神经符号合规：整合大语言模型与 SMT 求解器，实现自动化金融法律分析

Yung-Shen Hsia*、Fang Yu* 和 Jie-Hong Roland Jiang†

*国立政治大学管理信息系统系，台湾台北 电子邮箱：113356046@g.nccu.edu.tw,yuf@nccu.edu.tw

†国立台湾大学电机工程系，台湾台北 电子邮箱：jhjiang@ntu.edu.tw

arXiv:2601.06181v1 [cs.AI] 7 Jan 2026

摘要—**金融法规日益复杂，阻碍了自动化合规，尤其增加了在尽量减少人工监督的情况下维持逻辑一致性的难度。我们提出一种神经符号合规框架，将大语言模型（LLM）与可满足性模理论（SMT）求解器相结合，以实现**形式化可验证性**和**基于优化的合规纠正。**大语言模型解读法规和执法案件，并生成 SMT 约束；求解器则确保一致性，并在出现处罚时计算恢复合法性所需的**最小事实修改。**与注重透明性的方法不同，我们的方法强调由逻辑驱动的优化，提供可验证且符合法律规定的推理，而非事后解释。系统在来自台湾金融监督管理委员会（FSC）的 87 起执法案件上进行了评估，SMT 代码生成的正确率达到 86.2%，推理效率提升超过 100×，并能持续纠正违规行为，为基于优化的合规应用奠定了初步基础。**

索引词—**金融判例分析、检索增强生成（RAG）、大语言模型代理、SMT求解器、神经符号、法律科技、法律理解分析、约束求解、金融监督管理委员会（FSC）、优化、法律一致性、MaxSMT**

一、引言

随着法律法规日益复杂且变化愈发频繁，合规自动化已成为法律科技领域的一项关键挑战 [1], [2]。其中的核心难点在于，如何将法律条款转换为机器可读格式，同时保留其丰富的语义内涵 [3]。与此同时，人工智能加速了金融领域的转型，助力完成从异常检测到监管监测等任务 [4]–[6]。

SAT/SMT 求解和模型检测等形式化方法 [7], [8]近期已被用于法律推理，通过将法律和事实编码为约束来实现这一目标 [9]。大语言模型（LLMs）在理解法律文本方面表现出色 [10],[11] ，但在高风险任务中往往缺乏精确性。因此，有必要采用混合神经符号方法，将符号推理的可解释性与大语言模型的语言能力相结合 [12], [13]。

神经符号人工智能通过将基于逻辑的推理与深度学习相结合，实现了组合泛化和透明性。它已在视觉推理、机器人操作和概念落地等领域取得成功 [13]。这些能力在合规领域尤为重要，因为相关决策必须既可解释，又经得起法律审查。

近期将大语言模型与形式化推理系统相结合的研究，凸显了对可验证的人工智能推理日益增长的需求。例如，基于大语言模型的形式化编程（LLMFP）框架 [14]表明，将自然语言问题转化为可满足性模理论（SMT）公式，能够实现纯神经网络方法无法保证的完备性和可靠性。同样，Chen 等人 [15]指出，即使是双重逻辑链（DCoL）等先进的提示策略，也仍无法实现一致的逻辑推理，因为大语言模型本质上缺乏自验证机制。这些发现表明，尽管大语言模型擅长灵活地理解自然语言，符号求解器却能提供实现可证明正确性所需的严谨性。

基于这一认识，我们的工作将大语言模型与基于SMT的形式化验证相结合，以弥合可解释性与可验证性之间的差距，旨在提升监管合规任务中的透明性和逻辑一致性。

与这一趋势相呼应，关于基于大语言模型的智能体形式化验证 [16] 的研究进一步强调，应对大语言模型的推理过程进行形式化检验，并提出了多大语言模型辩论和自我纠正等策略，以提升一致性和可靠性。这一系列研究表明，更广泛的共识正在形成：将大语言模型与形式化方法相结合，对于开发可信且可验证的人工智能系统至关重要。

然而，很少有框架针对特定司法管辖区金融法律执法的独特特征。例如，台湾的监管生态系统涉及多语言文档、本地化法规以及详细的执法案件记录。这些因素带来的挑战，现有合规工具尚未充分应对。

在这项工作中，我们提出了一种结合大语言模型与基于可满足性模理论的验证方法的合规框架，专为

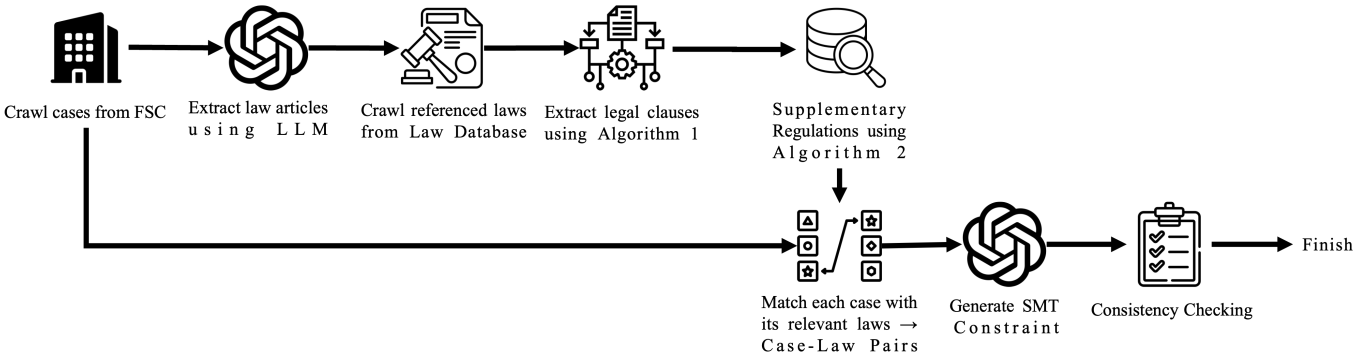


图 1：SMT约束生成流程

台湾的金融法律语境量身打造。RAG流程整合了法规、执法案件以及FSC的违规记录。法律规范被编码为约束，生成式模型则协助解读，以支持违规预测和法律指导。

我们在 87 起真实的 FSC 执法案件上评估了该系统，展示了其检测非法合同条款并生成符合规则的建议的能力。SMT层提升了可解释性和准确率，为更加透明、可验证的法律科技应用提供了潜在基础。

II. 相关工作

许多研究通过各种计算技术探索监管合规自动化。大语言模型在这一领域受到关注。在财务审计中，Berger 等人 [17] 评估了大语言模型用于自动化合规检查的效果，并比较了 LLaMA-2 和 GPT-4 等模型。GPT-4 在多语言任务中表现出色，而 LLaMA-2 在检测不合规方面更具成本效益。然而，结果波动和假阳性等挑战依然存在。同样，Liao等人 [18]提出了一种神经符号推理框架，其中大语言模型生成逻辑形式，再由符号求解器进行验证。Ning 等人 [19]进一步拓展了这一思路，整合多智能体规划和外部符号工具，以增强推理能力和可解释性。Shi等人 [20] 也提出了一种双系统神经符号架构，将启发式大语言模型推理与基于逻辑的验证相结合，从而提升准确率和可解释性。这些工作表明，语言模型与形式化方法在监管任务中的融合趋势日益显著。

SMT 求解器和符号方法提供了另一条研究路径。Judson等人 [21] 推出了*soid*，通过 KLEE 将 Z3 与符号执行相结合，为法律从业者提供了图形界面支持的形式化问责工具。Feng等人 [22]使用度量一阶时序逻辑（MFOTL）将法律约束转换为 SMT，从而能够在医疗保健和金融领域高效进行合规检查。类似地，Tsigkanos等人 [23] 应用 SMT 对经济网络建模，并通过逻辑约束确保一致性。

混合大语言模型–SMT系统旨在弥合自然语言与形式语言之间的鸿沟。语义自验证（SSV）框架 [24] 将大语言模型生成的约束与 SMT求解器反馈相结合，从而提高了形式化推理任务的准确率。这种集成在增强正确性的同时，减少了人工工作量。其他互补方法也丰富了这一研究领域。Beach等人 [25] 提出了一种语义框架，将法律专业知识与实现过程解耦，使非技术用户能够参与合规建模。此外，研究人员还探讨了反事实解释 [26], [27]，以提高算法决策的透明度和可争议性，尤其是在《通用数据保护条例》等数据保护法的框架下。

我们的工作将大语言模型–SMT 集成应用于监管合规，建立在上述趋势之上。我们不只关注翻译或对话建模，而是构建端到端流程，支持约束合成、验证和交互式纠正，从而提供可解释、可验证的法律分析。

III. 方法

A. *SMT*约束生成

我们首先介绍 SMT 约束生成流程。图1展示了将非结构化法律文件和案件文件转换为形式化 SMT 约束的整体过程。该流程始于数据收集和预处理，在此过程中对法律条文和案件文本进行解析和结构化。随后，提取关键法律要素及其逻辑关系，以构建适用于自动推理的高质量约束。

1) 案件爬取与初步条文提取：我们从公共存储库收集了金融监管案件摘要，每份摘要都包含结构化元数据和描述违规行为的叙述。我们使用大语言模型提取案例描述中明确提及的法律条款。

2) 法律条文检索与解析：针对每条被引用的条款，从官方法律数据库中检索相应的法律文本，并使用基于规则的算法（算法1）进行解析。提取出的条款被整理为字典，将条文编号映射到相应的标题、条款内容和全文内容。

随后将每个案例映射到其引用的条款，形成适用于语义检索和推理的结构化案例法关联数据集。

算法1 原始法律文档中的条款与款项提取

输入： 原始文档 D
输出： 字典 A 将条文编号映射到标题，条款和内容

初始化空文本 $T \leftarrow$""
使用pdfplumber打开 D ；提取并拼接文本到 T
将 T 拆分为多行 L
初始化空字典 A；设置 $a \leftarrow$None， $c \leftarrow$ None
对于每一行 l 在 L**执行**
  规范化 l （去除并合并空白）
  如果 l 匹配章节标题模式**则**
    继续
  结束 if
  如果 l 匹配 "条款 n " **then**
    $a \leftarrow n$；初始化 $A[a]$ ，包含标题、空条款，以及内容
    $c \leftarrow$无
  否则，如果 $a \neq$无 **则**
    如果 l 匹配条款模式（例如"$m.$ 文本"）**则**
      将条款文本追加到 $A[a]$[条款]；设置 $c \leftarrow text$
    否则，如果 $c \neq$无 **则**
      追加 l 到$A[a]$[最后一条条款]
    else
      追加 l 到$A[a]$[内容]
    结束 if
  结束 if
结束循环返回 A

3) 补充条款检索（基于 *RAG* 的增强）：为确保全面涵盖法律语境，检索增强生成（RAG）模块通过从法律数据库 $\mathcal{D}$ 中检索语义相关的条款和施行细则，扩展每条基础条款 L_0 。此过程通过纳入上下文相关的辅助条款，提高了 SMT 约束的完备性。

该流程详见算法 2，包含四个阶段：1）**查询生成：** 大语言模型生成一组多样化的查询 $\mathcal{Q} = \{q_1, \ldots, q_m\}$，以捕捉 L_0 的不同语义方面。

2）**混合检索：** 每个查询 q 都会执行 BM25 和向量搜索：

$$R_{\text{bm25}} = \text{BM25Search}(\mathcal{D}, q, K),$$
$$R_{\text{vec}} = \text{VectorSearch}(\mathcal{D}, q, K).$$

结果合并为 $R = R_{\text{bm25}} \cup R_{\text{vec}}$，并通过混合相关性函数进行评分：

$$s_{\text{hybrid}}(d, q) = \alpha \, \text{sim}_{\text{vec}}(d, q) + (1 - \alpha) \, \text{score}_{\text{bm25}}(d, q),$$

where $\alpha = 0.8$ balances semantic and lexical relevance.

3) 重排序与过滤： 前 K 个候选项由交叉编码器（FlagReranker）重新排序，并通过 LLM FilterUseful过滤，以保留不冗余且与上下文相关的条款。**4) 聚合：** 来自所有查询的过滤后条款 C 将合并并去重，形成最终补充集合$\mathcal{S}$。

该检索模块以*LegalSearchEngine*实现，整合了语义检索与词法检索、混合评分、重排序和大语言模型过滤。该模块使用*text-embedding-ada-002* 生成嵌入，使用*ChromaDB* 进行向量索引，并使用*BAAI/bge-reranker-v2-m3* 进行重排序。

算法 2 通过检索增强生成扩展补充条款——增强生成

输入： 基础条款 L_0；数据库 $\mathcal{D}$；权重 $\alpha \in [0, 1]$；检索数量 K
输出： 补充集合 $\mathcal{S}$

$\mathcal{Q} \leftarrow$大语言模型 GenQueries(L_0)
$\mathcal{S} \leftarrow \emptyset$
遍历 $q \in \mathcal{Q}$ **执行**
  $R_{\text{bm25}} \leftarrow$ BM25 搜索($\mathcal{D}$， q， K)
  $R_{\text{vec}} \leftarrow$ 向量搜索($\mathcal{D}$， q， K)
  $R \leftarrow R_{\text{bm25}} \cup R_{\text{vec}}$
  遍历 $d \in R$ **执行**
    计算混合得分：
    $s_{\text{hybrid}}(d) = \alpha \cdot$相似度$_{\text{vec}}(d, q) + (1 - \alpha) \cdot$得分$_{\text{bm25}}(d, q)$
  结束循环
  $R_{\text{top}} \leftarrow$ TopK(R， s_{hybrid}， K)
  $R' \leftarrow$ 重排序(R_{top}， 交叉编码器 q)
  $C \leftarrow$ LLM_FilterUseful(R', L_0)
  $\mathcal{S} \leftarrow \mathcal{S} \cup C$
结束于return $\text{Unique}(\mathcal{S})$

*4) SMT*约束生成：在获得案件—条款对后，系统继续生成 SMT（可满足性模理论）约束，以形式化表示事实情境与相应法律条款之间的关系。生成约束后，系统会执行一系列验证和一致性检查，以确保其正确性和逻辑可靠性。

a) 构建法律约束：法律推理要求所有行为都符合法定规范。为将这些规则形式化，我们使用大语言模型释义，将法律条款转换为表示其逻辑结构的布尔表达式。每个法条款项都表示为一组命题公式，通常涉及合取、析取和蕴含关系，以捕捉法律所规定的条件关系。

为说明这一过程，我们 प्रस्तुत来自金融监督管理委员会（FSC）的一个真实案例。[1] 通过使用我们的提取和RAG模块（第III-A3节），相关条款

[1]https://www.fsc.gov.tw/en/home.jsp?id=131&parentpath=0,2&mcustomize=multimessages_view.jsp&dataserno=202407120004&dtable=处罚

已从《保险法》（第143-4条和第143-6条）[2] 及其施行细则（第2至第5条）中检索出来。[3] 这些条款规定了资本适足性分类及相应的监管措施。编码后的SMT硬约束总结如下。

设 $r = \frac{\text{自有资本}}{\text{风险资本}} \times 100$。则资本等级分类定义如下：

$$capital_level = \begin{cases} 4, & r < 50 \text{ or } net_worth < 0, \\ 3, & 50 \le r < 150, \\ 2, & 150 \le r < 200, \\ 1, & r \ge 200. \end{cases}$$

监管措施表示为：

$cL2$执行 $\Leftrightarrow$ (计划已提交 $\wedge$计划已执行)，

$L3$执行 $\Leftrightarrow$ ($L2$执行 $\wedge$ 人员已移除 $\wedge$ 资产已批准)。

处罚规则简洁地定义如下：

$$\begin{aligned} penalty = 1 \text{ if } & (level = 4 \wedge \neg L4_exec) \\ & \vee (level = 3 \wedge \neg L3_exec) \\ & \vee (level = 2 \wedge \neg L2_exec), \end{aligned}$$

$$penalty = 0 \text{ otherwise.}$$

由此产生的公式集合构成法律约束库，为SMT编码及后续逻辑验证奠定基础。

b) 构建合规事实约束：除法定约束外，每个监管案件都包含必须形式化为逻辑谓词以供推理的事实条件。我们构建一组合规事实约束$\mathcal{H}_{\text{case}}$来表示案件的事实状态，例如实际执行的资本充足率、整改措施或监管措施。这些事实谓词以布尔值或数值形式编码，并与法定约束基础 $\mathcal{H}_{\text{law}}$ 结合，形成对案件的完整逻辑表示。

对于上述 FSC 案件，提取出的事实输入如下：

```
自有资本 = 111.09，风险
资本 = 100.0，净值 = 2.
97，已提交计划 =true，
已执行计划 =false
```

对于案件描述中未明确提及的变量，系统将其视为自由变量，而不为其赋予默认布尔值。这种设计使求解器能够在约束满足过程中动态探索可行的赋值，确保推理过程忠实于现有事实范围，同时避免作出没有根据的假设。

这些事实赋值构成针对具体案例的约束集 $\mathcal{H}_{\text{case}}$，将其与法律约束

[2]https://law.moj.gov.tw/ENG/LawClass/LawAll.aspx?pcode=0390002G

[3] https://law.moj.gov.tw/LawAll.aspx?pcode=G0390003

库 $\mathcal{H}_{\text{law}}$合并后，即可生成完整的可满足性模理论问题实例：

$$\mathcal{H}_{\text{law}} \cup \mathcal{H}_{\text{case}}.$$

这种统一表示使求解器能够验证案件的事实状态是否符合定法框架，或违反一项或多项编码条款。因此，它为后续的一致性检查和合规优化提供了一个基于逻辑的测试实例。

B. SMT 约束一致性检查与优化

1) 法律约束一致性检查：将所有法定条款编码为硬约束后，我们会进行一致性验证，以确保法律逻辑内部自洽。编码后的法律公式集合

$$\mathcal{H}_{\text{law}}$$

将提交给 SMT求解器。可满足（可满足）结果表明，法定条款在逻辑上是一致的，不存在内部矛盾。这有助于确保编码后的法律构成逻辑一致的基础。反之，若结果为不可满足，则表明法律编码内部存在不一致，需要反复完善，直至系统得到一致的 可满足 结果。

2）案件违法性检查：在确认法定约束在内部保持一致后，我们整合案件特定事实，并进行第二阶段的验证。该案件违法性检查通过评估组合约束集，确定事实条件是否正确触发了违反法定规定的情形：

$$\mathcal{H}_{\text{law}} \cup \mathcal{H}_{\text{case}} \cup \{Penalty = \text{False}\}. \tag{1}$$

如公式（1）所示，组合约束集同时纳入了法定约束与案件特定假设，并将处罚条件明确固定为*False* ，以检验潜在的违法行为。

在此阶段，预期结果为不可满足，这表明事实配置与合法条件不相容，因此构成有效的不合规情形。求解器证明不存在能够同时满足法定约束与事实约束的赋值，从而确认存在违法行为。

对于FSC案件，求解器确实返回不可满足，揭示出以下各项之间存在矛盾：

```
保险：资本水平、元：处罚条件、保险：
已执行三级措施
```

这些条目构成了不可满足核，确定了导致不一致的最小约束子集。该冲突表明，由于纠正措施不完整，违反了第143-4条和第143-6条，这与实际的行政处分一致。

如果求解器返回的是可满足，则说明逻辑表述未能捕捉预期的违法情形，系统会启动迭代优化过程，重新生成或调整编码后的约束，直到获得预期的不可满足 结果

。因此，案件违法性检查 会验证编码后的案件是否确实构成法定框架下的违法行为，但在此阶段不会尝试恢复可满足性，也不会修改任何事实谓词。

在获得有效的不可满足 结果后，系统进入优化阶段。在随后的步骤中，Z3 的*Optimize()* 模块会应用于同一约束集，将个案特定谓词放宽为软约束，同时恢复法定条款。

3) 最小合规解决方案的优化： 在完成上述验证和调整阶段后，优化阶段将寻找最小合规解决方案的过程形式化。在此，我们将先前识别出的不可满足 配置视为优化目标，并使用 Z3 的*Optimize()* 模块，推导出能够恢复完全逻辑可满足性的最小事实变更集。

构建硬约束和软约束后，我们使用 Z3 的*Optimize()* 模块计算最小合规解决方案。该过程会搜索一个模型，使其满足所有法定（硬）约束，同时尽可能少地修改事实（软）约束，从而以最小干预实现法律合规。从监管案例中提取的所有事实陈述都会编码为软约束，并通过add soft() 接口添加到求解器中。每个软约束 s_i 都会被赋予权重 $w_i = 1$，除非监管优先级另有规定–(例如，处罚可能会被赋予更高权重）。硬约束集 $\mathcal{H}$ 表示在任何情况下都必须成立的不可变法定义务。

*Optimize()*模块在内部通过加权 MaxSAT 方法运行，旨在最大化软约束的总体满足度，同时保持所有硬约束的逻辑一致性。形式上，优化问题可表示为：

$$\max_{\mathbf{x}} \quad \sum_{s_i \in \mathcal{S}} w_i \cdot \mathrm{s_i}(\mathbf{x}) \quad \text{s.t.} \quad \mathcal{H}(\mathbf{x}) = \text{true}, \tag{2}$$

其中，$\mathrm{s_i}(\mathbf{x}) \in \{0,1\}$ 表示 s_i 在模型$\mathbf{x}$下是否得到满足。

如果不存在完全可满足的赋值，Z3 会逐步松弛被违反的软约束，引入辅助布尔选择器 $\lambda_i \in \{0,1\}$ ，并最小化其激活数量：

$$\min_{\mathbf{x}} \sum_{i=1}^{|\mathcal{S}|} w_i \lambda_i, \quad \text{s.t.} \quad (\mathcal{H}(\mathbf{x}) \wedge \bigwedge_i (\mathrm{s_i}(\mathbf{x}) \vee \lambda_i)) = \text{true}. \tag{3}$$

如公式（2）所示，优化旨在法定框架下最大化所有软约束的满足度。当无法实现这一点时，公式（3）定义了松弛过程：通过激活数量最少的 λ_i 变量来最小化总违规成本。

该机制有效地搜索所有事实条件的极大可满足子集（*MSS*），对应于恢复法律合规所需的最小逻辑修改。概念上，Z3 会迭代放宽权重最低的约束，直到所有硬约束一致，

算法 3：通过加权 MaxSMT 实现最小合规 输入：硬约束 $\mathcal{H}$；软约束 $\mathcal{S} = \{s_i\}$ 及其权重 $\{w_i\}$输出：最小合规模型 $\mathbf{x}^*$；违规集合 Δ；成本$C(\Delta)$ 1： $O \leftarrow$Optimize() ▷ 初始化 MaxSMT 优化器 2：通过 O.add(h) 添加所有 $h \in \mathcal{H}$ 3：通过 O.add soft(s_i, w_i) 添加所有 $s_i \in \mathcal{S}$ 4：运行 status $\leftarrow O$.check() 5：如果 status =unsat，则 6：返回无可行合规方案：硬约束不一致 7：结束 if 8：$\mathbf{x}^* \leftarrow O$.model() ▷ 最优的最小违规模型 9：在 $\mathbf{x}^*$ 下评估每个 s_i ；收集 $\Delta \leftarrow \{s_i \mid s_i(\mathbf{x}^*) = \text{false}\}$ 10 $C\ \Delta \leftarrow \sum_{s_i \in \Delta} w_i$：计算 ( )：返回 x , , ( ) 11 $^*\ \Delta\ C\ \Delta$

在固定的法定框架下保留尽可能多的事实假设。

在实践中，求解器会找出最小的事实修订——例如翻转一个布尔值或调整阈值——从而得到满足 $\mathcal{H}$ 并最小化 $\sum_i w_i\lambda_i$的$\mathbf{x}^*$ 。如果$\mathcal{H}$ 不一致，求解器将返回不可满足。

对于 FSC 案例，优化结果表明只需改变一个事实：

已执行改善计划: false →true

这表明，该公司执行最初提交的改善计划后，即可恢复法律合规，满足《保险法》第143-4条和第143-6条的要求。

为增强可解释性，优化结果包括最优模型$\mathbf{x}^*$ 和差异集 Δ，其中列出了优化过程中变更的事实变量。 Δ 中的每个元素都指出了哪项事实条件违反了合规要求，以及该条件如何调整，从而形成一条明确的修正轨迹，供审计员或法律分析师解读。这条轨迹直接支持事后解释和模型问责。

每项违反的约束都对应一个具体的不合规要素，从而实现透明的追溯和可供人工审查的推理。与语言模型的解释模块集成后， Δ 中的每项违规都会转化为自然语言陈述，例如：“该公司未在法定期限内提交补救计划。”这一混合推理程序符合最小干预的法律原则，确保通过尽可能少的事实修正恢复合规，同时维护逻辑和法定要求的完整性。

该优化旨在维护逻辑可靠性与可解释性，为实现透明、可验证的合规自动化奠定基础。

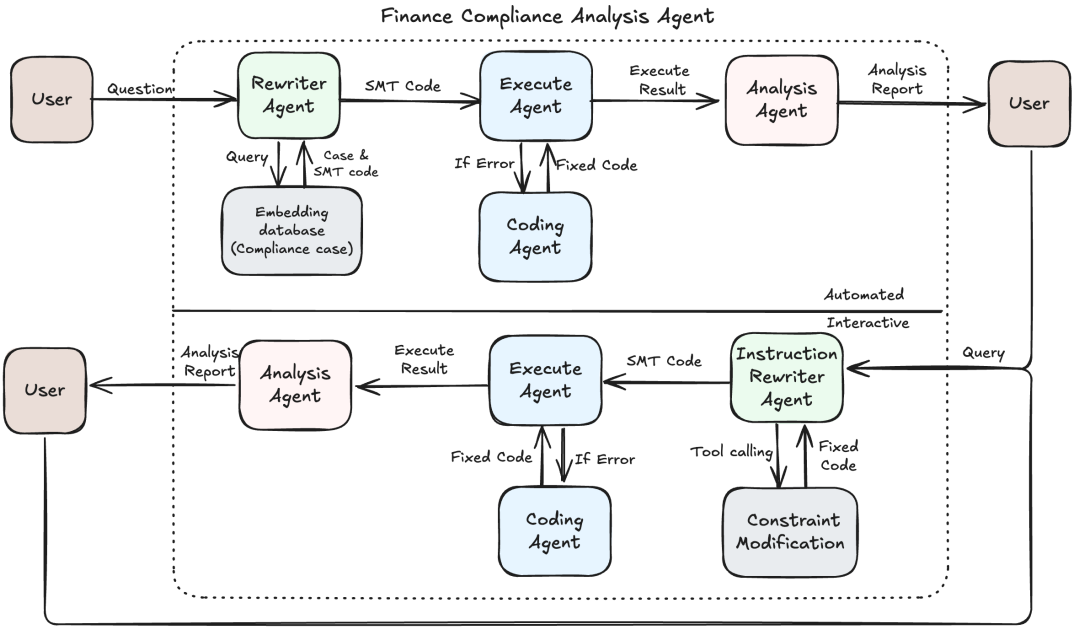


图 2：金融合规分析智能体的系统架构。

优化结果当*check()* 返回可满足时，由*Optimize()* 生成的模型$\mathbf{x}^*$ 会在所有满足 $\mathcal{H}$的模型中，使$\sum_i w_i\lambda_i$ 最小。因此，所导出的差异集 Δ 是事实违规的加权最小集合。在均匀权重下，这等同于最小基数修订；在非均匀权重下，则得到最小成本修订。在这两种情况下，$\mathbf{x}^*$ 都构成所需的最小合规解决方案。

C. 多智能体合规平台

在约束生成和验证框架的基础上，我们设计了一个模块化多智能体平台，将推理过程付诸实践。

1) 设计原则：金融合规分析智能体的设计遵循四项核心原则：(1) 模块化，使智能体能够独立替换或升级；(2) 可解释性，通过结构化分析和符号推理来确保；(3) 人在回路中的适应性，支持通过自然语言进行迭代优化；(4) 适应性，支持未来扩展至其他监管领域或司法管辖区。

2) 人工智能智能体系统架构：图 **??** 展示了金融合规分析智能体的架构。该模块化多智能体系统旨在评估金融法律合规性。工作流程始于用户查询，随后依次经过多个专业智能体，每个智能体负责推理与验证中的一个特定阶段。

改写智能体将用户查询重述为结构化的法律任务，并从基于嵌入的合规案例数据库中检索相关判例。该数据库存储了预先生成的用于约束评估的SMT代码。随后，执行智能体使用SMT求解器运行检索到的SMT代码，以验证其是否符合编码后的法律约束，既支持自动化评估，也支持交互式完善。

如果发生执行错误（例如语法不一致、格式错误的逻辑或未绑定变量），编码智能体会与执行智能体协作修复 SMT 代码。它会检查求解器跟踪记录和对话历史，以识别稳定的先前版本或生成纠正补丁，从而在求解器与智能体之间实现自我修复反馈循环。

分析智能体会解读求解器输出，并将其转换为人工审计员能够理解的结构化合规报告。当用户提供反馈或建议修改规则时，指令改写智能体会将这些输入转换为更新后的 SMT 逻辑，以便重新评估。最后，约束修改模块会对SMT公式进行底层编辑，例如切换逻辑条件、注入参数或修正变量赋值。

这些智能体协同工作，构成了一个稳健且自适应的工作流，将符号推理、自动纠正和交互式人工指导融为一体，从而实现端到端的法律合规分析。

表 I：大语言模型生成的 SMT 摘要 code

方法	准确率	平均时间（秒）
大语言模型生成SMT代码	0.8621	36.85

3) 智能体协作与工作流：系统通过各推理阶段的协同合作运行。改写智能体重新表述用户查询并检索相关的可满足性模理论案例，执行智能体则通过可满足性模理论求解器进行验证。出现错误时，编码智能体会启动自修复机制——追踪故障源头、执行

表 II：87 个案例中变量和约束的描述性统计。

	#Var	Real	Int	Bool	#Con	Hard	Soft	硬约束比例 —	软约束比例 —
平均值	12.85	3.58	1.97	7.31	30.81	19.30	11.51	0.63	0.37
Min	4	0	0	1	11	6	4	0.38	0.24
Max	40	21	7	21	63	39	25	0.76	0.63
Std	5.60	3.40	1.74	4.11	9.62	6.35	4.61	0.08	0.08

回滚或生成逻辑补丁——直至约束得出有效的可满足或不可满足结果。

成功执行后，分析智能体会将结果汇总为可解释的合规报告。为便于交互式探索，用户可以修改假设或逻辑；指令改写智能体会将这些输入转换为结构化工具调用，并将其分派给约束修改模块，由该模块更新并重新验证 SMT代码。

这种模块化、面向智能体的工作流程确保了可解释性、容错性和适应性，支持在不断演变的金融监管环境中开展自动化合规推理以及人机协同的合规推理。

IV. 实验评估

为验证我们的框架，我们针对台湾金融监督管理委员会发布的金融罚款案例进行了实证分析。我们共收集并分析了 87 个真实案例。我们还使用 Gradio 开发了一个交互式用户界面，以便与系统进行实时互动。作为基础设施的一部分，我们整理了一个全面的监管数据库，其中包含 75 部不同的法律，以及从台湾法学资料库中提取的 3,753 条法定条款。

本实验旨在回答以下研究问题：

- **RQ1**：大语言模型在法律领域生成可满足性模理论约束的能力有多强?
- **RQ2**：与仅使用大语言模型相比，将大语言模型与可满足性模理论相结合，是否能提升模型推理真实监管案件中非法条款的能力?
- **RQ3**：与单独使用大语言模型相比，将大语言模型与可满足性模理论相结合，是否能提高真实案例中法律合规修复的准确率和一致性?

我们通过定性和定量评估来探讨这些问题，重点关注推理表现、法律一致性和纠错准确率。

所有补充材料——包括详细的 SMT公式、其他案例研究和实现脚本——均已在 GitHub [28] 上公开提供。

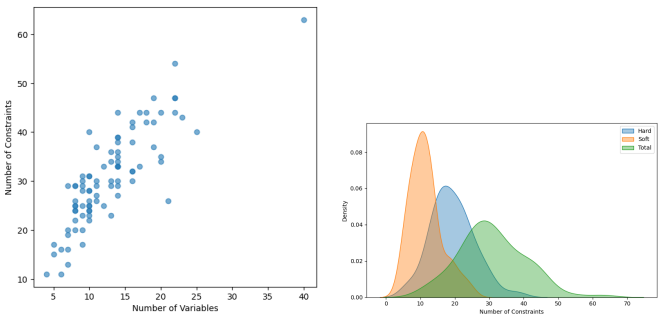
A. *RQ1*：*SMT*约束合成

我们对 87 起监管案件进行了生成实验。采用少样本学习方法，我们提示大语言模型针对每个案件生成相应的 SMT 约束。随后对生成的代码进行人工审查，以验证其逻辑可靠性和上下文正确性。如果代码有效，则将其归类为 正确；否则，我们提示模型反复修改其输出，直至问题得到解决，此类案件则被归类为不正确。人工检查仅用于验证生成的 SMT 约束与原始案件背景之间的事实一致性。我们未对模型输出进行任何人工编辑或修正；为确保公平比较，人工干预仅限于事实核验，不影响任何中间推理或求解器纠正步骤。该流程确保所有评估均客观反映模型的内在能力。

常见错误包括遗漏合规结果中受影响的客户、未在适用情况下将罚款设为零，以及产生逻辑上不一致的约束集，导致求解器返回结果异常。通过人工修正和验证，我们确保所有最终的 SMT代码均准确无误，可作为下游评估的真实值。

生成准确率和响应时间汇总于表 I。

案例研究规模分析。为明确案例研究的规模，我们分析了全部 87 个案例中的变量和约束数量。详细统计数据汇总于表 II。平均而言，每个案例约包含 13 个变量（#Var）和 31 个约束（#Con），规模最大的实例涉及 40 个变量和 63 个约束。约 63% 的约束为硬约束（硬约束比例），而 37% 为软约束（软约束比例），体现了严格监管规则与灵活规划考量之间的平衡。



(a) 变量与约束　　(b) 约束分布

图 3：变量与约束之间的关系 (a)，以及硬约束、软约束和约束总数的分布 (b)。

如图 3 所示，变量数量与约束数量之间的散点图呈现出明显的正相关。这一观察表明，随着变量数量增加，约束集也按比例增长，说明所提出的框架适用于中小规模的可满足性模理论问题。

图 3 还展示了硬约束、软约束和约束总数的分布。硬约束主要集中在 15 到 25 之间，而软约束大多分布在 8 到 15 之间。因此，约束总数通常在 25 到 40 之间，少数离群值超过 60。

表 III：非法术语分析：大语言模型与大语言模型+可满足性模理论的比较。

方法	每个案例的平均非法条款数	平均时间（秒）
大语言模型 +可满足性模理论	5.08	0.021
仅使用大语言模型	6.40	7.01

B. RQ2：非法术语推理

为支持现实法律案例中的合规工作，我们的系统旨在识别每个场景中的具体非法条款——即法定违规行为。突出显示这些条款有助于用户了解必须采取哪些行动才能符合相关法规。如表 III 所示，我们通过比较基线大语言模型与我们提出的 LLM+SMT 方法识别出的非法条款数量及平均响应时间，评估了这项能力。

我们的基于 SMT 的方法在代码生成过程中利用 *assert* 和 *track* 命令。当出现矛盾时——即 Z3 返回 *unsat* 结果时——这表明法定要求与案例描述之间存在违反法定规定的情况。例如，如果法律要求针对资本不足采取补救措施，但案例中未提及这些措施，系统就会标记出这一不一致。

由于 Z3 默认只返回一个最小不可满足核，我们将所有约束（包括原本的软约束）重新定义为硬约束，迫使求解器显示多个不可满足核。随后，我们提取各轮迭代中追溯到硬约束的所有组成部分，以确定完整的“非法条款”集合。软约束中的其余元素则作为自由变量处理，并逐步重新引入，直至系统达到可满足状态。通过对这些轮次的结果取集合并集，计算非法条款的唯一总数。

为验证这一机制，我们还使用相同的案例和法律文本直接向大语言模型提问。然而，大语言模型的输出往往缺乏逻辑一致性和可解释性。相比之下，我们的 SMT 增强方法不仅能得出相当的结果，还能将推理效率提升两个数量级以上，并提高可解释性。

在一个真实的保险案例中，[4] 我们的可满足性模理论模块在 0.191 秒内识别出以下非法条款：要求净资产、自有资本总额、风险资本总额、定义改进计划、要求最终资本充足率、定义前期资本充足率以及定义总改进额。这些对应着必须解决的正式法定违规行为，以恢复合规状态。

[4]参见金融监督管理委员会对三商美邦人寿保险股份有限公司的处罚公告：https://www.fsc.gov.tw/ch/home.jsp?id=131&parentpath=0,2&mcustomize multimessages view.jsp&dataserno=202407120004&dtable=处罚

相比之下，大语言模型耗时约 5 秒，并返回了一份更长、精确度较低的列表，其中包括：资本充足率低于法定水平、净值比率低于法规要求、资本严重不足、未提交可行的资本改善计划、未确保 2024年 的资本充足性、计划中的混合型工具私募存在高度不确定性、备用计划不足、违反《保险法》第 143-6 条第 1 款第 1 项、与关联方进行受限交易，以及不符合资本适足性法规。

进一步检查发现，这些额外术语可分为三类：**(1)情境误分类**——指示存在财务风险、但并非违法行为的情况（例如，私募的不确定性）；**(2) 咨询性条款的误解**——将软性建议错误地标记为强制要求（例如，缺少应急计划）；**(3) 模糊或复合标签**——过于笼统，无法追溯到具体条款的陈述。

这些发现凸显了我们基于可满足性模理论的方法的关键优势：精确的法律建模、符号化一致性检查，以及识别可采取行动的违规行为的能力。在金融监管等审计性和精确性至关重要的领域，这一优势尤为宝贵。

C. RQ3：合规恢复

在分析非法条款识别之后，我们进一步评估模型纠正现实案例的能力。此任务考察需要进行哪些修改，才能使案例修订后的法律结果不再受到处罚。我们对全部87个案例进行了这项纠正分析。

对于基于可满足性模理论的方法，我们提高了与处罚相关的软约束的权重，并将目标处罚值设为零。这样，SMT求解器便能返回一个可满足的赋值，从而消除处罚，并将其作为比较的真实值 。重要的是，此任务中的真实值并非指某个唯一的赋值，而是指案例的可满足性状态（可满足 或不可满足）。对于可满足的实例，可能存在多个有效赋值；任何满足所有约束的赋值都被视为正确。因此，我们的评估不要求大语言模型复现某个特定的SMT生成赋值；只要在存在可满足解时生成一个有效的可满足解即可。

总共有 63 个案例可满足，其中只有 13 个附带了供参考的示例赋值，其余案例则不局限于单一解。这种设置符合现实中的法律推理，因为同一案例可能存在多种合规的修订方案。

在此实验中，向大语言模型提供软约束变量及其原始值。它可以自由决定是否翻转给定变量的值，或为其赋予新值。大语言模型还会收到相应的案例文本和法条依据，以支持其推理过程。

为进一步评估模型鲁棒性，我们随机选取了 24 个案例，并在

表 IV：合规恢复案例的性能。

方法	准确率	精确率	0	F1分数	平均时间（秒）
大语言模型 + 可满足性模理论	1.0000	1.0000	1.0000	1.0000	0.0040
仅使用大语言模型	0.3333	0.6190	0.2063	0.3080	1.4700

表 V：未使用 SMT 求解器的大语言模型的混淆矩阵。

预测	实际可满足	实际不可满足
可满足	13（真正例）	8（假正例）
不可满足	50（假负例）	16（真负例）

求解器中引入了矛盾约束。例如，我们可以规定资本适足率必须超过 200%，该比率按自有资本除以风险加权资本计算。如果随后添加一项将自有资本设为零的约束，系统就会变得不可满足。这些新增约束也会提供给大语言模型。如果大语言模型能够正确推理出该问题没有可行解，就应返回不可满足的判断。

性能对比见表 IV，且仅大语言模型基线的混淆矩阵见表 V。

从结果中可以看出，大语言模型往往默认预测为不可满足，即使存在有效解的案例也是如此。这种偏差导致大量假负例。此外，与 SMT 求解器相比，大语言模型的合规恢复缺乏一致性和完备性。相比之下，基于 SMT 的方法在合规纠正任务中准确率较高，推理速度也显著更快，每次纠正约需 0.004 秒，比大语言模型的平均响应时间快 300 多倍。

D. 实验结果总结

在全部三个研究问题中，结果始终表明，将 SMT 求解器整合进法律推理流程是有效的。对于**RQ1**，大语言模型在超过86%的案例中生成了语法和语义均正确的 SMT代码，从而能够精确地建模法律约束。在**RQ2**中，SMT增强方法在非法条款识别方面达到了与基线大语言模型相当的水平，同时响应更快、逻辑更清晰。对于**RQ3**，由大语言模型+SMT驱动的合规恢复达到了完美的准确率，在所有主要指标上均优于基线。总体而言，混合式大语言模型+SMT框架兼具稳健性和可解释性，在实际法律推理和合规纠正方面展现出巨大潜力。

跨司法管辖区适应性。尽管我们的评估聚焦于台湾的金融监管环境，所提出的框架不依赖特定司法管辖区。通过调整检索语料库和提示示例，以反映目标司法管辖区的判例和法律结构，该系统可扩展至美国或英国等普通法体系。未来的工作将探索多语言和判例驱动的适配，以进一步评估跨司法管辖区适用性。

V. 结论

我们提出了**神经符号合规**，这是一种混合框架，通过多智能体设计，将基于大语言模型的理解与基于可满足性模理论的推理相结合。每个智能体都针对特定领域的专业知识进行建模，从而能够对金融监管开展可解释、可验证的分析。在台湾的 FCS 案例中，我们的系统在推理一致性和推理速度方面优于仅使用大语言模型的基线。这项工作展示了基于智能体的神经符号人工智能如何连接自然法律语言与机器可验证逻辑，推动可扩展的监管自动化。

致谢

本研究部分受国家科学及技术委员会（NSTC，台湾）资助，项目编号为113- 2221-E-004-010-MY2和114-2221-E-002-183-MY3；另获台大—台达电子创新研究基金项目资助，项目编号为NTU-DE-114FR032；并获中华电信股份有限公司委托项目资助，项目名称为“国立政治大学金融人工智能研究项目：繁体中文大语言模型与人工智能智能体平台在投资分析报告生成中的应用”。

表 VI：国际化　　潜力：法律领域迁移所需的最少工程修改　　er

模块	目的	是否应修改?
检索增强生成语料库	检索相关法律文本和判例	是（应替换为本地法律语料库。）
提示模板	控制查询格式和语言语境	是（应根据语言和法律措辞进行调整。）
条款解析器	将法律条款映射为逻辑模板	部分适用（应重新定义解析模式。）
法律映射词典	将条款标签关联至法规	是（应针对目标司法管辖区进行更新。）
Z3规则引擎	强制执行符号化法律约束	否（无需修改即可重复使用。）
SMT松弛循环	执行最小修正搜索	否（它仍然与领域无关。）
Gradio 用户界面	提供用户界面	否（它可以直接重复使用。）

参考文献

[1] T. Libal和T. Novotn´a，“迈向透明的法律形式化”，载于《可解释与透明人工智能及多智能体系统：第三届国际研讨会》，EXTRAAMAS 2021，7，2021年，3 3 2021，296 313线上会议，5月——修订精选论文。Springer，第–页。[2] I. Ponkin和A. Redkina，“法律的数字化形式化”，《国际开放信息技术期刊》，第7卷，48，2019年1，39第–页。[3] H. Surden，“可计算法律与人工智能”，科罗拉多大学法学院法律研究论文（即将发表），《剑桥私法与人工智能手册》（将于2024年发表），2024年。[4] S. E. Mohsen、A. Hamdan和H. M. Shoaib，“金融机构的数字化转型与人工智能整合”，《财务报告与会计期刊》，2024年。[5] R. J. Bolton和D. J. Hand，“统计欺诈检测：综述”，《统计科学》，第17卷，第3期，第235–255页，2002年。[6] D. W. Arner、J. Barberis和R. P. Buckley，“金融科技的演进：危机后的新范式”，《乔治敦国际法杂志》，第47卷，第1271页，2015年。[7] C. Barrett、C. L. Conway、M. Deters、L. Hadarean、D. Jovanovi´c、T. King、A. Reynolds和C. Tinelli，“cvc4”，载于《计算机辅助验证：第23届国际会议，CAV 2011》，美国犹他州雪鸟城，2011年7月14日至20日。会议录第23卷。Springer，2011年，第171–177页。[8] L. De Moura和N. Bjørner，“Z3：一种高效的SMT求解器”，载于《系统构造与分析工具及算法国际会议》。Springer，2008年，第337–340页。[9] C. A. Bradley和T. W. Morrison，“总统权力、历史惯例与法律约束”，《哥伦比亚法律评论》，第1097–1161页，2013年。[10] S. Athul、A. Saxena、J. Mahajan、L. Panikulangara、S. Kulkarni和S. Bang，“Legalmind系统与基于大语言模型的法律判决查询系统”，载于《2024年量子计算与新兴商业技术趋势国际会议》。IEEE，2024年，第1–5页。[11] D. Shu、H. Zhao、X. Liu、D. Demeter、M. Du和Y. Zhang，“LawLLM：面向美国法律体系的法律大语言模型”，载于《第33届美国计算机协会信息与知识管理国际会议论文集》，2024年，第4882–4889页。[12] P. Hitzler、A. Eberhart、M. Ebrahimi、M. K. Sarker和L. Zhou，“人工智能中的神经符号方法”，《国家科学评论》，第9卷，第6期，文章编号nwac035，2022年。[13] J. Mao、J. B. Tenenbaum和J. Wu，“神经符号概念”，arXiv预印本arXiv:2505.06191，2025年。[14] Y. Hao、Y. Zhang和C. Fan，“严谨地规划任意任务：基于大语言模型的形式化编程实现通用零样本规划”，arXiv预印本arXiv:2410.12112，2024年。[15] M. Chen、G. Li、L.-I. Wu、R. Liu、Y. Su、X. Chang和J. Xue，“语言模型能否模拟求解器？使用大语言模型进行逻辑代码模拟”，载于《可信软件工程：理论、工具与应用国际研讨会》。Springer，2024年，第102–121页。[16] Y. Zhang、Y. Cai、X. Zuo、X. Luan、K. Wang、Z. Hou、Y. Zhang、Z. Wei、M. Sun、J. Sun等，“大语言模型与形式化方法融合以构建可信人工智能智能体：路线图”，arXiv预印本arXiv:2412.06512，2024年。[17] A. Berger、L. Hillebrand、D. Leonhard、T. Deußer、T. B. F. De Oliveira、T. Dilmaghani、M. Khaled、B. Kliem、R. Loitz、C. Bauckhage和R. Sifa，“利用大语言模型实现财务审计中的自动化监管合规验证”，载于《2023年IEEE大数据国际会议（BigData）》，2023年，第4626–4635页。[18] H. Liao、S. He、Y. Xu、Y. Zhang、K. Liu和J. Zhao，“神经符号协同蒸馏：推进小语言模型处理复杂推理任务”，载于《AAAI人工智能会议论文集》，第39卷，第23期，2025年，第24 567–24 575页。[19] M. Ning、Z. Zhou、Q. Wang、X. Huang和K. Huang，“GNS：通过多模态大语言模型的神经符号推理解决平面几何问题”，载于《AAAI人工智能会议论文集》，第39卷，第23期，2025年，第24 957–24 965页。[20] J. Shi、X. Ding、H. Zhao、T. Liu和B. Qin，“衔接神经与符号推理：用于可解释问答的双系统框架”，载于《ICASSP 2025——2025年IEEE声学、语音与信号处理国际会议（ICASSP）》，2025年，第1–5页。[21] S. Judson、M. Elacqua、F. Cano、T. Antonopoulos、B. K¨onighofer、S. J. Shapiro和R. Piskac，“SOID：一种用于自动化决策法律问责的工具”，载于《国际计算机辅助验证会议》。Springer，2024年，第233–246页。

[22] N. Feng、L. Marsso、M. Sabetzadeh 和 M. Chechik，《通过有界可满足性检查提前验证法律合规》，载于国际计算机辅助验证会议。Springer，2023年，第374–396页。[23] C. Tsigkanos、A. Arleo、J. Sorger 和 S. Dustdar，《企业如何进行交易？利用可满足性对金融交易网络进行估算与验证》，载于*2019*年第*20*届*IEEE*数据科学信息重用与集成国际会议（*IRI*）。IEEE，2019年，第15–22页。[24] M. Raza 和 N. Milic-Frayling，《几乎必然推理：借助语言模型和逻辑求解器生成经验证的形式化》。[25] T. H. Beach、Y. Rezgui、H. Li 和 T. Kasim，《建筑领域自动监管合规的基于规则的语义方法》，《*Expert Systems with Applications*》，第42卷，5231，2015年12月，第5219–页。[26] S. Wachter、B. Mittelstadt 和 C. Russell，《无需打开黑箱的反事实解释：自动化决策与《通用数据保护条例》》，《哈佛法律与技术杂志》，第31卷，第841页，2017年。[27] H. Andreas、M. Armgardt 和 M. Gunther，《法律语境中因果责任的反事实》，《人工智能与法律》，第31卷，第1期，第115–132页，2023年。[28] Y.-S. Hsia 和 F. Yu，《神经符号合规》，2025年。 [在线]。地址：https://github.com/shiayongshen/Neuro-symbolic-compliance