

AI4Drawing项目统筹

前期工作

理工科常见绘图类型调研

- 将“常见绘图类型”按**语义结构**而不是按学科命名：
 - Pipeline/流程（线性、分支、循环、异常处理）
 - Layered Architecture/分层架构（层级+跨层依赖）
 - System Block Diagram/模块图（分组+接口+数据/控制流）
 - Comparison/对比图（baseline vs ours，多列对齐）
 - Experiment Design/实验路线（数据—训练—评测—消融）
- 每类图抽取“最小结构要素”：节点类型、边类型、分组/泳道、典型布局。

关键产出物

- 《Top 图类定义表》：每类图的结构要素、常见子结构、典型模板数量（建议 5-10 个模板/类）。
- 《V1 覆盖范围决策》：明确 V1 只做 2-3 类（建议：方法框架图、系统架构图、实验流程图）。

传统科研绘图工具

- 不仅盘点工具，还要拆解科研人员的**真实定稿工作流**：
 - 他们用什么工具起草？用什么工具定稿？中间怎么迁移？
 - 最痛的是哪一步：布局、对齐、配色、图标、文字、版本迭代、协作？
- 明确你要适配的“定稿工具”优先级：
 - 推荐优先：**draw.io (XML)**、**PPT (shapes)**（覆盖率高、上手低）
 - 次优先：SVG 分组、Mermaid/PlantUML（版本控制友好）

关键产出物

- 《定稿工具优先级与格式规范》：每种导出的“必须可编辑特性”（如分组层级、文本可编辑、箭头连接保持）。
- 《草稿→终稿迁移路径图》：明确你的工具插在哪一环节，替代哪段耗时。

新兴绘图工具

- 对每个新兴工具做“能力矩阵”：

- 单图生成质量、可编辑性、跨图一致性、导出格式、可控性、失败模式、定价/部署。
- 重点整理“失败模式谱系”：
 - 关键链路缺失、概念漂移、未定义节点、跨图不一致、样式 token 越界等。这些将直接转化为你 V1 Checker 的约束清单。

关键产出物

- 《竞品能力矩阵（可复用表格）》
- 《失败模式 Taxonomy + Checker 约束清单 V0》

与本项目高度相关的已有产品调研

- 明确你与“AI 图工具（如 DiagramGPT 一类）”的差异不是“我也能生成”，而是：
 - a. 图套件（多幅图）一致性
 - b. 标准化交付物（可编辑源文件 + 复刻包 + 校验报告）
 - c. 约束闭环可靠性（Checker/Repair）
- 产出“反向需求”：哪些功能竞品做得很好但你不做（例如终稿精美渲染），避免范围失控。

关键产出物

- 《差异化声明（Differentiation Statement）》：一页纸讲清楚你是谁、不是谁。
- 《不做清单（Not-to-do List）》：防止团队漂移。

产品定位

- 把价值主张固化成可量化指标（后续做用户验证与商业化都需要）：
 - Time-to-final 降幅（目标： $\geq 40\%$ 或按阶段设定）
 - 编辑距离（草稿→终稿的最小修改数）
 - 结构错误率（Checker 拦截后显著下降）
 - 跨图一致性（同概念符号一致率）

关键产出物

- 《V1 验收指标与阈值》：没有阈值就无法推动工程收敛。
- 《用户价值计算器（内部）》：用任务耗时估算 ROI。

用户研究与样本收集

- 你需要的不是“大而全访谈”，而是“任务样本库”
 - 目标：收集 30-100 份真实科研图需求（哪怕是粗文本 + 参考图），覆盖三类图。

- 来源：实验室内部、同学论文、公开论文中的方法图/架构图（注意版权：只抽结构，不直接复用图像）。

关键产出物

- 《图需求样本库（结构化）》：每条包含输入文本、期望结构要素、目标图类、难点标签。

Diagram IR 草案与导出后端选择

- 你项目的技术壁垒与工程可控性来自 IR。
 - 建议 1 周内输出 IR schema V0（节点/边/分组/样式 token/术语表）。
 - 同时确定 V1 只打通 2 个后端：**draw.io XML + Mermaid 或 PPTX**。

关键产出物

- 《IR Schema V0》
- 《后端导出规格与最小可编辑要求》

Checker/Repair 约束闭环 PoC

- 你已经认定 LLM-only 不可靠，Checker 是系统的“裁决器”。
 - 先不追求全面，只做 10-15 条最关键约束：
 - 无孤立节点
 - 关键链路闭合（输入→核心→输出）
 - 边引用合法
 - 分组层级合法
 - token 来自调色板
 - 跨图术语一致（若锁定）
 - Repair 先做“最小修复”：补边、合并同名概念、重命名统一、补缺失模块占位等。

关键产出物

- 《Checker 规则 V0》+ 《修复策略 V0》
- 《PoC Demo》：从一段输入生成 IR → Checker 报告 → Repair → 可导入 draw.io

建议节奏

第 1 周：范围锁死 + 样本库

- 决定 V1 图类（2-3 类）
- 建样本库（≥30 条）
- 输出竞品矩阵与失败模式清单

第 2 周：IR V0 + 导出后端 V0

- IR schema V0
- 打通 draw.io XML 导出（或 PPTX shapes）
- 能生成“可编辑草稿”而不是图片

第 3–4 周：Checker/Repair PoC + V0 内测

- Checker 10–15 条约束
- Repair 最小修复闭环
- 内测：同学/实验室 5–10 人任务测试，测 Time-to-final 与编辑距离