学习排序：SAT 求解器的初始分支顺序

Arvid Eriksson[*, 1], **Gabriel Poesia**[2], **Roman Bresson**[3], **Karl H. Johansson**[1], **David Broman**[1]

[1]瑞典皇家理工学院[2]哈佛大学肯普纳研究所[3]穆罕默德·本·扎耶德人工智能大学

arXiv:2603.07176v1 [cs.AI] 7 Mar 2026

摘要

找到良好的分支顺序是高效求解 SAT 问题的关键，但寻找这样的分支顺序并非易事。因此，在求解前采用基于学习的方法预测良好的分支顺序，具有很大的潜力。本文探讨了如何使用图神经网络作为冲突驱动子句学习（CDCL）SAT 求解器的预处理步骤来预测分支顺序。我们表明，为现有 CDCL SAT 求解器提供良好的初始分支，可以带来显著收益。此外，我们提出了三种标注方法，以可行的方式找到这样的初始分支顺序。最后，我们训练了一个图神经网络来预测这些分支顺序，并通过评估表明，由图神经网络初始化的排序在随机 3-CNF 和伪工业基准测试上实现了显著加速，且能够泛化到远大于训练集的实例。然而，我们也发现，对于变量数量较多的问题以及更困难的工业实例，这些预测无法带来加速。我们认为，这是因为求解器的动态启发式会迅速覆盖所提供的初始化，同时这些实例本身十分复杂，使得图神经网络难以进行预测。

1 引言

布尔可满足性问题（SAT）是典型的 NP 完全问题，所有 NP 问题都可以归约到该问题（Biere 等人，2009）。这类问题是形式化验证和自动定理证明的核心，因此，高效求解这类问题是实现证明自动化的关键一步。最先进的 SAT 求解器依靠多种启发式来驾驭问题固有的组合搜索空间。尽管这些启发式通常在平均情况下表现良好，但固定且由人工设计的启发式在相当多的情况下仍无法有效引导搜索：许多具有实际价值的问题实例，即使借助现有最优秀的 SAT 求解器，仍然无法求解。

与依赖静态规则不同，另一种途径是从数据中学习启发式。具体而言，图神经网络（GNN）能够学习 SAT 实例的非平凡结构（Selsam 等人，2018），而强化学习则可用于为 SAT 求解器学习有效的引导启发式（Kurin 等人，2020）。然而，神经网络推理的计算开销很大，在搜索过程中使用神经启发式往往不如使用速度快几个数量级的简单启发式，即便神经启发式平均而言可能做出更好的决策。

因此，一个重要的开放问题是开发高效的混合求解器，将神经网络的能力与基于规则的先进 SAT 求解器的效率结合起来。其思路是利用神经网络在关键决策中引导求解器，同时依靠基于规则的启发式实现纯粹的效率。目前已有一些此类混合求解器，例如 RDC-SAT（Zhai 和 Ge，2025年），这是一种分治式深度强化学习混合求解器；以及用于不可满足核预测的 NeuroCore（Selsam 和 Bjørner，2019），这两种方法都会偶尔调用图神经网络来辅助某些关键决策。然而，这些方法都需要对求解器进行大幅修改，使其在求解过程中与图神经网络通信。一种侵入性较低的方法是 NeuroBack（Wang 等人，2024年），它为求解器提供

*通讯作者：arveri@kth.se

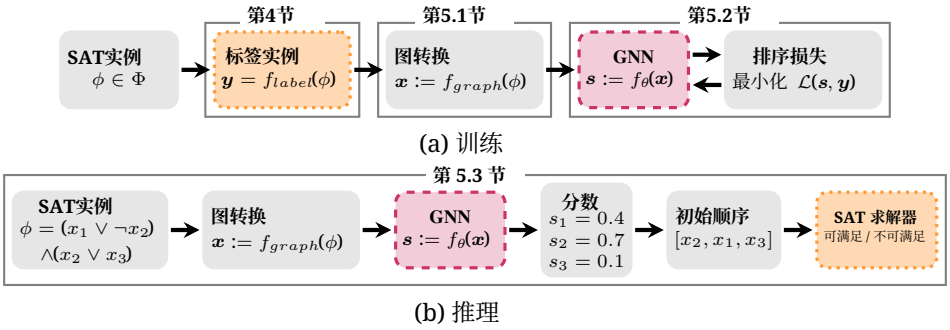


图 1：用于训练（上）和推理（下）的流水线，其中包含一个分支顺序预测图神经网络预处理步骤。橙色方框（点状边框）表示使用修改版 SAT 求解器的步骤，紫色方框（虚线边框）表示使用图神经网络的步骤。

每个变量赋值的初始猜测。然而，这只针对 SAT 求解器的两项主要决策之一：相位选择。另一项关键决策——每一步选择哪个变量进行分支——则保持不变。

本文探讨通过预测初始分支顺序来加速现有的回溯 SAT 求解器。我们的贡献如下：

- 我们通过实验验证，初始分支顺序会显著影响整体求解时间，并展示了良好初始化顺序可能带来的收益（第 3 节）。
- 我们提出了三种启发式方法，用于识别可快速学习的分支顺序，并将其用作预测性图神经网络的标签（第 4 节）。
- 我们展示了一个图神经网络流水线（图 1）能够预测此类分支顺序，并在多个领域实现加速（第 5 节）；此外，它通常能够泛化到比训练期间所用实例大一个数量级的实例。

我们在两种 SAT 求解器——MiniSat（Sorensson 和 Een，2005年）和 CaDiCaL（Biere 等人，2024）——上的经验表明，预测初始分支顺序本身就是一个复杂的问题，会受到许多设计选择的影响。我们的初步实验显示，在生成的 SAT 实例中，准确预测初始分支顺序可以带来显著收益。然而，我们也发现并阐述了在工业实例中获得类似加速所面临的主要挑战；由于这类实例的规模和结构各异，训练也因此面临独特的难题。总体而言，我们的工作为未来研究指出了若干潜在方向：进一步探索混合 SAT 求解器的设计空间，并将变量分支顺序视为神经网络与最先进的求解器之间一个尚未充分探索、却能产生重大影响的交汇点。

2 相关工作

端到端神经 SAT 求解器。使用神经网络求解 SAT 实例是近年来兴起的趋势，其中 NeuroSAT（Selsam 等人，2018）是一项开创性工作。NeuroSAT 将 SAT 求解视为二元分类问题，利用图神经网络预测实例是否可满足。SATformer（Shi 等人，2023）和 SAT-GATv2（Chang 和 Liu，2025年）等多项研究进一步发展了这一思路，并提高了预测准确率。然而，这些方法的预测性质无法为解提供任何可靠保证，而且其准确率会随着实例规模增大而迅速下降。

采用在线神经启发式方法的混合求解器。一种既能保证找到解、又能利用神经网络预测能力的方法，是使用混合求解器——一种基于标准搜索的 SAT 求解器，并通过神经网络提供某种预测能力加以增强。例如，NeuroCore（Selsam 和 Bjørner，2019）将图神经网络集成到现有求解器中，持续覆盖分支顺序，使其转向预测可能包含不可满足核的区域，从而加快求解，同时保留 SAT 求解器所具备的所有保证。Graph-Q-SAT

表 1：强制初始分支从特定变量开始时，与中位数相比，选择最佳变量和第 10 百分位变量所对应的求解时间及加速比分别为多少。

实例	变量数	子句数	中位数（秒）	最佳加速比（%）	前 10% 加速比（%）
3-CNF 100	100	430	0.0007	2233.3	536.4
3-CNF 150	150	645	0.016	63.1	30.8
3-CNF 200	200	860	0.12	13.1	9.1
3-CNF 250	250	1075	0.98	5.6	4.1
Ramsey	37 410	162 400	3.41	0.8	0.5
Manthey	3996	29 400	0.59	2.0	0.9
Queen	256	6744	1.82	6.4	4.6
SATCoin	134 781	648 806	1.26	22.5	5.4

（Kurin 等人，2020）以及 Yolcu & Poczos（2019）提出的图神经网络也会调用图神经网络，但改为在每一步都调用，并通过强化学习来学习最优的分支启发式。同样，RDC-SAT（Zhai 和 Ge，2025年）会在并行 SAT 求解中每次进行分裂时调用图神经网络。然而，上述所有方法都需要频繁调用图神经网络，这对于大型实例而言计算开销很高，而大多数经典求解器启发式的运行时间都是常数级。

采用神经初始化启发式的混合求解器。一种折中的做法是在求解前调用图神经网络一次，并用它初始化求解器的内部结构，从而提供热启动。NLocalSAT（Zhang 等人，2020）将这一方法用于随机局部搜索求解器，通过预测初始解来提供合适的初始猜测。NeuroBack（Wang 等人，2024）则将这一策略应用于冲突驱动子句学习（CDCL）求解器，通过预测每个变量的初始相位（极性）来实现。值得注意的是，NeuroBack 是少数能够证明其在大型工业实例上可为最先进的求解器带来实际运行时间加速比的学习方法之一。然而，相位选择只是回溯求解器的两个关键部分之一，另一个是分支选择。因此，我们的工作首次研究如何为回溯 SAT 求解器预测合适的初始分支顺序。

3 分支顺序对求解时间的影响

众所周知，在回溯 SAT 求解器中，分支顺序对于高效求解至关重要（Nordstr¨om, 2015）。然而，据我们所知，这种影响的大小尚未得到精确刻画。首先，我们通过测量从特定变量开始分支时的求解运行时间，对求解顺序的影响进行实证检验。

我们使用 CaDiCaL（Biere 等，2024）进行以下实验：我们从 SAT 竞赛中随机抽取四个生成的 3-CNF 实例（变量数量特定且均匀随机），以及四个工业界可满足 SAT 实例；这些实例均为 2015年至2024年期间的实例，且 CaDiCaL 能在 10 秒内求解。对于这些实例，我们随机抽取 50 个变量，并对每个抽取的变量进行 1000 次强制优先分支，其余变量的顺序则随机选择。通过这种方法，我们可以量化建议的单变量分支顺序对求解时间的影响。
从图 2 和表 1 中可以看出，对于大多数实例，所选首个变量会显著影响求解器的求解时间。这说明良好的初始分支顺序可能产生重要影响，也表明变量之间存在某种排序，能够区分哪些变量适合优先分支、哪些不适合。有些实例的方差较低，这进一步说明最终的求解时间取决于变量本身，而非随后采用的随机顺序，这也表明存在某种可由图神经网络学习的变量特征。

结果进一步表明，与中位数相比，选择一个合适的首个分支变量有望带来加速（工业实例约为 1% 至 10%）。如果能恰当地选择完整的分支顺序（而不只是首个变量），这种效果还会进一步放大，这也正是本文后续内容的动机。

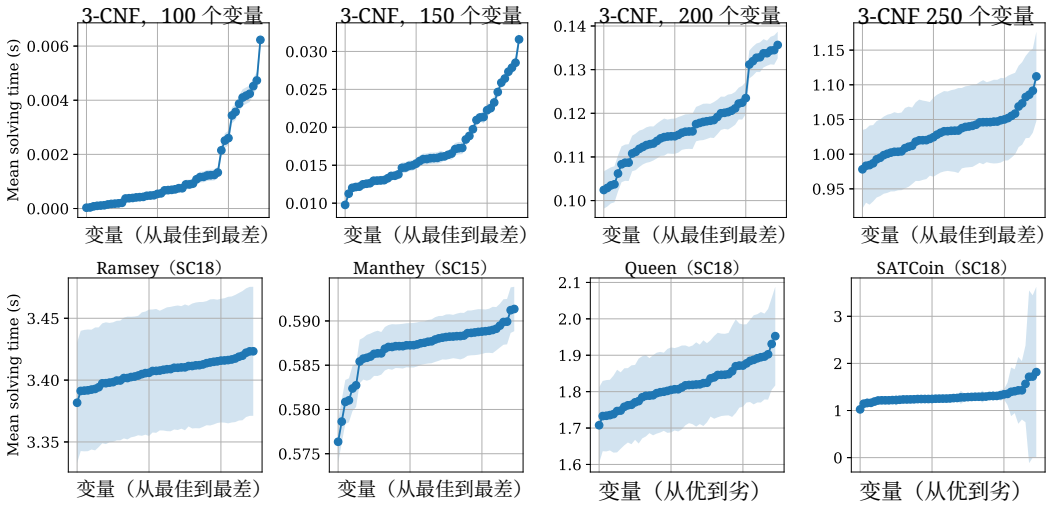


图 2：将随机抽取的 50 个变量之一设为第一个变量、其余变量的顺序随机排列时的平均求解时间。阴影区域表示每个变量经过 1000 次迭代得到的 95%置信区间。

4 识别良好的分支顺序

为了训练图神经网络来预测给定问题的最佳分支顺序，我们首先需要估计该问题的最佳顺序。为此，我们既可以直接使用求解问题时从求解器收集的信息（白盒方法），也可以将求解器视为黑盒，并针对不同分支顺序优化问题的求解时间。我们提出了三种标注方法：冲突标注、首变量标注和优化标注。第 5 节将进一步比较将这些标注用作图神经网络标签时对求解器带来的加速效果。首变量标注和优化标注的伪代码见附录 A。

4.1 冲突标注

对于白盒方法，我们可以根据求解过程中每个变量参与冲突的总次数，对变量进行排序，从而为每个问题标注。这种简单的标注方法只需对问题求解一次，即可完成标注，并优先对容易引发冲突的变量进行分支。这建立在这样一个假设之上：优先对最容易引发冲突的变量进行分支，能够加快求解。不过，这一假设是合理的，因为高冲突变量很可能是后门变量，尽早解决这些变量可以使问题变得简单（Williams等人，2003）。

4.2 首变量标注

首变量标注通过多次运行求解器，为每个变量指定一个分数：每次运行都强制该变量作为首个变量，其余变量的顺序则随机排列，然后对该变量在这些运行中的传播次数取平均值（与第3节中的实验类似）。标签是按这些分数排序后的变量列表。这种标注方法基于每个变量的独立贡献；理想情况下，图神经网络容易学会这种方法，因为它试图隔离每个变量的独立贡献，同时消除其余变量随机排序带来的噪声。

4.3 遗传标注

遗传标注则将标注视为一个优化问题：寻找能够使传播次数最少的变量顺序。从 k 个变量的随机排列开始，我们保留最佳排列，并通过随机交换排列中的变量来生成新的候选排列，以爬山法的方式在 m 代中迭代改进该排列。

4.4 在求解器中使用预测的分支顺序

利用上述标注策略，我们的最终目标是标注大量训练实例，并学习如何在推理时间为新问题预测良好的变量分支顺序。然而，大多数 CDCL 求解器没有内置接口来提供建议的初始分支顺序。因此，为了开展实验，一个关键的设计决策在于如何将我们预测的顺序传递给求解器。求解器通常会为每个变量保留一个“活跃度”，并且通常会在发现变量参与冲突时更新该活跃度。我们发现，这些活跃度是以某种顺序初始化变量的一种简单方式。这一观察使我们能够修改这里使用的两个 SAT 求解器：MiniSat（Sorensson 和 Een，2005年）和 CaDiCaL（Biere 等人，2024年）。在这两个求解器中，我们通过初始化 VSIDS（Moskewicz 等人，2001年）启发式方法所使用的活跃度，将初始顺序无缝整合到求解器中；该启发式方法决定对哪个变量进行分支。对于 CaDiCaL，VMTF（Ryan，2004年）队列也会按照建议顺序进行初始化。

5 预测分支顺序

在本节中，我们研究图神经网络能够在多大程度上学习第4节中提出的分支顺序启发式，以及这些预测能够在多大程度上加快各类SAT随机、伪工业和工业实例的总体求解时间。

5.1 图表示

在使用图神经网络之前，我们需要将 SAT 实例转换为图（见图 1a）。SAT 实例可以按照图 3 中的示例转换为图。Yolcu & Poczos（2019）将 SAT 实例表示为三部图，其中变量（x_i）和子句（c_i）表示为节点，每条边表示某个变量出现在某个子句中。每条边还带有权重，用于表示变量在子句中是否以否定形式出现（实线表示否定，虚线表示未否定）。我们还添加了一个与每个子句节点相连的元节点（m），以加快消息传递，这与 Wang 等人（ 2024 ）的做法相同。最后，我们通过为每个节点添加独热编码特征，标明该节点的度数相较于所有其他同类型节点所属的四分位区间，从而增强各节点的特征；这种做法能略微提升性能。

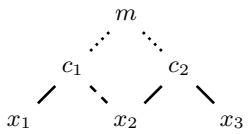


图3：.....的图 $(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3)$

5.2 图神经网络与损失函数

我们使用一个 GNN f_θ，其输入是 $\boldsymbol{x}$ 以图表示的 SAT 实例 $\boldsymbol{s}$，输出为 $= f_\theta(\boldsymbol{x})$，其中包含 SAT 实例中每个变量的得分。随后，我们根据变量的预测得分进行排序，从而得到预测顺序。我们的实现采用 NeuroBack（Wang 等人，2024）中的 GNN 架构，因为该架构在保持线性内存复杂度的同时支持注意力机制。

由于我们希望图神经网络能够预测最优顺序，因此将其建模为一个排序问题，目标是使变量在预测顺序中的排名与标注训练实例时生成的参考顺序中的排名一致。为此，我们可以借鉴信息检索领域的工具，具体来说，是学习排序中使用的损失函数（Liu，2007年）。我们采用 LambdaRank（Burges等人，2005年），因为它强调将高相关性 的变量正确排在预测顺序的前列。对于变量 x_i，我们将其相关性定义为 $rel_{x_i} = 1/log_2(R(x_i))$，其中 $R(x_i)$表示 x_i 在相应 SAT 实例的真实顺序中的排名。这一相关性定义体现了我们的优先级：最重要的变量会被赋予非常高的相关性，而排名靠后的变量则不那么重要。这也符合 SAT 求解器的树状搜索特性：最初的决策对性能有着不成比例的巨大影响。

5.3 评估

为了评估我们的方法，我们训练分支预测图神经网络，并使用 MiniSat（Sorensson 和 Een，2005年）和 CaDiCaL（Biere 等人，2024年）对其进行评估。我们在三种不同类型的分布上进行训练和评估：随机 3-CNF SAT、G4SATBench，以及 2015年至2024年的 SAT 竞赛数据。

训练和评估时，我们使用传播次数作为性能指标，因为它与求解时间成正比，不受系统影响，并且对小型实例而言更加可靠。我们在 CPU 上进行标注和评估，在 GPU 上进行训练；具体设置和耗时见附录 B。在 CPU 上运行图神经网络还会产生额外开销。对于变量数超过 150 的实例，与求解时间相比，这部分开销可以忽略不计；否则则影响显著。推理时间与运行时间的比较见附录 C，所用超参数见附录 D。

首先，我们研究图神经网络增强求解器在随机3-CNF SAT实例上的性能。我们使用 4.5 的变量-子句比生成实例；这一比值对应相变阈值，此时 SAT 和 UNSAT 实例出现的频率大致相同，且求解难度最高（Crawford & Auton, 1996）。这些样本在变量数 20 至 40 之间均匀分布，且 SAT 和 UNSAT 实例数量相同。我们选择使用较小的实例进行训练，因为第 3 节的结果表明，较小实例的标签方差较低，因此更容易学习。我们使用 9000 个实例进行训练，并使用 1000 个实例进行验证。测试时，我们使用完整的 Uniform Random-3-SAT SATLIB基准套件（Hoos & St¨utzle, 2000年），其中包含与训练数据相似的问题，变量数范围为 20 至 250。对于每个变量数，都有 100 个 SAT 实例和 100 个 UNSAT实例（变量数为 50 和 100 时各有 1000 个，变量数为 20 时则没有 UNSAT 实例）。我们针对每种标注方法和求解器分别训练一个图神经网络，因此共训练了六个不同的图神经网络。

图 4 展示了按变量ID顺序分支时，相对于默认顺序的求解时间降幅。我们看到，对于包含 200 个或更少变量的实例，使用图神经网络增强求解器可显著提升性能，求解时间有时可减少 50% 以上。在包含 20 至 40 个变量的实例上训练的网络，也能泛化到规模大得多的实例；对于比训练实例大 5 至 10 倍的实例，CaDiCaL 的求解速度也有明显加快。不过，对于更大的实例，性能收益会迅速下降。

有趣的是，这三种标注方法的性能相近，其中冲突标注在 SAT 实例上的表现略差。然而，这三种标注方法的可学习性差异很大：使用 CaDiCaL 时，冲突标注中预测顺序与真实顺序的平均斯皮尔曼相关系数为 0.37，而首个变量和遗传变量的对应数值分别为 0.33 和 0.062。这表明，图神经网络正在学习这些标注方法共有的某种底层特征，而建议顺序本身，尤其是遗传标注得出的顺序，很难预测。增加数据规模或训练时间似乎也不会改变这一点，这进一步表明模型估计的是某种底层特征。

其次，我们研究了 G4SATBench（Li 等人，2024）提供的测试套件上的性能，该套件包含随机问题、组合问题和伪工业问题。因此，它处于随机3-CNF SAT 与工业实例之间，适合作为两者之间的折中。我们在该套件中每个问题类别和可满足性类别下的 1000 个“中等”难度实例上进行训练，得到包含 14 000 个问题的训练集。随后，我们在该套件中每个问题类别和可满足性类别下的 100 个“困难”难度实例上进行评估，共得到 1400 个测试实例。

图 5 展示了 CaDiCaL 在 G4SATBench 上的结果。为简洁起见，我们将 MiniSat 的结果放在附录 E 中，因为不同方法之间没有明显差异。我们观察到，与随机实例类似，对于较容易的实例，传播次数减少幅度较大（首个变量标注和遗传标注可减少 10%–20%）。然而，对于需要超过 10^5 次传播的较难实例，我们没有发现图神经网络与基线方法之间存在明显差异。对于这些问题，各标注方法的预测能力差距甚至更大。冲突标注、首个变量标注和遗传标注的斯皮尔曼相关系数分别为 0.36、0.15 和 0.006。这可以解释为变量数量较多，导致首个变量标注和遗传标注产生的标签噪声更大。不过，我们发现，使用首个变量标签和遗传标签进行训练，性能仍优于使用冲突标签进行训练。

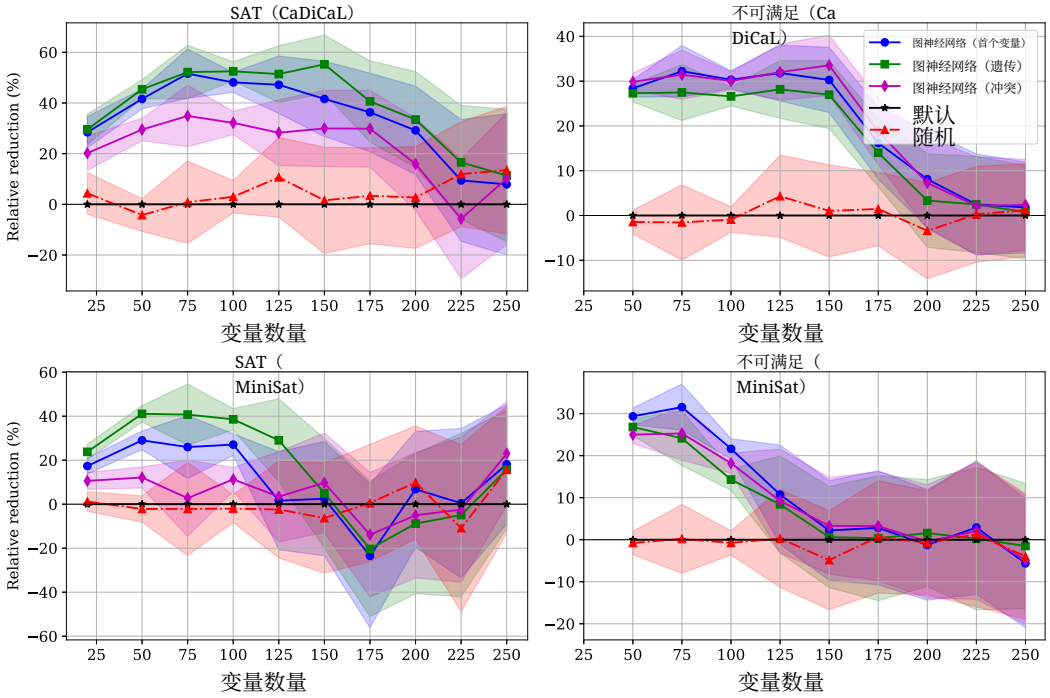


图 4：相对于默认变量顺序，在 SATLIB 均匀随机3-CNF基准测试上传播次数的平均减少量。上方：CaDiCaL 分别在 SAT 和 UNSAT实例上。下方：MiniSat 分别在 SAT 和 UNSAT实例上。阴影区域表示整个数据集的95%置信区间。

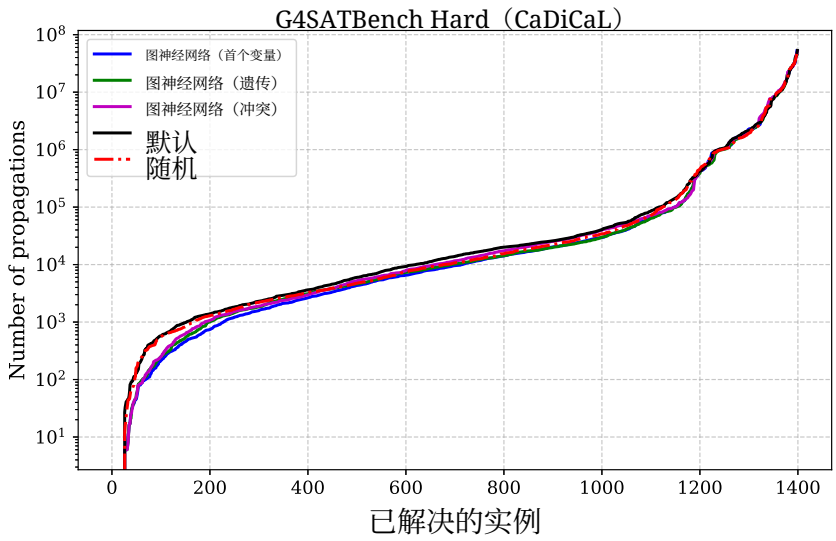


图 5：困难 G4SATBench实例上的传播次数。在大约 10^5 次传播之前，加速比为 10%-20%。

最后，我们研究了模型在工业实例上的性能。我们先使用 DataBack 数据集（Wang 等人，2024）中的实例进行训练，该数据集包含各种以生成实例为主的实例；随后在 2015 至 2023 年 SAT 竞赛主赛道的数据上进行微调。然后，我们在 2024 年 SAT 竞赛主赛道上评估性能。由于 DataBack 和 SAT 竞赛中的实例难度显著更高，使用首个变量标注和遗传标注在计算上不可行（因为标注时需要对每个实例求解多次）。因此

如图6所示，使用冲突标注训练单个图神经网络。与基线方法相比，我们没有观察到显著差异。

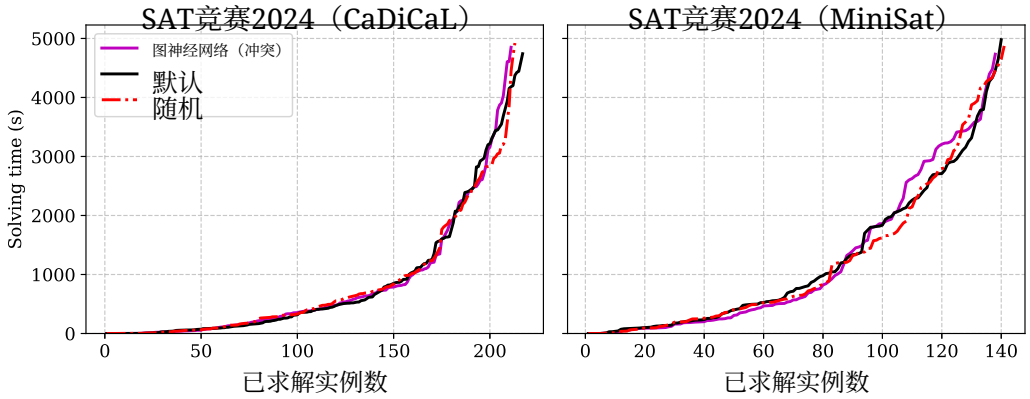


图6：CaDiCaL 和 MiniSat 在 SAT竞赛2024主赛道上的求解时间。

我们推测，在所有实验中，较难实例上的表现平平有两个原因。首先，我们认为，较长的求解时间会使求解器自身的启发式逐渐覆盖图神经网络建议的顺序，从而抵消其作用。这也可以解释 MiniSat 表现较差的原因：我们建议的顺序仅通过 VSIDS活动度施加，而由于缺少 CaDiCaL 中的 VMTF队列，这些活动度很快就会被覆盖。其次，我们认为，这些问题的复杂性使图神经网络更难可靠地预测出良好的初始顺序。为探究前一种原因，我们开展了一项初步实验，让求解器重新参考附录F中所示的建议顺序。

6 结论与未来工作

我们研究了将预测初始分支顺序作为 SAT 求解器的预处理步骤。实验表明，通过预测良好的初始分支顺序可以提升求解性能。此外，我们提出了三种为此类分支顺序生成标签的方法，并表明将这些标签用于图神经网络时，三种方法的效果相近。最后，我们表明，使用此类标签训练的图神经网络能够提升最先进模型在生成实例和伪工业实例上的性能，不过随着变量数量增加，性能提升幅度会下降。这有力地表明，图神经网络能够利用 SAT 问题的结构来改善最先进求解器的初始分支顺序；然而，随着问题越来越难以求解，其作用似乎会逐渐减弱。

7 致谢

相关计算得以使用瑞典国家超级计算学术基础设施（NAISS）提供的资源完成，该基础设施部分由瑞典研究理事会通过编号为 2022-06725 的资助协议提供资金支持。本研究还部分得到了瑞典研究理事会杰出教授资助（2017-01078）、克努特与爱丽丝·瓦伦堡基金会 Wallenberg Scholar 资助，以及瑞典战略研究基金会 FUSS SUCCESS资助的支持。

参考文献

Armin Biere、Hans van Maaren 和 Toby Walsh。可满足性手册。SAGE出版社，2009年1月。ISBN 978-1-60750-376-7。Google 图书 ID：shLvAgAAQBAJ。

Armin Biere、Tobias Faller、Katalin Fazekas、Mathias Fleury、Nils Froleyks 和 Florian Pollitt。CaDiCaL 2.0。载于 Arie Gurfinkel 和 Vijay Ganesh（编），计算机辅助验证，第133–152页，沙姆，2024年。施普林格·自然瑞士。ISBN 978-3-031-65627-9。doi：10.1007/978-3-031-65627-9 7。

Chris Burges、Tal Shaked、Erin Renshaw、Ari Lazier、Matt Deeds、Nicole Hamilton 和 Greg Hullender。使用梯度下降学习排序。载于第*22*届国际机器学习大会论文集 - *ICML '05*，第89–96页，德国波恩，2005年。ACM出版社。ISBN 978-1-59593-180-1。doi：10.1145/1102351.1102363。URL http://portal.acm. org/citation.cfm?doid=1102351.1102363。

Wenjing Chang 和 Wenlong Liu。SAT-GATv2：一种基于动态注意力的图神经网络，用于求解布尔可满足性问题。*Electronics*，14(3)，2025年1月。ISSN 2079-9292。doi：10.3390/electronics14030423。URL https://www.mdpi.com/2079-9292/14/3/423。公司：多学科数字出版研究所 分销商：多学科数字出版研究所 机构：多学科数字出版研究所 标签：多学科数字出版研究所。

James M. Crawford 和 Larry D. Auton。随机3-SAT中交叉点的实验结果。人工智能，81(1)：31–57，1996年3月。ISSN 0004-3702。doi：10.1016/0004-3702(95)00046-1。URLhttps://www.sciencedirect.com/science/article/pii/0004370295000461。

Holger H Hoos 和 Thomas St¨utzle。《SATLIB：用于 SAT 研究的在线资源》。*Sat*，2000：283–292，2000年。

Vitaly Kurin、Saad Godil、Shimon Whiteson 和 Bryan Catanzaro。图网络上的 Q学习能否学习出适用于 SAT 求解器的通用分支启发式？，2020年11月。网址：http://arxiv.org/abs/1909.11830。arXiv:1909.11830 [cs]。

Zhaoyu Li、Jinpei Guo 和 Xujie Si。G4SATBench：使用图神经网络对 SAT求解进行基准测试并推动其发展，2024年5月。网址：http://arxiv.org/abs/2309.16941。arXiv:2309.16941 [cs]。

Tie-Yan Liu。《面向信息检索的学习排序》。《信息检索基础与趋势》，3(3)：225–331，2007年。ISSN 1554-0669，1554-0677。doi：10.1561/1500000016。网址：http://www.nowpublishers.com/article/Details/INR-016。

Ilya Loshchilov 和 Frank Hutter。《解耦权重衰减正则化》。2018年9月。网址：https://openreview.net/forum?id=Bkg6RiCqY7。

Matthew W. Moskewicz、Conor F. Madigan、Ying Zhao、Lintao Zhang 和 Sharad Malik。《Chaff：构建高效的 SAT 求解器》。载于第*38*届年度设计自动化会议论文集，DAC '01，第530–535页，纽约，美国纽约州，2001年6月。计算机协会。ISBN 978-1-58113-297-7。doi：10.1145/378239.379017。网址：https: //dl.acm.org/doi/10.1145/378239.379017。

Jakob Nordstr¨om。论证明复杂性与 SAT求解之间的相互作用。*ACM SIGLOGNews*，2(3):19–44，2015年8月。ISSN 2372-3491。doi：10.1145/2815493.2815497。网址：https://dl.acm.org/doi/10.1145/2815493.2815497。

Lawrence Ryan. 子句学习SAT求解器的高效算法。2004年。

Daniel Selsam和Nikolaj Bjørner. 利用不可满足核预测引导高性能SAT求解器，2019年7月。URL http://arxiv.org/abs/1903.04671. arXiv:1903.04671 [cs].

Daniel Selsam、Matthew Lamm、Benedikt B¨unz、Percy Liang、Leonardo de Moura和 David L Dill. 从单比特监督中学习SAT求解器。*arXiv*预印本 *arXiv:1802.03685*，2018年。

Zhengyuan Shi、Min Li、Yi Liu、Sadaf Khan、Junhua Huang、Hui-Ling Zhen、Mingxuan Yuan和Qiang Xu. SATformer：基于Transformer的不可满足核学习。载于*2023*年*IEEE/ACM*国际计算机辅助设计会议（*ICCAD*），第1–4页，2023年10月。doi: 10.1109/ICCAD57390.2023.10323731. URL http://arxiv.org/abs/2209.00953. arXiv:2209.00953 [cs].

Niklas Sorensson 和 Niklas Een。MiniSat v1. 13-a：一种带冲突子句最小化功能的 SAT 求解器。*SAT*，2005(53):1–2，2005年。

Wenxi Wang、Yang Hu、Mohit Tiwari、Sarfraz Khurshid、Kenneth McMillan 和 Risto Miikku-lainen。NeuroBack：使用图神经网络改进 CDCL SAT 求解，2024年5月。网址：http://arxiv.org/abs/2110.14053。arXiv:2110.14053 [cs]。

Ryan Williams、Carla P. Gomes 和 Bart Selman。典型情形复杂度的后门研究。发表于国际人工智能联合会议，第 3 卷，第 1173–1178 页，2003 年。网址：https://www.researchgate.net/profile/Richard-Williams-54/publication/2871393_Backdoors_To_Typical_Case_Complexity/links/0c96051c562040e5f8000000/ Backdoors-To-Typical-Case-Complexity.pdf。

Emre Yolcu 和 Barnabas Poczos。学习布尔可满足性的局部搜索启发式。发表于神经信息处理系统进展，第 32 卷。Curran Associates, Inc.，2019 年。网址：https://proceedings.neurips.cc/paper/2019/hash/12e59a33dea1bf0630f46edfe13d6ea2-Abstract.html。

Shumao Zhai 和 Ning Ge。在分治 SAT 求解器中通过强化学习学习分裂启发式。载于第十三届国际学习表征会议，2025年。

Wenjie Zhang、Zeyu Sun、Qihao Zhu、Ge Li、Shaowei Cai、Yingfei Xiong 和 Lu Zhang。NLo-calSAT：通过解预测提升局部搜索。载于第二十九届国际人工智能联合会议论文集，第 1177–1183 页，2020年7月。doi：10. 24963/ijcai.2020/164。网址：http://arxiv.org/abs/2001.09398。arXiv：2001.09398[cs]。

标记伪代码

以下伪代码尽可能详细地描述了标记算法，以便据此实现。对于遗传标注，我们还会逐代减少随机交换的幅度，使其距离每代减半。

算法 1：首变量标注

输入：SAT实例 ϕ，求解器 f_{solve}，每个变量的试验次数 k
输出：变量顺序 π_{label}
$X \leftarrow$变量位于 ϕ 中;
对于每个 $x_i \in X$ **执行**
  评分[x_i] $\leftarrow 0$;
  对于 $j \leftarrow 1$**至** k **执行**
    $\pi_j \sim \mathcal{U}$（ $X \setminus \{x_i\}$的所有排列）；
    排序 $\leftarrow$ （$x_i \,||\, \pi_j$）；
    分数[x_i] $\leftarrow$ 分数[x_i] $+ f_{solve}(\phi,$ 顺序);
  分数[x_i] $\leftarrow$分数[x_i] $/k$；
$\pi_{label} \leftarrow$ X 中的变量按分数排序;
返回 π_{label}；

算法 2： 遗传标注

输入： SAT 实例 ϕ，求解器 f_{solve}，种群规模 k，代数 m
输出： 变量顺序 π_{label}

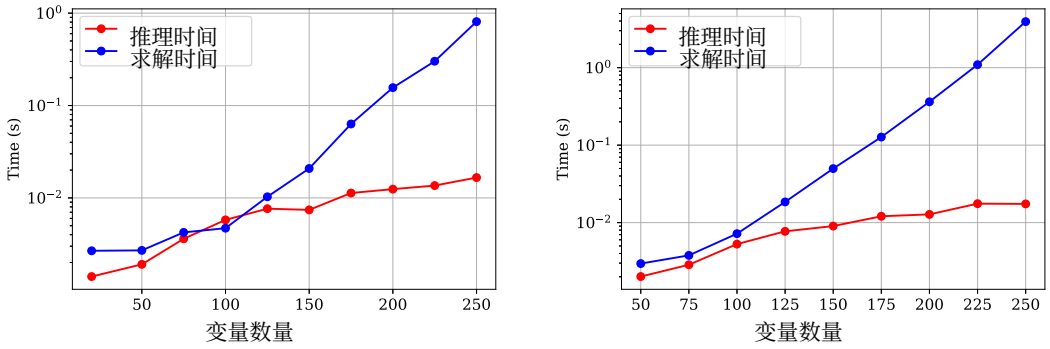
$X \leftarrow$变量，位于 ϕ 中;
对于 $j \leftarrow 1$**到** k，**执行**
  $\pi_j \sim \mathcal{U}$（X的所有排列）;
  分数$[\pi_j] \leftarrow f_{solve}(\phi, \pi_j)$;
$\pi_{best} \leftarrow$ 使分数$[\pi_j]$ 最小的参数;
对于 $g \leftarrow 1$**到** m，**执行**
  $l \leftarrow |\pi_{best}|/2$;
  对于 $j \leftarrow 1$ **到** k，**执行**
    $\pi'_j \leftarrow$对 π_{best} 执行 $|\pi_{best}|$次随机交换，交换长度最大为 l;
    得分$[\pi'_j] \leftarrow f_{solve}(\phi, \pi'_j)$;
  $\pi_{best} \leftarrow \arg\min^{\pi'_j}$ score$[\pi'_j]$;
  $l \leftarrow l/2$;
return π_{best} 作为 π_{label} ;

B 计算设置

训练图神经网络时，实验在单块 NVIDIA A100 GPU 上进行；进行标记和测试时，则使用搭载 AMD EPYC 7742 64 核处理器的 CPU。通过求解每个实例 100 次来标记 1 万个实例，使用全部 64 个核心耗时不到一分钟。训练 10 个训练轮次也不到一分钟。

C 推理时间

为了公平比较，我们在图 7 中将图神经网络在 CPU 上的推理时间与 CaDiCaL 的求解时间进行比较，以了解对于哪些随机 3-CNF 实例而言该成本可以忽略不计。正如预期，我们看到推理时间呈现线性模式，而求解时间呈现指数模式。将其与图 4 中的结果相结合可以看出，在大约 150 个变量之后，推理所增加的成本是值得的。在 GPU 上进行推理时则无需同样的顾虑，因为这要快得多。



(a) SAT 实例　　(b) UNSAT实例

图 7：比较仅使用 CPU 求解 SAT（a）和 UNSAT（b）实例时，使用 CaDiCaL（默认顺序）的推理时间与求解时间。请注意，对数纵轴。

D 模型超参数

在训练过程中，我们使用表 2 所示的超参数。

表 2：模型超参数

超参数	**值**
模型架构	图 来自 NeuroBack（Wang 等人，2024）
优化器	AdamW（Loshchilov 和 Hutter，2018）
Adam β_1	0.9
Adam β_2	0.999
学习率	0.001
训练轮数	10
批量大小	128
GSA 模块	3
LSA 模块	3
Dropout 率	0.2
权重衰减	0.01
梯度裁剪	1.0
损失函数	LambdaRank

E MINISATON G4SATBENCH

参见图8。

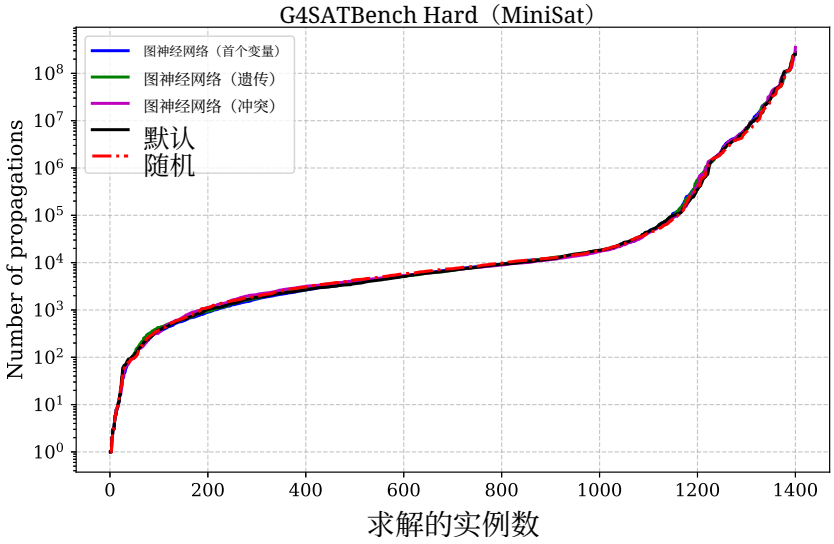


图8：困难 G4SATBench实例的传播次数。没有明显的加速。

F 关于建议顺序的提醒

我们研究了，在求解耗时较长的实例时，提醒求解器采用建议的分支顺序是否有助于提升性能。具体做法是，每次重启时（大约每 1000 次冲突）将建议顺序加入 VSIDS活动度，方法是更新变量 i through

$$activities(i) \leftarrow remind\ factor \times max(activities) \times decay\ factor^{-R_{GNN}(i)}$$

where 活动度是 VSIDS活动度，衰减因子为 0.95。 $R_{GNN}(i)$ 是 t 的排名

he varia
按建议顺序排列。

我们使用 MiniSat（Sorensson 和 Een，2005年）在2015年至2024年的SAT 竞赛上对此进行评估，超时时间为5000，其中建议顺序是不同提醒因子对应的冲突标签。图 9

显示了结果。对于较大的实例，我们发现不进行提醒是最快的选项，从而表明每次重启都进行提醒的朴素方案似乎并不能缓解我们无法加速难以求解实例的问题。提醒表现不佳也可以归因于我们不断覆盖关键的VSIDS活动度，从而使求解器更难妥善探索问题。

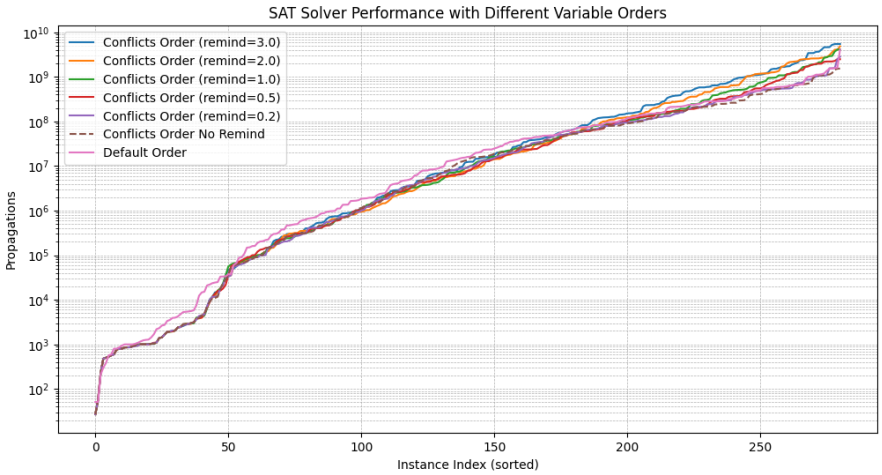


图 9：不同提醒因子的传播次数。仅绘制了七种初始化方式均能求解的问题。