

已思考 21s

写入

AI4Drawing（科研结构化绘图草稿引擎）产品 PRD (V1.0)

0. 文档信息

- 产品代号：AI4Drawing – Draft Engine
 - 文档版本：V1.0
 - 目标读者：产品/研发/算法/设计/运营/潜在合作导师与评审
 - 核心定位：面向高校理工科论文/基金/技术报告的“强结构、强逻辑”科研图，提供**可编辑草稿 + 可复刻元信息包 + 结构正确性验证报告**，让用户用传统工具低成本定稿。
-

1. 背景与问题陈述

1.1 背景

科研图（方法框架图、系统架构图、实验流程图等）在论文发表、基金申请书与技术报告中属于“高风险交付物”：图的逻辑正确性、结构清晰性与一致性直接影响评审理解与信任。

1.2 现状痛点

现有方案存在明显断层：

- 传统绘图工具（draw.io/Visio/PPT/Illustrator 等）：可控但耗时；一致性、复用与维护成本高。
- 代码式绘图（Mermaid/PlantUML/TikZ 等）：可版本控制、可复现，但学习门槛高；布局与精修仍需成本。
- LLM/AI 绘图：自然语言生成高效，但对强逻辑结构图存在显著风险，尤其在多轮修改中容易出现**漂移与幻觉误差**；仅靠 LLM 的自然语言微调难以满足高可靠需求。

1.3 关键洞察（来自市场进攻点与专家评价）

- 作为学生团队，不可能全面进攻“通用绘图市场”；必须选择可聚焦、可落地、可形成壁垒的楔子。
- 对强逻辑科研图，“LLM-only 多轮微调”不可靠；正确路线是：**LLM 生成候选 + 形式化约束检查 + 自动修复闭环**。
- 真实科研用户在高风险图上追求细粒度可控：并非一定重画，但强倾向于在传统工具中**编辑可控源文件**完成定稿。因此产品交付物必须是“可编辑资产”，而非仅 PNG。
- 差异化不应是“我也能生成草稿”，而应是：
 - a. **多幅图一致性**（一套图风格与符号一致）
 - b. **可复刻元信息包**（图标/配色/布局/语义映射/规则）
 - c. **结构正确性可证明**（约束检查报告）
- 市场价值必须可量化：以 Time-to-final、编辑距离、结构错误率、跨图一致性等指标定义收益。

- 主要风险：草稿被认为“不值钱”、被通用AI图工具快速覆盖、范围过大导致交付失败；对应策略是：输出可编辑源文件 + 标准化交付物 + 聚焦2-3类高频图。
-

2. 产品目标与非目标

2.1 产品目标 (Must)

1. 在用户输入（文本需求/结构要点/论文段落）基础上，**一次性产出多幅科研结构图草稿**（至少覆盖2-3类高频图）。
2. 草稿以**可编辑源文件**交付（优先：draw.io XML、PPTX shapes、Mermaid/PlantUML代码、SVG分组），并附带**复刻元信息包**。
3. 对每幅图输出**结构正确性验证报告**（约束检查结果、风险点、需要人工确认的冲突点）。
4. 支持跨图一致性：同一概念符号/颜色/命名一致，风格表全局复用。

2.2 非目标 (Out of scope, V1 明确不做)

- 不承诺“一键生成可直接投稿/可直接提交基金的终稿图”。
 - 不追求覆盖全部学科全部图类（生物机制图、复杂电路图等），V1聚焦常用强结构图。
 - 不以美术审美为主战场（终稿审美由用户在传统工具中完成）。
-

3. 用户与场景

3.1 目标用户画像

- 高校理工科研究生/博士生：需要快速产出论文/基金中的结构图草稿，迭代频繁。
- 科研助理/实验室成员：承担图表产出与风格统一工作。
- PI/导师：需要快速看到“方案逻辑图草稿”用于讨论与决策。

3.2 典型场景 (Top 3)

1. 论文方法框架图：输入-处理-输出链路，含模块分组与对比 (Baseline vs Ours)。

2. 系统架构图：分层（数据层/模型层/服务层/应用层），含数据流与控制流。
 3. 实验流程图：数据准备-训练-评测-消融-可视化，含分支与异常处理。
-

4. 产品定位与价值主张

4.1 定位（一句话）

AI4Drawing 是面向科研结构图的“草稿生产引擎”：以可编辑资产和复刻说明书交付高质量草稿，并用约束验证确保结构可靠，显著降低从想法到终稿的成本。

4.2 核心价值（对用户）

- 从“0 到可讨论草稿”耗时大幅降低。
 - 从“草稿到终稿”改动更少、更一致：有风格表、有图标与配色清单、有结构校验。
 - 降低风险：关键链路缺失、孤立节点、命名不一致等错误在定稿前被显式暴露。
-

5. 竞争与替代（不做泛泛而谈，明确对标点）

用户的替代路径通常是：

- 传统工具手工画（可靠但慢）
- LLM 生成 Mermaid/PlantUML 再手改（快但容易出错、缺少一致性与规范交付物）
- AI 图工具生成后编辑（已有产品覆盖“单图生成+编辑”能力）

因此 AI4Drawing 的差异化必须落在：

1. 科研套件化：多幅图协同生成与一致性管理
 2. 标准化交付物：可编辑源文件 + 复刻元信息包 + 校验报告
 3. 可靠性闭环：IR + 约束检查 + 修复策略，而非 LLM-only
-

6. 产品原则（设计约束）

1. **可控优先于华丽**: 宁可输出结构清晰的草稿，也不输出不可编辑的“漂亮图”。
 2. **确定性交付**: 输出必须可编译/可导入；任何不确定点必须标注并要求人工确认。
 3. **一致性是刚需**: 跨图风格表/符号表是系统一等公民。
 4. **以约束收敛幻觉**: LLM 是候选生成器，最终正确性由检查器裁决。
 5. **聚焦高频图类**: V1 只做最常见的 2-3 类图，做到最好。
-

7. 核心功能需求 (V1)

7.1 输入 (Input)

- 文本输入（必选）：
 - 结构化要点（推荐）：模块列表、输入输出、关键链路、分组层级、对比项
 - 论文段落/基金段落（可选）：系统自动抽取关键实体与关系
- 风格输入（可选）：
 - 选择内置风格模板（论文简洁风/基金强调风/答辩汇报风）
 - 自定义调色板（限定 Token）与字体/线条规范
- 生成任务配置：
 - 需要生成的图类型（方法框架/系统架构/实验流程）
 - 图数量（1-N）
 - 语言（中文/英文，允许双语标签）

7.2 输出 (Output) — 标准交付物（必须全部具备）

对每幅图输出以下三件套：

A. 可编辑草稿源文件（至少一种格式，优先多格式）

- draw.io (diagrams.net) XML: 直接导入编辑
- PPTX (shapes) : 便于答辩与基金材料

- Mermaid / PlantUML：便于版本控制与快速改动
- SVG（分组/层级保持）：便于在矢量编辑器中调整

B. 复刻元信息包 (Replication Pack)

- 图标清单：每个模块使用的图标类别/名称（或占位符规范）
- 调色板 Token：主色/辅助色/风险色/验证色等，及其使用规则
- 字体与字号：标题/正文/注释层级
- 布局规则：网格、间距、对齐、分栏、分组边框样式
- 语义映射表：概念 → 符号/颜色/形状（例如“风险/告警”统一用风险色+虚线框）

C. 结构正确性验证报告 (Validation Report)

- 语法可用性：是否可渲染/可导入
- 结构完整性：必备节点/边是否齐全，是否存在孤立节点
- 逻辑约束：DAG/循环、输入输出闭合、分组层级合法
- 一致性检查：跨图命名/符号/颜色是否一致
- 风险点列表：需要人工确认的歧义处（明确标红，不擅自编造）

7.3 生成与迭代（核心交互）

- 一次生成多幅图：同一项目下生成“图套件”
 - 图内迭代：
 - 修改需求要点 → 重新生成候选 → 校验 → 输出更新
 - 跨图一致性管理：
 - 全局“符号表/风格表”可锁定
 - 概念字典（Glossary）统一命名与缩写规则
-

8. 系统设计（技术方案与关键组件）

8.1 核心架构（LLM-only 不成立的工程解法）

采用“候选生成 + 约束验证 + 修复闭环”：

1. 需求解析器：从用户文本提取实体、关系、层级与约束
2. IR 生成器：输出统一的中间表示（Diagram IR）
3. 编译器：IR → draw.io XML / PPTX / Mermaid / PlantUML / SVG
4. 约束检查器（Checker）：对 IR 与导出结果做一致性与正确性校验
5. 修复器（Repair）：对不通过约束的点做最小修改或提出人工确认项

8.2 Diagram IR（中间表示，产品壁垒之一）

IR 至少覆盖：

- Node: id、label、type (process/data/store/decision/monitor...) 、icon_token、style_token、ports
- Edge: src、dst、edge_type (data/control/feedback/exception) 、label、style_token
- Group/Layer: 层级/泳道/子系统边界
- Layout: 网格/对齐锚点/相对位置约束（不追求像素级，但要可稳定渲染）
- Global Tables：
 - StyleSheet (调色板、线型、字号)
 - SymbolTable (概念→符号映射)
 - Glossary (命名/缩写/中英对照)

8.3 约束系统（确保“幻觉收敛”）

约束分层：

- 语法约束：导出格式可渲染/可导入
- 结构约束：无孤立节点；边引用合法；关键链路存在

- 逻辑约束：层级合法；输入输出闭合；循环关系显式标注
- 一致性约束：跨图同概念同符号/颜色；术语统一
- 风格约束：颜色必须来自 token；字体字号符合模板

输出策略：

- 通过：直接交付
 - 不通过：自动修复（最小编辑距离），若仍不通过 → 标注人工确认项并交付“可用但带风险”的版本
-

9. 关键指标 (KPI/度量体系：价值必须可量化)

9.1 产品价值指标（对用户）

- Time-to-final：从需求到终稿时间减少比例（目标：显著降低）
- 编辑距离：草稿 IR → 专家终稿 IR 的最小修改代价
- 结构错误率：关键链路缺失/孤立节点/命名漂移等错误频次
- 跨图一致性：同一概念符号一致率、命名一致率、颜色一致率
- 可复刻元信息完整度：复刻包字段覆盖率与可用性评分

9.2 系统质量指标（对工程）

- 可渲染/可导入通过率（按输出格式分别统计）
 - Checker 拦截率（发现问题比例）与误报率
 - 修复成功率（无需人工介入即可通过约束的比例）
 - 生成稳定性（同输入重复生成差异度）
-

10. 风险评估与应对（必须落实到设计）

风险 A：用户认为“草稿不值钱”

- 应对：

- 强制交付 “可编辑源文件 + 复刻元信息包 + 校验报告”
- 提供与传统工具深度兼容（draw.io/PPTX 优先），让用户是“改稿”而不是“重画”
- 用 KPI 展示节省的时间与编辑距离

风险 B：被现有通用 AI 图工具快速覆盖

- 应对：
 - 把壁垒放在“科研图套件化生产”：跨图一致性、符号表、风格表、约束报告
 - 提供科研模板与概念字典（论文/基金常见结构）
 - 输出可复刻包形成事实标准（不仅是生成结果）

风险 C：范围过大导致交付失败

- 应对：
 - V1 聚焦 2–3 类最高频图（方法框架、系统架构、实验流程），做到最好
 - 先做“80/20”结构：分组/对比/异常处理等高价值能力优先
 - 后续按学科/图类增量扩展
-

11. 版本规划（学生团队可落地路线）

V0（MVP，目标：可用）

- 支持 2 类图：方法框架图、实验流程图
- 输出：Mermaid + draw.io XML（至少二选一可编辑）
- 提供基础复刻包（图标 token、调色板、字号、布局规则）
- 提供基础校验：可渲染、孤立节点、关键链路存在

V1（目标：形成差异化）

- 增加第 3 类图：系统架构图（分层+数据流/控制流）
- 引入全局 StyleSheet + SymbolTable + Glossary

- 跨图一致性检查与锁定
- 修复器：不通过约束时自动最小修改并输出变更记录

V2 (目标：可商业化增强)

- PPTX shapes 高质量导出（面向基金/答辩）
 - 团队协作：项目空间、权限、版本历史、符号库共享
 - 扩展模板库与学科包（按领域增量）
-

12. 商业化与开源双线策略（与进攻点一致）

12.1 开源版（Open）

- 核心：Diagram IR + 编译器 + Checker 基础能力
- 模板与示例：高频图类模板仓库（可社区共建）
- 目标：形成标准、扩大生态、促进学术产出与引用

12.2 商业版（Pro）

- 增强：高质量 PPTX/draw.io 导出、团队协作、企业/实验室私有化、更多模板包
 - 服务：技术支持、模板定制、合规与数据隔离
 - 收费逻辑：按席位/按团队/按导出高级格式与模板包
-

13. 学术研究输出计划（从产品楔子自然生长）

围绕“草稿引擎 + 可验证结构 + 可复刻交付物”，可形成可发表问题：

1. 科研结构图的 IR 设计与多格式编译：从文本到可编辑草稿资产
2. 约束满足式生成：LLM 候选 + Checker + Repair 的可靠性闭环
3. 多幅图一致性：符号表/风格表跨图维护与一致性度量
4. 草稿可用性评测：结构正确性、编辑距离、Time-to-final 等指标体系
5. 数据集/基准：以 IR/复刻包为标注目标，而非仅 PNG 图像

14. 验收标准 (Definition of Done)

V1 验收必须满足：

- 能从同一项目输入生成至少 3 幅图（方法/架构/流程）并保持风格一致
 - 每幅图输出三件套：可编辑源文件 + 复刻包 + 校验报告
 - 校验报告能拦截并标注关键结构问题（孤立节点、缺失链路、命名漂移等）
 - 用户可在 draw.io 或 PPT 中直接打开并编辑，完成定稿所需改动量显著减少（用编辑距离或用户测试时间验证）
-

附录 A：复刻元信息包字段（建议最小集合）

- palette_tokens: {primary, secondary, risk, verify, accent, text}
- typography: {font_family, title_size, body_size, note_size}
- icon_tokens: [{concept, icon_id, usage_rule}]
- layout_rules: {grid, node_padding, group_margin, alignment, arrow_style}
- glossary: [{term, abbr, zh/en, canonical}]
- symbol_table: [{concept, shape_token, color_token, icon_token}]

附录 B：基础约束清单 (V1)

- Renderable：目标格式可渲染/可导入
- NoOrphan：无孤立节点
- EdgeValidity：边引用合法
- RequiredChain：关键链路存在（输入→核心处理→输出）
- GroupHierarchy：分组层级合法，不交叉冲突
- TokenOnly：颜色/字体/线型来自 token 表
- CrossDiagramConsistency：跨图同概念符号/命名一致（若锁定）