PROPOSAL 02

有限预算下的局部形式化：AI–PDE 研究证据的选择、核验与复验

Budgeted Local Formal Verification for AI-Assisted PDE Research

核心主张：联合选择核验位置与数学粒度，比一律整段形式化、固定细粒度核验或只挑容易步骤，更有效地控制研究主张中的错配，并保留有效进展。
明确边界：局部化服务于人为/AI 操作的可控性与独立复验，不表示 Lean 内核对长证明不可靠。
状态：方法设计与实验计划；未报告尚未执行的实验或未完成的新定理。

摘要

形式化证明通过核验，确认的是给定形式化命题的逻辑成立，而不自动确认该命题与 PDE 论文中的目标及调用方式一致。研究级解析证明还涉及难以整体形式化的外部估计、变化的参数范围和持续修订的证据。本文研究有限核验预算下的局部形式化：如何选择需要检查的数学位置与粒度，使独立研究者能够确认一个结果实际证明了什么、依赖什么，以及可用于哪些后续结论。我们提出一种面向证明依赖与调用接口的核验调度方法，将完整数学责任组织为局部单元，并共同考虑形式化成本、语义审查成本、复验成本及对最终主张的影响。每个单元绑定正文陈述、固定 Lean 目标、允许的外部输入、定义和版本；证明在独立环境中核验，未闭合输入随证据导出，修订后只保留仍被有效证明支持的调用。我们设计包含正确证明、自然产生的错配和明确标记的受控变体的评估任务，检验参数量词、函数空间、比较对象及失效依赖的处理，并比较不同核验策略的有效证据保留、错误主张接受与专家成本。研究重点不是再次展示 Lean 能核验局部证明，而是建立可测量的资源分配机制，使部分形式化在真实 PDE 研究中产生明确、可独立复验且不越界的证据。

为什么值得独立成篇

若只实现若干 Lean 引理和一份检查清单，这篇不足以独立支撑方法论文。需要交付**核验位置与粒度的选择算法**，证明它相对合理基线具有可重复优势，并揭示什么时候细分有益、什么时候接口增多反而增加审查负担。

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1. 最接近工作与本篇的独立问题

承认工具与形式化基础已存在；把新意限定在预算、接口和研究修订。

已有工作	已提供的能力	本文拟检验的额外问题
Lean 官方核验机制	固定目标、证明项及依赖的检查；独立重放工具。[1, 2]	谁选择目标，哪些局部最值得核验，有限预算怎样分配。
ProofFlow, ICLR 2026	依赖图与中间引理的忠实形式化。[3]	部分形式化时的接口覆盖、成本与动态修订。
Pistis, 2026 预印本	对自然语言证明忠实性的条件与形式化搜索。[4]	非全形式化 PDE 证据中，局部单元与外部解析输入的管理。
APE-Bench I	真实形式化库中的修改、重构及修复任务。[5]	自然语言研究稿与局部代码之间的对应及外部复验。

三个具体研究问题

Q1：选择哪里。是否应优先核验影响多个后续关键结论的接口，而不是优先核验容易写进 Lean 的局部代数？图上的度数不等于数学重要性，存在替代证明时尤其如此。

Q2：切成多大。按一行、一条引理、一个闭合模块切分，会如何改变形式化时间、接口审查量、定位能力和独立重放成本？本项目不先验断言“越小越安全”。

Q3：修订之后。一个局部证据通过后，改变比较对象、参数范围或上游引理时，哪些调用仍有效？一个证明分支失效并不表示命题为假；另一条完整证明可能继续支持它。

两个不能冒充的新意

从若干有效蕴含组合出条件性结论，是应满足的逻辑正确性性质，不是值得单独包装的重大理论发现。检查 `sorry`、固定软件版本、使用独立核验器，是必要工程基础，也不能单独承担论文创新。[1, 2]

独立性检验：将论文一的研究生成器换成普通 LLM 或人工证明稿，本篇算法仍应可运行；将论文三的重构器完全移除，本篇主实验仍应成立。

2. 核验流程与责任边界

预算用尽时输出证据的实际支持范围，不把未检查部分判为通过。

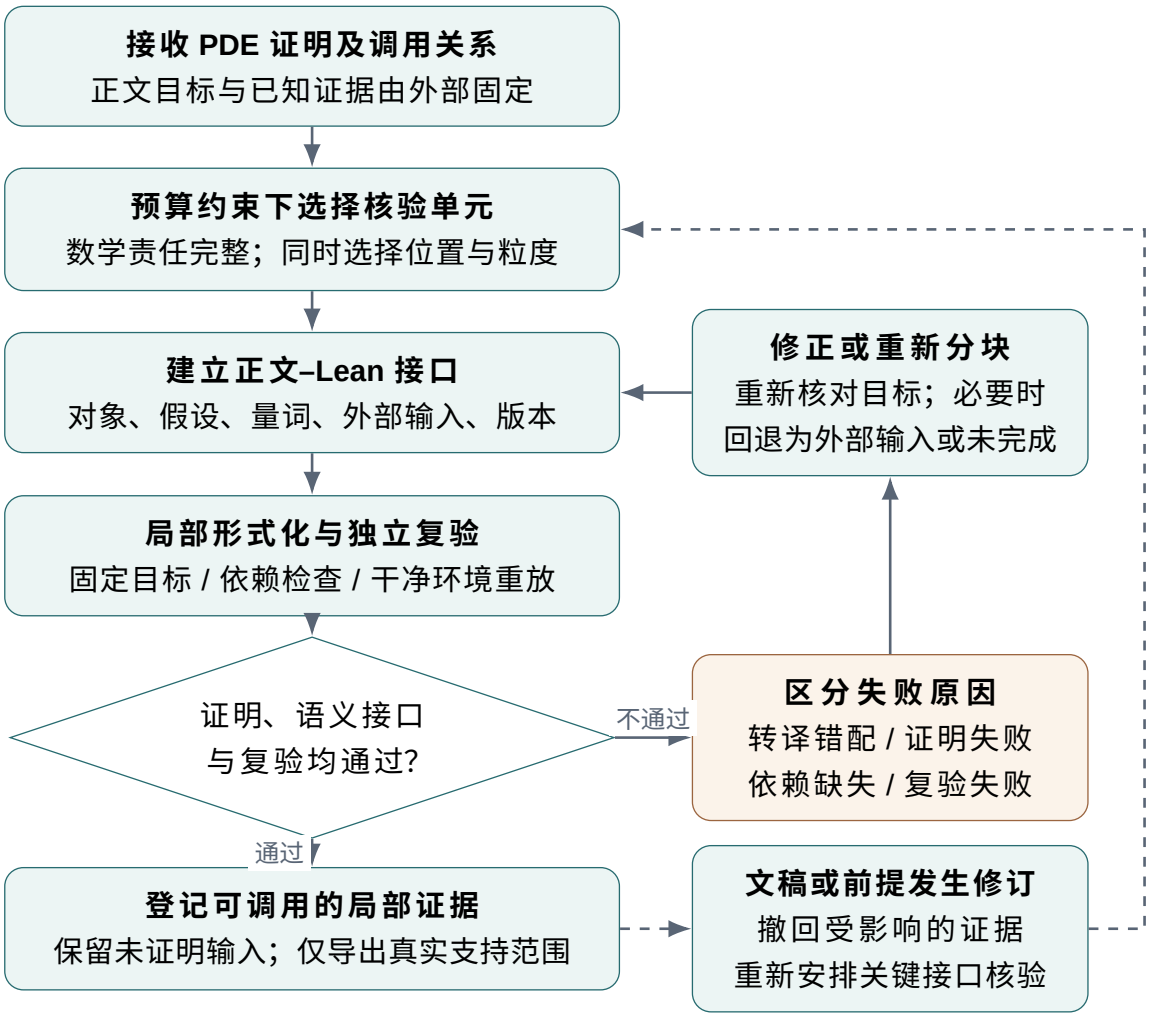

图 2：每个局部单元的形式化目标与定义由可信输入约束。核验调度器不能自行加强假设来提高通过率；正文到形式化目标的语义对应不由一次编译成功自动解决。

3. 核验单元的定义与可执行的语义接口

局部按数学责任划分；小代码加大量隐藏前提，不是更好的局部核验。

3.1 单元必须包含的内容

一个单元包含正文位置与命题版本，Lean 中的明确目标，变量及对象对应，允许使用的前提及其来源，常数的依赖与量词范围，证明文件、环境锁定及重放命令。元数据为研究系统服务，不要求正文增加大量新符号。

三**种证据状态分开：**完整形式化证明；相对于明确外部输入成立的形式化推导；仅经解析审查的非形式化证明。尚未证明的片段保持未完成。不能将后三者统称为“已被 Lean 证明”。

3.2 一个极易被正确代码掩盖的错配

对非负量 a_ε，下面是不同命题：

$$\forall \varepsilon \in (0,1],\ \exists C_\varepsilon > 0:\ a_\varepsilon \le C_\varepsilon, \qquad \exists C > 0,\ \forall \varepsilon \in (0,1]:\ a_\varepsilon \le C.$$

取 $a_\varepsilon = \varepsilon^{-1}$，前者成立，后者不成立。若论文需要一致估计，而代码准确证明前者，内核没有犯错；错误发生在转译或调用接口。

因此，量词顺序、允许常数依赖及参数域必须出现在固定目标中。**若可信目标文件本身翻译错了，再多次内核重放也不能发现“原文其实想说什么”**。该目标需由独立语义审查确定。Lean 官方文档同样保留对目标及其定义的检查责任。[1]

3.3 只导出实际证明的命题

设原假设为 H，未闭合外部输入为 $U_1, \dots, U_k$。局部连接均有有效证据且组合正确时，允许导出的范围是

$$(H \wedge U_1 \wedge \cdots \wedge U_k) \Longrightarrow T.$$

仅当这些输入在 H 下已有证据，才可导出 $H \Rightarrow T$。若上式的组合本身也缺失，则只能标记为未完成，不能靠列出假设称为“条件定理”。

3.4 循环依赖怎么处理

普通依赖图只组织证据。先验估计中的循环必须封装成一个**带进入条件与退出结论的闭合模块**，内部给出连续性、初始小性、吸收、改进与延拓所需步骤。不能因为图中把循环压成一个节点，就宣称完成闭合。

4. 方法核心：联合选择核验位置与粒度

需要一个真正可比较的决策规则，而不是把“关键步骤优先”写成口号。

4.1 候选集合

对已提取的证明依赖，为同一数学责任产生不同候选：单条估计、估计与适用条件的组合、包含系数选择的小闭合模块。每个候选显式记录其输入边界与输出边界，避免小单元通过把困难内容搬进假设而获利。

4.2 决策规则

先在独立开发集测量各类候选的形式化时间、语义审查时间、重放时间和实际发现错配的情况。下一步按

$$\text{优先级}(j) = \frac{\widehat{\Delta}_j}{\widehat{c}_{\text{formal}}(j) + \widehat{c}_{\text{interface}}(j) + \widehat{c}_{\text{replay}}(j)}$$

选择候选，其中 $\widehat{\Delta}_j$ 是基于当前证据状态估计的**对预登记目标的新增核查效益**，并非模型自报“我觉得危险”的置信度。效益估计综合接口变更类型、缺失支持、历史定位表现及下游目标影响；参数在开发阶段冻结。

这是一条可实现的启发式而非安全概率定理。若成本不可用共同单位换算，采用分别约束专家时间与机器预算的排序，而不任意折算。报告单纯关键性、随机选择及成本优先的对照，检验更复杂规则是否值得。

4.3 防止依赖图的虚假收益

一条已核验推导只有在其所有前提获得相应支持时才支撑后继；多个替代证明不重复计数。某接口影响许多节点，但若这些节点都有不依赖它的完整证明，则不应赋予同等高优先级。状态更新针对证明证据，而不只针对命题名称。

图与接口提取也会出错，因此以原始文本与局部证明项反查依赖。提取质量作为独立诊断项，而非默认正确。随机审查少量低优先级接口，防止调度器永久忽略未知失效类型。

4.4 预算与终止

预算不足时不再扩张形式化范围，输出已核验单元、外部输入、开放义务和重放日志。某单元超时，允许重新切分，但原来的失败成本仍计入。仅在实际完成检查后更新状态，不能把“计划将由 Lean 检查”记成通过。

> **理论边界：**不主张对任意证明图实现全局最优核验，不主张安全性由图覆盖率保证。可证明的范围仅是明确前提下的证据组合与状态传播正确性；真正的研究贡献需由资源—有效性对比建立。

5. 实验设计：不把简单语法错误当成研究级成果

主任务应包含“代码合法，但正文或调用对象不匹配”的困难情况。

5.1 三个任务簇

自然记录。从研究修订中提取真实发生的假设漂移、误差对象变化、参数一致性遗漏；必须能定位原文、修订及专家判断。无法定位的事件不能写成历史事实。

受控对照。围绕已确认有效的局部证明，改变一个接口因素：量词、函数空间、定义归一化、比较剖面或来源版本。错误与仍然正确的变体混合出现；实验结果明确标注为受控干预，不能冒称模型自然错误频率。

动态任务。先提供有效局部证据，再改变一个上游条件；要求重新导出定理。评价是否准确撤回受影响调用，并保留另一条未失效的证明分支。

5.2 拟定规模

先导：6 个母模块，用于确定切分粒度和审查成本。主实验：12 个新的母模块，各含 2 个目标接口；每接口保留 1 个正确版本与 4 个受控变体，共 120 个任务实例。**统计独立单位仍是 12 个母模块，不是 120 个独立研究问题**。另外收集自然错误轨迹，单独报告。

局部对象覆盖：能量等价、Young 参数组合、时间权重、Sobolev/插值的适用条件、退化参数一致性及上下界的对象匹配。真正可形式化的解析引理与仅形式化逻辑组合的任务分层统计，不混合宣称形式化深度。

5.3 所有方法共享的核验底线

均使用固定可信目标、相同 Lean/Mathlib 环境、同一允许公理清单和最终独立检查。必须检查传递依赖，并明确外部假设；不能仅检查当前文件是否出现 `sorry`。[2]

使用干净环境、固定源码和工具链重放；可用官方验证工具对目标匹配及导出的证明项进行独立检查。[1] 不通过故意削弱基线的核验安全性，来制造本方法优势。

5.4 主基线

整模块核验；固定细粒度核验；基于依赖的分块方法；成本优先或随机局部选择；本方法的位置—粒度联合选择。全部方法获得相同机器预算、专家审查额度及正式检查器。ProofFlow/Pistis 只能在任务接口相符且可复现时作为直接基线；适配实现必须明确标注。[3, 4]

6. 指标、判定与实验结论边界

不把一律拒绝的系统评成最可靠，也不以代码行数衡量数学覆盖。

6.1 三项主结果

有效证据保留率：在确实有正确可用证据的任务中，系统保留并准确声明其支持范围的比例；拒绝全部任务会在此失分。

错误主张接受率：在专家预先确认存在特定错配的任务中，仍将相应原目标写成已建立的比例。错误形式化命题、错误调用和证据不足分别给出分母，不混用“假阳性”。

复验成本：第三方在干净环境中确认固定目标、核对声明范围并重放所需的专家时间与机器资源。成功者平均时间可能有选择偏差，因此必须同时报告未完成比例与预算内完成率。

6.2 研究性分析

绘制核验预算变化下的**有效保留—错误接受—成本**关系，不做任意加权总分。按粒度分层分析：单元变小是否降低定位成本，却增加接口审查？何时一整个闭合模块比分散小引理更经济？这是论文最值得形成的新认识。

位置选择、粒度选择、语义对齐、修订传播分别消融。所有版本采用同一最终内核检查，避免把工具差异误当作调度器优势。

6.3 必须避免的统计结论

受控错误上的拦截率不能外推成真实研究中错误发生的概率。零次错误接受不能证明错误率为零。对母模块作分组自助法或合适的层级分析；模型与专家背景分层，报告区间而非只列最好结果。

若高成本整体验证能达到更低错误率，而本方法更经济，应表述为**预算下的权衡改进**，不声称全面更安全。若只在合成量词变体上有效、自然研究错配上没有收益，则贡献范围应明确收缩。

6.4 小型逻辑保证

可形式化以下性质：在接口对应正确、输入证据有效、版本一致的条件下，已登记推导组合出的结论不超出其外部前提。证明可对无环证据组合归纳；先验循环须已由独立闭合模块处理。**该性质约束实现，不能证明自动语义判断永远正确。**

> **验收时应问：**同一笔审查预算，系统是否让更多正确研究证据可以被独立使用，同时没有以隐藏输入为代价？这比“生成了多少行 Lean”更有意义。

7. 实施路线与独立成篇的门槛

第二篇最容易退化为工程包装，必须通过下面的独立性检验。

最小实现

先完成 5 类接口的规范与固定目标，再实现单元抽取、候选切分、预算调度及版本化调用。每次核验产物包括正文片段、形式化命题、证明及依赖、输入状态和重放记录。将提取、形式化、核验与最终导出分开，防止一个模型既改目标又给自己验收。

机器预算记录模型调用、工具运行、重试与失败；人工预算记录语义对齐、解析审查和复验。开发标注成本单列，不隐藏在基线之外。正式测试期间不允许人工对本方法定向补关键证明而不给基线同等待遇。

三个里程碑

M1：正确处理“固定非零波强”与“对波强一致”的量词区别，并给出可复验的正确与错误使用对照。此阶段仅证明接口机制可工作。

M2：在包含替代证明和一次上游修订的模块中，正确更新受影响证据；比较不同粒度真实成本，不预设结论。

M3：在保留母模块与自然研究记录上，证明联合选择相对强基线的预算收益；完成独立第三方复验，公开可释放的任务与所有结果。

投稿门槛与停止条件

独立投稿至少需要：选择算法本身有实质贡献；与固定粒度、强依赖方法和同检查器基线比较；自然任务与受控任务均有证据；专家语义审查和复验成本实际测量。

若缺少上述增量，应将局部 Lean 基础设施并入第一篇或第三篇，而不是为了凑三篇单独投稿。这是研究组合的质量控制，不是预先否定第二篇的潜力。

九页主文安排

第 1 页问题与范围；第 2 页相关工作和局部证据模型；第 3–4 页选择算法、接口与修订；第 5 页一条完整案例；第 6 页任务与预算；第 7–8 页主要结果、粒度权衡及消融；第 9 页局限与结论。代码核验细节与独立复验说明放附录，但正文必须能辨明到底形式化了什么。

8. English abstract and references

不把“局部化”包装成内核可靠性层级；它优化的是形式化与审查过程。

English abstract

A checked formal proof establishes a formal statement, but does not by itself establish that the statement matches its intended use in a PDE manuscript. Research-level analysis also contains external analytic inputs, parameter-sensitive claims, and proofs that change during revision. We study budgeted local formal verification: choosing where to verify and at what mathematical granularity so that independent researchers can determine what a result proves, assumes, and supports. Our method schedules verification units using their interfaces, downstream relevance, and measured formalization, semantic-review, and replay costs. Each unit binds a manuscript claim to a fixed Lean target, explicit external inputs, definitions, and versions. Checked evidence retains its unresolved premises; revisions trigger revalidation of affected uses while preserving unaffected alternative proofs. We design evaluations combining valid controls, documented research mismatches, and labeled interventions involving quantifiers, function spaces, comparison objects, and stale dependencies. All methods share the same trusted targets and final proof-checking safeguards. The central evaluation compares valid evidence retained, incorrect claims accepted, and independent verification cost across budgets. The contribution is not the use of Lean for short proofs, but a testable allocation mechanism for making partial formalization useful, reproducible, and explicit about its support boundaries in mathematical research.

参考文献

[1] Lean Language Reference. *Validating a Lean Proof*. 官方文档，核查于 2026 年 9 月 18 日。

[2] Lean Language Reference. *Axioms*. 官方文档，核查于 2026 年 9 月 18 日。

[3] Cabral et al. *ProofFlow: A Dependency Graph Approach to Faithful Proof Autoformalization*. ICLR 2026. 正式论文集。

[4] Mao et al. *Does the Proof Prove It That Way? Faithful Formalization of Elements Proofs*. arXiv:2608.15432, 2026. 预印本。

[5] Xin et al. *APE-Bench I: Towards File-level Automated Proof Engineering of Formal Math Libraries*. arXiv:2504.19110, 2025. 论文。