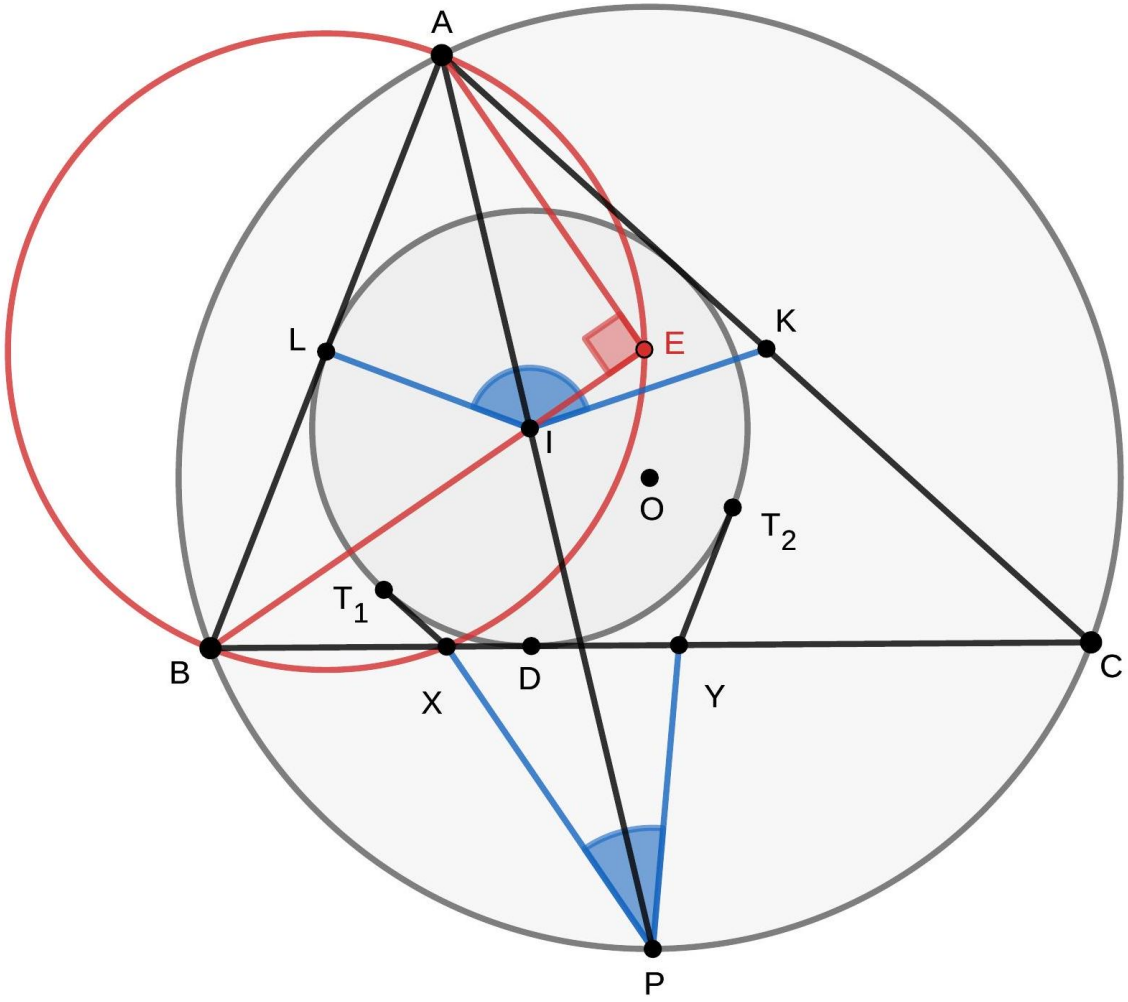


图 A.11　2024 年 IMO 第 4 题的图，包含 AlphaGeometry 的辅助构造点 E。

2013 年 IMO 第 3 题：设三角形 ABC 的 A 对应的旁心圆与边 BC 相切于点 A_1。类似地，设 B_1 在 CA 上，C_1 在 AB 上，分别使用 B 和 C 对应的旁心圆。假设三角形 $A_1B_1C_1$ 的外心位于三角形 ABC 的外接圆上。证明三角形 ABC 是直角三角形。

在这个问题中，AlphaGeometry 简单地取弧 $A\hat{B}C$（包含 B）的中点 D 作为额外点，这是一个高度非传统的构造，因为它不对称。然而，它让 AlphaGeometry 发现

B, A_1, D, I_a 是共圆点，这是一个关键结果，仅当 $AB \perp AC$ 时才成立。为了证明这一事实，AlphaGeometry 利用 O_1 和 D 产生了相似三角形对 $\triangle O_1C_1B_1 \sim \triangle O_1BC$ 和 $\triangle DA_1B_1 \sim \triangle DBA$，然后利用这些结果进行导角，得出 $\angle DA_1I_a = \angle DBI_a$，从而证明 B, A_1, D, I_a 是共圆点。

另一个例子是 2014 年 IMO 第 3 题，这是 IMO 中的一个较难的几何问题。

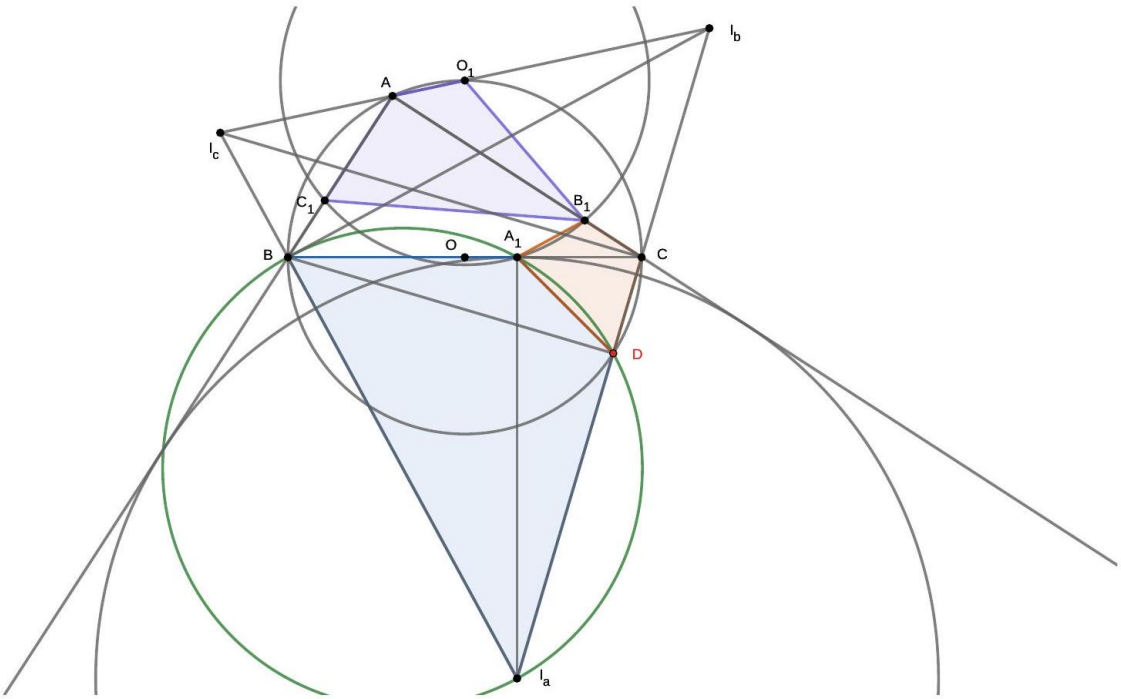


图 A.12　2013 年 IMO 第 3 题的图，包含 AlphaGeometry 的辅助构造点 D。这辅助了证明 BA_1DI_a 是共圆的，成为解决该问题的关键。

2014 年 IMO 第 3 题：凸四边形 $ABCD$ 满足 $\angle ABC = \angle CDA = 90^\circ$。点 H 是从 A 到 BD 的垂足。点 S 和 T 分别在边 AB 和 AD 上，使得 H 位于三角形 SCT 内部，并且 $\angle CHS - \angle CSB = 90^\circ$，$\angle THC - \angle DTC = 90^\circ$。证明直线 BD 与三角形 TSH 的外接圆相切。

令我们惊讶的是，AlphaGeometry 成功地证明了一个更一般的结果 $OH \perp BD$，这意味着当结合原问题中的条件 $H \in BD$ 时，三角形 HST 的外接圆与 BD 相切。为此，AlphaGeometry 构造了点 E, F, G, I，分别是 S 关于 OH 的对称点，H 关于 AT 的对称点，H 关于 AS 的对称点，以及 H 关于 ST 的对称点。由于给定条件 $\angle CHS - \angle CSB = 90^\circ$ 和 $\angle THC - \angle DTC = 90^\circ$ 表明三角形 CHS 和 CHT 的外心分别位于 AB 和 AD 上，因此 F 和 G 的构造创建了圆内接四边形 $CHSG$ 和 $CHTF$，这有助于角度追踪。此外，E 和 I 的构造创建了以 S 为中心的圆内接四边形 $HGIE$，点 C 和 T 现在分别成为三角形 FHI 和 FGI 的外心。综合这些事

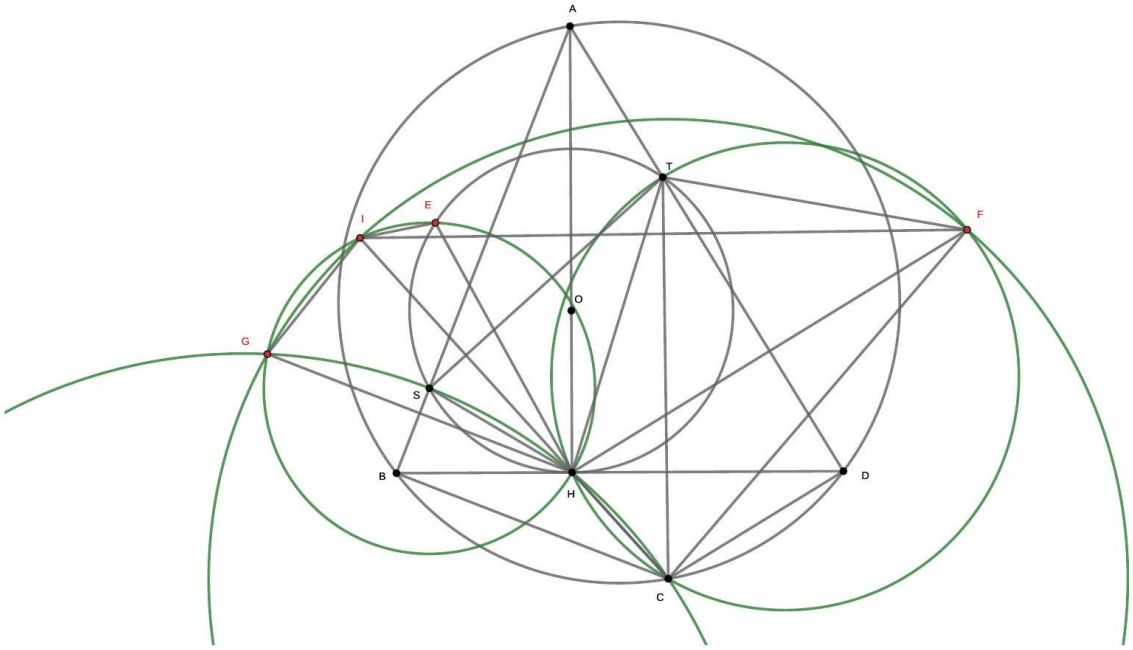


图 A.13　2014 年 IMO 第 3 题的图，包含 AlphaGeometry 的辅助构造。

实，AlphaGeometry 获得了一个非凡的角度追踪证明，与大多数人类参赛者使用的比例追踪（可能结合阿波罗尼奥斯圆的知识）、三角学或反演等常见方法形成鲜明对比。这表明 AlphaGeometry 能够仅凭简单的推理引擎解决难题。

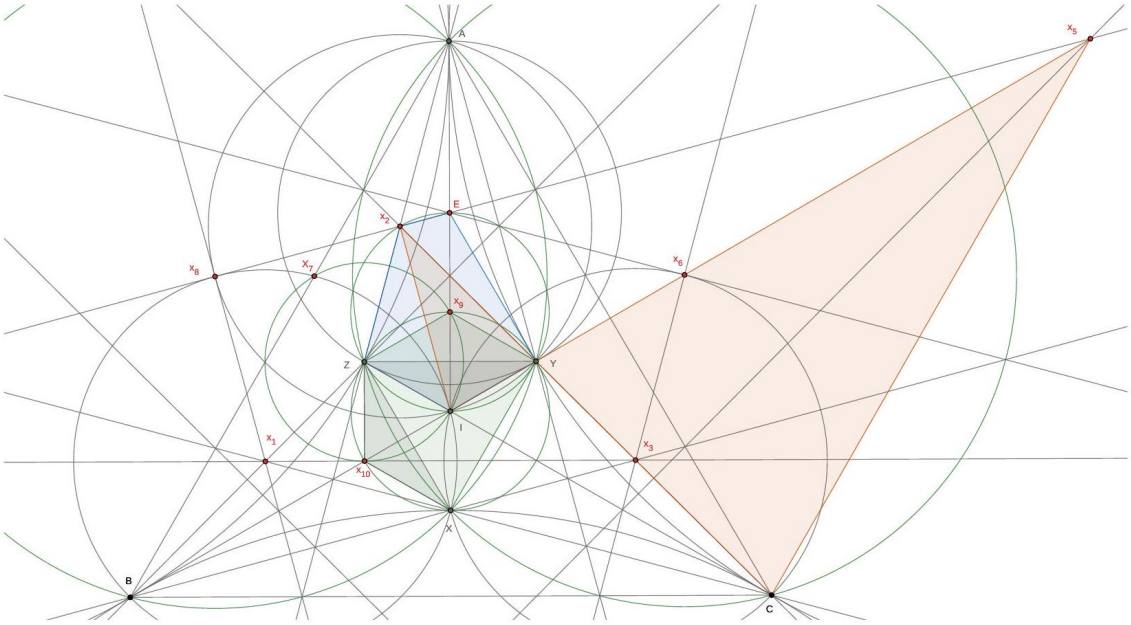


图 A.14　2009 年 IMO 预选题 G7 的图，包含 AlphaGeometry 的辅助构造（红色）、关键的圆内接性质（彩色多边形）和关键的相似三角形对（彩色三角形对）。

我们的最后一个例子是 2009 年 IMO 候选问题 G7。

2009 年 IMO 候选问题 G7：设 ABC 是一个三角形，内心为 I，设 X, Y 和 Z

分别是三角形 BIC, CIA 和 AIB 的内心。设三角形 XYZ 是等边三角形。证明三角形 ABC 也是等边三角形。

据我们所知，这个问题之前只有计算方法的解决方案，例如使用复数、三角计算或通过不等式论证的反证法。由于 AlphaGeometry 无法使用这些计算和推理工具以及高级的欧几里得几何知识，我们原本预计这个问题无法被 AlphaGeometry 解决。然而，AlphaGeometry 通过仅使用角度和比例追踪，构造关键的辅助点，成功地找到了一个优雅的解决方案。首先，AlphaGeometry 证明 X 和 Z 是关于 BI 的对称点，通过对称性可知 I 是三角形 XYZ 的外心。从这一点可以证明 $AB = AC$，通过对称性可知三角形 ABC 是等边三角形。然而，这个问题的主要挑战在于利用三角形 XYZ 是等边三角形的条件，即 $XY = YZ$ 及其类似的对应条件。为此，AlphaGeometry 构造了一系列关键三角形的外心：

1. D 作为三角形 BXC 的外心。
2. E 作为三角形 AYZ 的外心。
3. X_1 作为三角形 BIX 的外心。
4. X_2 作为三角形 AIY 的外心。
5. X_3 作为三角形 CIX 的外心。
6. X_4 作为三角形 ABZ 的外心。
7. X_5 作为三角形 ACZ 的外心。
8. X_6 作为三角形 AXZ 的外心（稍后我们将证明 A, C, X, Z 是共圆的）。
9. X_7 作为 I 关于 BZ 的对称点。
10. X_8 作为三角形 AXY 的外心（稍后我们将证明 A, B, X, Y 是共圆的）。
11. X_9, X_{10} 是使得三角形 IZX_9 和 IZX_{10} 为等边三角形的点。
12. X_{11} 作为 Z 关于 BI 的对称点（使用第 4.1 节中描述的点替换技术，证明其等价于 X）。

起初，这些构造似乎非常反直觉，因为大多数人不会构造这些点。鉴于点 X, Y, Z 的性质，与这些点和这种特定配置相关的几何性质并不多，这使得这个问题对人类来说很难找到综合解决方案。然而，这些外心构造有助于形成相等或相似的三角形对，这使得 AlphaGeometry 能够利用 $\triangle XYZ$ 是等边三角形的事实来解决问题。

所有这些例子表明，AlphaGeometry 在构造辅助点方面非常高效，能够提供相当优雅的解决方案来解决难题，而无需使用复杂的欧几里得几何知识和工具。因此，它能够创造出人类通常不会想到的高效解决方案。

A.D 对最难的 IMO 候选问题的额外评估

为了进一步研究 AG2 的鲁棒性，我们对 IMO 候选问题中未被选入 IMO 的问题进行了额外评估。由于预选题按难度排序，我们从 2002 年至 2022 年每年 IMO 候选问题的末尾中选取了 29 个未出现在 IMO 中的可被形式化的问题。形式化后，我们得到了 30 个问题，并将其称为 IMOSL-AG-30。如图 15 所示，完整的 AG2 系统（见图 4）解决了 30 个问题中的 20 个。这表明，尽管 AG2 是一个非常有能力的系统，可以解决广泛的奥林匹克几何问题，但仍有改进的空间。

A.E 用语言模型生成完整证明

如本文其他部分所述，我们的推理设置仅利用语言模型生成辅助点，然后运行符号引擎。另一方面，模型是在完整证明上进行训练的，因此很自然地会问，模型在不使用符号引擎的情况下生成完整证明的能力如何。鉴于使用贪婪解码时，我们的模型 + DDAR 只能解决 2 个 IMO 问题，因此在没有进一步调整的情况下，模型无法生成完整的证明并不令人惊讶。但模型能否生成部分解决方案呢？为了研究这个问题，我们构建了工具来验证每个演绎证明步骤的有效性。具体来说，我们将证明步骤的前提中的谓词隔离出来，并将其添加到一个新的 DDAR 引擎中，然后仅针对该步骤中使用的演绎规则运行演绎闭包。如果新的 DDAR 能够证明该步骤的结论，并且图形中的结论数值检查通过，则认为该步骤是经过验证的。

我们的步骤验证识别以下错误：

- 语法错误：该步骤的语法错误。
- 定理名称错误：该步骤引用了一个不存在的定理（演绎规则）的名称。
- 步骤引用错误：该步骤引用了一个不存在的先前步骤。
- 点未找到错误：该步骤引用了一个不存在的点，或者是一个构造无效的点。
- 数值错误：由于数值不稳定，DDAR 失败。
- 未验证：该步骤的前提在其使用的演绎规则下不能推导出结论。
- 无效辅助点：辅助点无效，因为该步骤的语法错误，或者在几何上无效（例如，两条平行线的交点等）。
- 验证通过：所有检查通过，未发现上述错误。

在评估中，我们使用温度为 1.0 的 32 个样本，对 2000-2024 年 IMO 问题的语言模型进行查询。查询时不使用任何句子结束标记，以便生成完整的证明。然后我们计算模型在所有样本和所有问题中平均生成的有效证明步骤数量。结果表明，我们的模型几乎不会犯语法错误（见图 16），生成的大部分步骤是有效的（完全验证或正确但未验证）。一个令人惊讶的发现是，小型和大型模型的表现相似。这些

结果支持了大型语言模型可以自给自足、不依赖外部工具的观点，但在推理速度得到提高且幻觉问题完全解决之前，这些工具在数学应用中仍将至关重要。

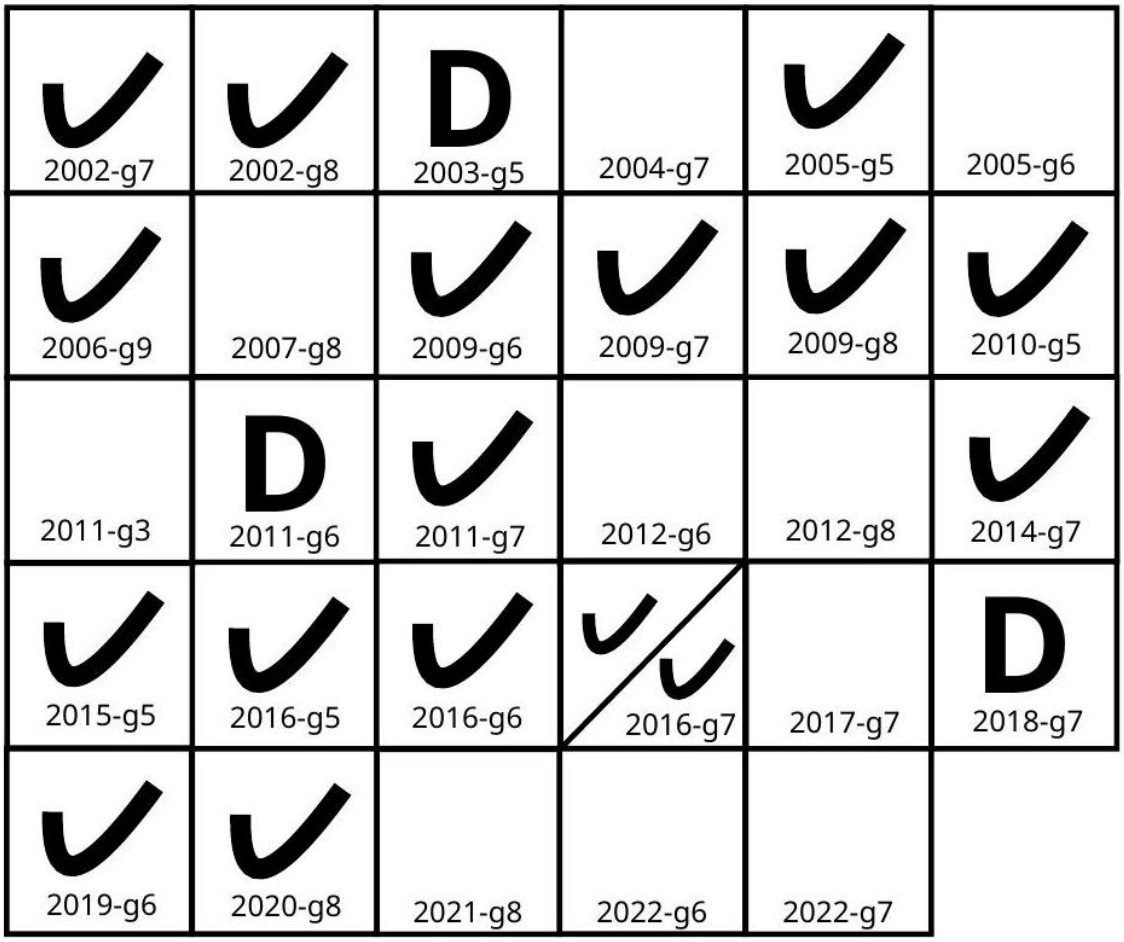


图 A.15　AlphaGeometry2 在最难的 IMO 候选问题上的结果。

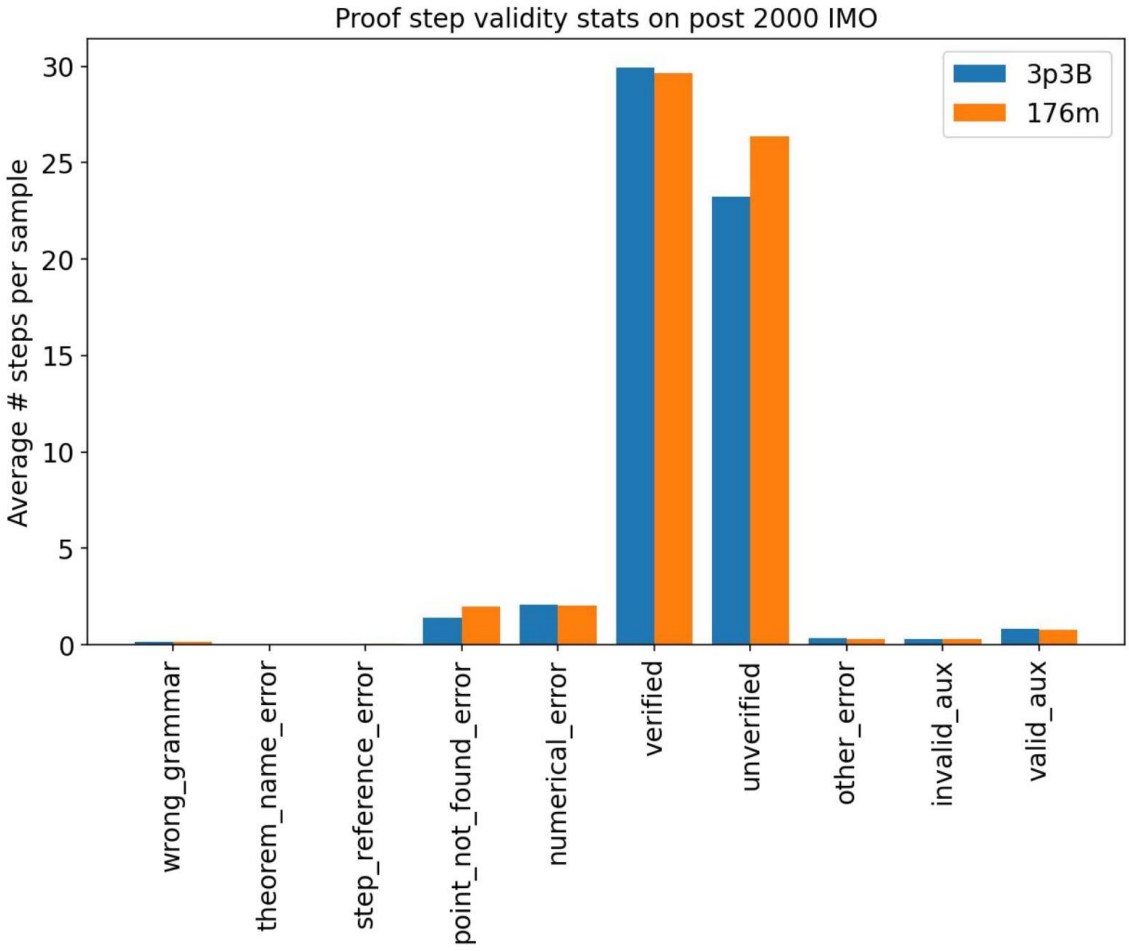


图 A.16　证明步骤有效性统计。模型几乎不会犯语法错误。小型和大型模型表现相似。

附录 B　补充内容

B.1　AlphaGeometry 所用的公理系统

表 B.1　限制性形式化语言

名称	含义
cong a b c d	$AB = CD$
perp a b c d	$AB \perp CD$
para a b c d	$AB \parallel CD$
coll a b c	A, B, C 共线
cyclic a b c d	点 A, B, C, D 共圆
eqangle a b c d e f g h	AB 和 CD 之间的有向角与 EF 和 GH 的相同
eqratio a b c d e f g h	$\frac{AB}{CD} = \frac{EF}{GH}$
aconst a b c d x	AB 和 CD 之间的角度等于 x，其中 $x \in [0, 180)$
rconst a b c d y	$AB : CD = y$，其中 y 是一个常数

表 B.2　构造性形式化语言

名称	含义
a = on_line a b c	A 在直线 BC 上
a = on_circle a b c	A 在以 B 为圆心，BC 为半径的圆上
a = on_pline a b c d	A 在过 B 且平行于 CD 的直线上
a = on_tline a b c d	A 在过 B 且平行于 CD 的直线上
a = on_aline a b c d e f	A 在满足 $\angle ABC = \angle DEF$ 的直线上
a = on_bline a b c	点 A 在 BC 的垂直平分线上
a = on_dia a b c	A 在以 BC 为直径的圆上
a = on_circum a b c d	A 在 BCD 的外接圆上
a = free a	随机找一个自由点 A
a b = segement a b	随机找一条自由线段 AB
a b c = triangle a b c	随机找一个三角形 ABC

B.2　AlphaGeometry 所用的推导规则

```
perp A B C D, perp C D E F, ncoll A B E => para A B E F
```

```
cong O A O B, cong O B O C, cong O C O D => cyclic A B C D
eqangle A B P Q C D P Q => para A B C D
cyclic A B P Q => eqangle P A P B Q A Q B
eqangle6 P A P B Q A Q B, ncoll P Q A B => cyclic A B P Q
cyclic A B C P Q R, eqangle C A C B R P R Q => cong A B P Q
midp E A B, midp F A C => para E F B C
para A B C D, coll O A C, coll O B D => eqratio3 A B C D O O
perp A B C D, perp E F G H, npara A B E F => eqangle A B E F C
   D G H
eqangle a b c d m n p q, eqangle c d e f p q r u => eqangle a b
    e f m n r u
eqratio a b c d m n p q, eqratio c d e f p q r u => eqratio a b
    e f m n r u
eqratio6 d b d c a b a c, coll d b c, ncoll a b c => eqangle6 a
    b a d a d a c
eqangle6 a b a d a d a c, coll d b c, ncoll a b c => eqratio6 d
    b d c a b a c
cong O A O B, ncoll O A B => eqangle O A A B A B O B
eqangle6 A O A B B A B O, ncoll O A B => cong O A O B
circle O A B C, perp O A A X => eqangle A X A B C A C B
circle O A B C, eqangle A X A B C A C B => perp O A A X
circle O A B C, midp M B C => eqangle A B A C O B O M
circle O A B C, coll M B C, eqangle A B A C O B O M => midp M B
    C
perp A B B C, midp M A C => cong A M B M
circle O A B C, coll O A C => perp A B B C
cyclic A B C D, para A B C D => eqangle A D C D C D C B
midp M A B, perp O M A B => cong O A O B
cong A P B P, cong A Q B Q => perp A B P Q
cong A P B P, cong A Q B Q, cyclic A B P Q => perp P A A Q
midp M A B, midp M C D => para A C B D
midp M A B, para A C B D, para A D B C => midp M C D
eqratio O A A C O B B D, coll O A C, coll O B D, ncoll A B C,
   sameside A O C B O D => para A B C D
para A B A C => coll A B C
midp M A B, midp N C D => eqratio M A A B N C C D
eqangle A B P Q C D U V, perp P Q U V => perp A B C D
eqratio A B P Q C D U V, cong P Q U V => cong A B C D
cong A B P Q, cong B C Q R, cong C A R P, ncoll A B C => contri
   * A B C P Q R
cong A B P Q, cong B C Q R, eqangle6 B A B C Q P Q R, ncoll A B
    C => contri* A B C P Q R
eqangle6 B A B C Q P Q R, eqangle6 C A C B R P R Q, ncoll A B C
    => simtri A B C P Q R
eqangle6 B A B C Q R Q P, eqangle6 C A C B R Q R P, ncoll A B C
    => simtri2 A B C P Q R
eqangle6 B A B C Q P Q R, eqangle6 C A C B R P R Q, ncoll A B C
   , cong A B P Q => contri A B C P Q R
eqangle6 B A B C Q R Q P, eqangle6 C A C B R Q R P, ncoll A B C
   , cong A B P Q => contri2 A B C P Q R
eqratio6 B A B C Q P Q R, eqratio6 C A C B R P R Q, ncoll A B C
    => simtri* A B C P Q R
eqratio6 B A B C Q P Q R, eqangle6 B A B C Q P Q R, ncoll A B C
```

```
    => simtri* A B C P Q R
eqratio6 B A B C Q P Q R, eqratio6 C A C B R P R Q, ncoll A B C
   , cong A B P Q => contri* A B C P Q R
para a b c d, coll m a d, coll n b c, eqratio6 m a m d n b n c,
    sameside m a d n b c => para m n a b
para a b c d, coll m a d, coll n b c, para m n a b => eqratio6
   m a m d n b n c
```

致　谢

我想首先感谢我的导师顾陈琳老师。您是我心目中清华最好的老师，是我科研与人生道路上第一位引路人。哪怕我有一点点非常小的进展，您也从不吝惜给我真诚的点赞和鼓励。和您在一起，总能听到数不完的有趣故事。感谢和您“开眼看世界”的那段时光，和您在清芬三楼聊博士生活的三条建议，在拾年咖啡馆聊法国周末的旅行，在天元聊法国的小偷和波兰的炖菜，在办公室聊巴黎的左派大学……您的友善、热情与幽默感染着我，在我最迷茫困惑的大三给了我坚定走下去的勇气与信心。

感谢何凌冰老师。和您一起度过了难忘的五个学期。感谢您在课堂上和我们分享数学与人生，在茶话会上聊过去与未来。感谢您精心准备的每一个数学定理和每一个 joke。感谢您推荐我参加学堂班，去外国上暑校，让我有机会接触一个完全不一样的世界。曾经看您的讲义、听课、考试都是一种痛苦。但正是这些刻骨铭心的经历，让我对这个园子里的数学课有了难忘的集体记忆。

感谢刘恩至老师，您的马原课是我在清华上过最好的课。我做梦也不会想到，在本该划水的思政课上，我能接触到可能影响我一辈子的世界观。大二上学期每个周四的上午，似乎整个世界都在新理论的光茫笼罩下下黯然失色。

感谢大学的朋友们。感谢耿哥，没有你我可能坚持上完何老师的分析课，未来的我会怀念和你一起指点江山，笑傲江湖，粪土当年万户侯的日子；感谢万神，智慧与激情，批判与包容。和你的每一次聊天都是思维与洞见的碰撞；感谢茹姐，有你陪伴的日子很开心，还记得我们一起看剧喝啤酒的夜晚，一起在顾老板的办公室吃蛋糕聊天的圣诞；感谢堃哥，凌晨两点在寝室的床上失眠时，回忆起我们一起去赤峰追极光、去蒙古看草原的故事，总能把我带入甜蜜的梦乡。感谢思然，和你一起打开 AI 新世界的大门。感谢 AFC 和美丽的 Yuki 女士，让我在精神孤独时找到家一样的组织。

感谢我的父母。千里之外的你们一直是我大学里最坚强的后盾。感谢你们永远尊重我的想法，感谢你们总是最真诚地祝愿我找到我自己喜欢的生活。每当我在我被攀比、炫耀与嘲讽的洪流席卷时，想起千里之外的母亲还在忙碌地准备着公益讲座和公益咨询，不求回报地对陌生人置以最真诚的微笑，想起父亲用心准备好每一顿饭菜，细致地备着每一堂课。你们用行动教会我，世界上最珍贵美好的东西就是最简单纯粹的，真正的幸福在于内心保持的善意与热爱。能成为你们的孩子，是我最大的幸运。

最后衷心感谢导师包承龙副教授对我毕设科研项目的精心指导。

希望未来的我们，一起回忆起园子里匆匆掠过的瞬间，还能有满满的幸福感。如果有天在地球的某个角落重逢，希望我们还能笑着谈起当年没来得及说完的话——

“只要不失去你的崇高，整个世界都会向你敞开。”

旅行者，巴黎见。

声　明

本人郑重声明：所呈交的综合论文训练论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：宋晨东　　日 期：2025.6.12

综合论文训练记录表

学生姓名	宋晨东	学号	2021013331	班级	致理-数 11
论文题目	平面几何自然语言的形式化、自动构图和定理集构建				
主要内容以及进度安排	1.调研平面几何求解器 AlphaGeometry，并将其本地部署并测试，并深入研究其形式系统。 2.基于 AlphaGeometry 形式化语言的特点，使用“删点”策略，提出全新的限制性形式化语言到构造性形式语言的转换算法，在机器合成的 69K 限制性形式化语言的数据集中达到 86.3%的翻译成功正确率。 3.基于联立坐标求解和搜索回溯，搭建平面几何自动构图系统，在 200 道复杂形式化语言叙述的构图任务中自动作出 123 个几何图形。 4.搭建基于语法检验、数值检验、语义检验的平面几何自动形式化正确性验证管线，构造形式化任务基准测试 100 题，探索提示词中样例数和细微指令对翻译效果的影响。 5.将开源数据集 Numina_Math 通过自动形式化管线获得 2720 道高质量形式化几何习题，在 AlphaGeometry 上进行测试，比较不同几何数据集的特点。 指导教师签字：包承志 考核组组长签字：[illegible] 2025 年 2 月 19 日				
中期考核意见	论文已调研了相关前沿，并提出了全新的转换算法，论文进展顺利，下一步按计划进行，进一步完成实验和测试 考核组组长签字：[illegible] 2025 年 3 月 28 日				

指导教师评语	宋晨东同学的毕业论文主要研究平面几何中的定理自动形式化和图形绘制流程，并进一步利用LLM设计了自动形式化翻译方法，并在大规模数据集上对应证了方法的有效性。在论文训练过程中，宋晨东同学出色地完成了选题、研究、写作及服从各个环节的要求，是一篇优秀的本科生毕业论文。 指导教师签字：何圣柏 2025年 6月 5日
评阅教师评语	本文研究了平面几何的自动形式化，提出了一种构造性语言的自动转换算法，搭建了自动绘图系统 评阅教师签字：史作强 2025 年 6月 5日
答辩小组评语	论文研究了平面几何中的定理自动形式化，提出了自动形式化翻译方法，数值实验验证了方法的有效性。论文写作规范，条理清晰。答辩中宋晨东同学回答问题准确。达到了本科毕业论文训练的要求。答辩小组认为这是一篇优秀的毕业论文，一致同意通过其论文答辩。 答辩小组组长签字：王定立 2025年 6月 5日

总成绩：A

教学负责人签字：[illegible]

2025年 6月 13日