

pptx-wzq 技术分析报告

PPT 多模态教学知识库自动化构建系统 • 版本 2.4.0 • 2026-08-24

定位

把任意 PPT 课件解构为「多模态教学知识库」：文本 / 公式 / 图块（可视逻辑块）三路抽取 + AI 解读 + 教材级文案，输出结构化 JSON / Markdown / 渲染图三件套。

技术栈

python-pptx 生态 + OOXML 原生解析 (lxml/ElementTree) + DeepSeek-V4-Flash (语义解读/文案) + qwen3.7-plus (VLM 兜底) + PowerPoint COM 渲染。

交付物

sources/ (XML 段 • 矢量 • 资源图) + images/ (渲染图) + visual_blocks.json + visualBlock_text_binding.json + textbook.md + captions.md。

1. 系统定位与核心能力

pptx-wzq 面向高校教学场景，把「讲稿 PPT」自动转成可检索、可复用、可二次出版的多模态知识库。核心能力：

图块（可视逻辑块）优先：以「组合即图块」为第一原则，把作者用 PPT 组合工具绘制的逻辑图/示意图整体保留，而非拆散成碎片。

源语言级解构：grpSp 组合保留原生 XML 段（p:/a:/r: 前缀），Visio/vsd 剥离原生文件，公式转 LaTeX——信息不降维。

模型按块路由：有 XML 的块 → DeepSeek 读 XML（公式转 LaTeX）；纯像素图块 → qwen VLM 兜底；DeepSeek 空响应自动降级。

教材级文案：整个 PPT 视为一部教材，DeepSeek 自主分章分节（一节可含多页），每页内容标注章节，500 字为限（不足扩写、超出直出整理）。

生命周期自管：全流程成功即清理过程文件；中断保留断点、二次运行自动接续。

2. 技术架构

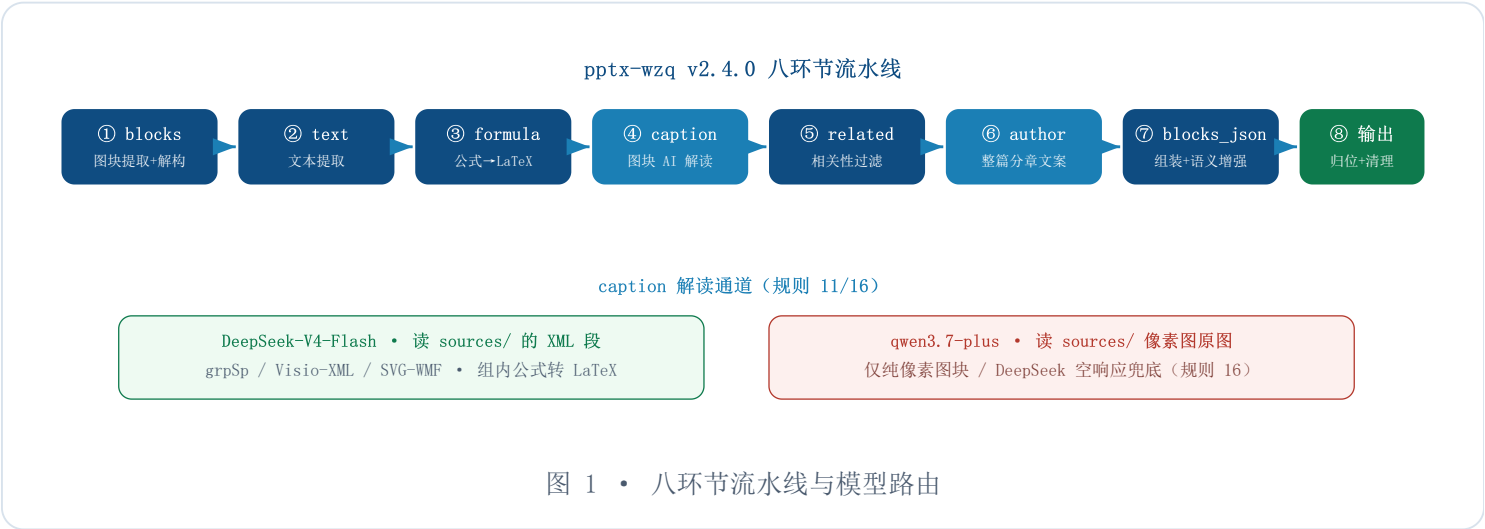
2.1 模块划分

模块	职责	Token
<code>extract_pptx_images.py</code>	OOXML 原子对象提取：图片/形状/组合 (grpSp)/表格/公式 (OLE • OMML)/图表；grpSp XML 段原生提取；srcRect 裁剪；PowerPoint COM 整页渲染	0（本地）
<code>extract_texts.py</code>	页面文本提取（in_group 标记、组内文字标注「图块内文本」）、表格 Markdown 化	0
<code>visual_blocks.py</code>	单阶段确定性拆块（grpSp→group / visio / raster≥20% / 矢量）、块渲染、语义增强（DeepSeek）、跨模态关系	DeepSeek
<code>cli_blocks.py</code>	blocks/caption 命令：XML 段导出（sources/）、rldimg 资源复制、caption 路由（DeepSeek XML / qwen 兜底）、binding 导出、captions.md	DeepSeek+qwen
<code>cli_author.py</code>	教材文案：整篇分章分节、每页标注章节、500 字为限（扩写/直出整理）、自动分批	DeepSeek
<code>cli_related.py</code>	块相关性过滤（剔除 logo/作者/装饰块）+ 审计 json	DeepSeek
<code>cli_paser.py</code>	总编排器：8 环节流水线、断点续传（state.json）、产物归位、成功即清理	—
<code>cli_text/formula/img</code> 等	叶子命令（可单独调用）；img 已从流水线移除（并入 blocks 自举）	0

2.2 命令体系（10 个 console_script）

pptx-paser 总编排器（一条命令跑完 8 环节） pptx-blocks 图块提取 + 解构 + caption（--caption-sources 模式） pptx-text 文本提取 pptx-formula 公式提取 pptx-caption 解读 pptx-related 相关性过滤 pptx-author 教材文案 pptx-bind 图文绑定 pptx-html 教材 HTML pptx-deck Deck 生成

3. 工作流程（8 环节）



3.1 各环节要点

- ① blocks: 单阶段确定性拆块 (grpSp→group / visio / 像素图≥20% / SVG-WMF 矢量; 无自由聚类); grpSp XML 段每组合一个文件导出 sources/; rldimg 资源图落盘; PowerPoint COM 整页渲染 + bbox 裁剪出块 PNG (images/). 本地 规则 1-9/10/12
- ② text: 页面正文提取 (排除页眉页脚/母版固定文本); 组内文字标「[图块内文本]」不混入正文; 表格输出 Markdown 表格。 本地
- ③ formula: 非组合公式 → LaTeX (OMML 原生 / MTEF / OCR 三级); 组合内公式排除 (随 XML 段由 DeepSeek 转 LaTeX)。 本地
- ④ caption: 按 sources/ 文件名顺序解读——.xml → DeepSeek 读 XML (超长本地压缩 + 3 次重试 + 规则模板兜底); .png → qwen 读图。 DeepSeek+qwen
- ⑤ related: DeepSeek 判断块与页面正文相关性, 剔除 logo/作者/装饰块, 输出 related_filter.json 审计。 DeepSeek
- ⑥ author: 整篇分章分节 (# 第X章 → ## 第X节, 一节可含多页), 每页内容首行标注所属章节; 原文 ≤500 字扩写至不少于 500 字, >500 字直出整理 (不改原意、不增字数)。 DeepSeek

⑦ blocks_json: 组装 visual_blocks.json (v2.0 schema) + DeepSeek 语义增强
(expression_goal/role/features/vlm_caption/teaching_use) + 跨模态关系; 导出
visualBlock_text_binding.json; 归位 images/ + sources/ + captions.md。 [DeepSeek](#)

⑧ 输出: 成功即删除全部过程文件 (只留交付物); 中断保留断点 (state.json) + doc_md5 换源检测, 二次运行自动接续缺失步骤。 [本地](#)

4. 图块识别规则（16 条定稿）

4.1 识别层（1-9）——「组合即图块声明」

#	规则	实现
1	一个 grpSp 组合即是一个图块，组合内一切内容读取为该图块内容	kind="group" 原子对象，children 递归收编（含文字/图片/公式/OLE）
2	嵌套组合不单独提取	嵌套 grpSp 仅作外层 children（保留分层结构）
3	OLE Visio/vsdx 独立成块（除非在组合内）	kind="visio" 分支（v2.0 补上，此前静默丢弃）
4	非组合像素图单独成块；重叠其上的文本并入	raster 独立块 + 重叠文本并入
5	组合内公式作块内容；非组合公式独立提取	formula 步骤排除 in_group；组内公式由 DeepSeek 转 LaTeX
6	首页（题目页）/尾页（致谢页）图块舍弃	skip_cover_pages：第 1 页与末页整页不产块（文本/公式仍提取）
7	非组合像素图面积 < 整页 20% 舍弃	raster_min_area_ratio=0.20；组合内豁免
8	表格仍读取为表格（文本/Markdown）	表格移交 text 步骤，输出 Markdown 表格
9	grpSp 内 srcRect（裁剪显示）须全程一致	children 携带 src_rect + 资源裁剪落盘 + 描述标注

4.2 解构/解读层（10-16）

#	规则	实现
10	grpSp 保留整段 XML，页标记，每组合一个独立 .xml 文件存 sources/	sources/slide_{页}_{块id}_grp.xml（原生 p:/a:/r: 前缀）
11	grpSp 块 caption 用 DeepSeek 读 XML；组内公式转 LaTeX 融入	_ds_read_xml + 超长压缩 + 重试
12	grpSp 块用 PowerPoint 渲染 PNG 存 images/；PNG 不送 qwen	COM ExportAsFixedFormat PDF + PyMuPDF（仅 PowerPoint）

13	Visio/vsdX 可剥离 → .vsdx 存 sources/ + 渲染 PNG 存 images/	原生文件剥离 + 渲染
14	Visio 不可剥离 → XML 段，同 grpSp 处理	sources/slide_{页}_{块id}_ole.xml
15	SVG/WMF 等矢量同 Visio: 尽量 XML 段，不行才 PNG	sources/slide_{页}_{块id}_vec.xml
16	仅无法用 DeepSeek 解读 XML 的块（纯像素图）才送 qwen	模型按块路由 + 空响应降级链

5. 输出文档体系

5.1 结果目录结构（成功运行后）

```
<名>_result/
├─ sources/ # 图块源资源（解读唯一输入源）
│  ├─ slide_07_blk_01_grp.xml # grpSp XML 段（页注释 + 原生前缀，规则10）
│  ├─ slide_05_blk_02.vsdX # Visio 可剥离（规则13）
│  ├─ slide_05_blk_03_ole.xml # Visio 不可剥离 → XML（规则14）
│  ├─ slide_08_blk_04_vec.xml # SVG/WMF 矢量 XML（规则15）
│  ├─ slide_19_blk_01.png # 像素图块原图（qwen 解读输入，规则16）
│  └─ rldimg/ # grpSp XML 内 r:embed 引用的资源图片
│     └─ slide_07_blk_01_image4.png
├─ images/ # 块渲染图（PowerPoint 渲染，仅供人阅览）
│  └─ slide_07_blk_01.png
├─ <名>_visual_blocks.json # 核心结构化（v2.0 schema）
├─ <名>_visualBlock_text_binding.json # 块↔文本图文关联（v1.0）
├─ <名>_textbook.md # 教材文案（篇→章→节→页）
├─ <名>_captions.md # 块解读（绑定 sources/ 文件名）
├─ <名>_texts.md / _text_entries.json # 文本清单（组内文字标[图块内文本]）
├─ <名>_formulas.md # 非组合公式（LaTeX）
└─ <名>_related_filter.json # 相关性过滤审计
    （v2.0 起成功即清理：无 过程文件/ 目录；中断时保留断点供续传）
```

5.2 文件格式总表

文件	格式	说明
<名>_visual_blocks.json	JSON pptx_multimodal_slide_v2.0	slide_info / textual_content / visual_blocks[] / cross_modal_relations[] / summary
<名> >_visualBlock_text_binding.json	JSON pptx_visual_block_text_binding_v1.0	cross_modal_relations 独立视图: text_anchor /

		relation_type / semantic_link; summary 含 sources_total / xml_sources_total / rldimg_total
<code>sources/slide_{页}_{块id}_grp.xml</code>	PPTX 原生 XML 子集	首行 `<!-- 第 N 页 grpSp: 名称 -->` + `<p:grpSp>...</p:grpSp>` 原始段（含文字/公式 oMath/填充/srcRect）
<code><名>_textbook.md</code>	Markdown	`# 教材` → `# 第X章 章 名` → `## 第X节 节名` → `## 第 N 页`（首行 `> 所属章节: ...`）
<code><名>_captions.md</code>	Markdown	每条绑定 sources/ 文件 名 + 解读通道标注 （DeepSeek-XML / qwen- VLM / 规则兜底）
<code><名>_texts.md</code>	Markdown 表格	TXT 编号 / 类型（标题 • 内容 • 表格行 • [图块内文 本]）/ 文本 / 坐标
<code><名>_formulas.md</code>	Markdown	非组合公式 LaTeX 汇总

5.3 VisualBlock 内部格式（visual_blocks.json 的块对象）

```
{ "block_id": "blk_01", "page": 7, "block_type": "战略管理概念框架", "bbox": { "x": 142.3, "y": 226.4, "w": 1026.2, "h": 400.2}, "z_index_range": [12, 45], "is_single": false, "text": "战略哲学 商道 天道 人道 ...", "assets": { "xml_source": "./sources/slide_07_blk_01_grp.xml", "raster_png": null, "rldimg": ["../sources/rldimg/slide_07_blk_01_image4.png", ...] }, "internal_structure": {"nodes": [...], "edges": [...]}, "semantic_description": { "block_type": "战略管理概念框架", "expression_goal": "展示战略管理概念框架的核心逻辑", "expression_role": "将抽象概念通过战略哲学/商道/天道/人道具象化...", "expression_features": ["概念框架", "层次结构", "关系图"], "vlm_caption": "该图块以“战略哲学”为中心...", "teaching_use": "教学辅助图示", "formula_latex": "", "caption_source": "deepseek_xml" }, "member_obj_ids": [...], "vector_resources": [] }
```

6. 输入 PPTX 预处理要求

核心思想：用 PPT 自带工具给解析器打「块边界」标注。组合的数量 = 该页图块数量的上限基准，可据此验收。预处理不是必须的（无组合也能跑），但组合能让图块提取完全确定、可回归。

对象	预处理操作	解析器行为
逻辑图/示意图 （要整体成块）	用「开始 → 排列 → 组合」（Ctrl+G）把底图+文字框+箭头合成一个组合	整个组合 = 1 个 group 块，XML 段导出 sources/，渲染图存 images/
嵌套组合	有意为之才用：嵌套 = 外层块的子结构；想分开就把组合移出外层	嵌套组合并入外层 children，不单独成块
Visio / vsdx 工程图	不要组合进其他形状（否则被吞并）；保持独立 OLE 对象	独立成 visio 块：可剥离 → .vsdx；不可剥离 → XML 段
公式	正文行内公式保持独立（不组合）→ 自动进 formulas.md；想并入图块就把公式移进组合	组合内公式随块转 LaTeX；非组合公式独立提取
首页/尾页	无需操作：默认按页序跳过第 1 页与末页的图块	封面/致谢不产块（文本/公式仍提取）
小像素图（装饰图 标）	小于整页 20% 的图默认舍弃；想保留 → 组合进相邻图形	组合内小图豁免；非组合小图丢弃并写入审计
表格	无需操作：表格始终按表格读取	输出 Markdown 表格（texts.md），不产块
带裁剪的图片 （srcRect）	无需操作（自动保持显示一致）	元数据/资源/渲染/描述全程按裁剪显示
环境	本机安装 Microsoft Office（渲染通道仅 PowerPoint）+ 配置 DASHSCOPE_API_KEY（qwen）、DEEPSEEK_API_KEY	无 Office 渲染降级（无块图）； 无 Key 跳过对应解读

7. 生命周期与可靠性

7.1 成功即清理

全流程成功后，中间产物（by_page / atomic_objects.json / manifest / 各步骤工作目录）全部删除，结果目录只留交付物。
删除失败（如文件占用）打印警告而非静默，提示可手动清理。

7.2 中断续传

- 中断时保留 work 目录 + state.json（步骤状态机 pending/running/partial/done/failed）。
- 再次运行：已完成步骤跳过，缺失步骤自动接续；doc_md5 检测换源 → 全量重跑。
- author 步骤支持缺失页补跑（--pages）。

8. 版本演进

版本	里程碑
1.5.0	六步管线重构（img 并入 blocks 自举）、三件套交付物、版本统一
2.0.0	16 条规则落地：组合即块、单阶段确定性拆块、XML 段导出、DeepSeek 读 XML caption 路由、PowerPoint 渲染、_organize 修复（rldimg 归位 + 成功即清理）
2.1.0	captions 绑定 sources/ 源文件 + 解读通道标注；每页扩写 500 字
2.2.0	textbook 规则改 500 字为限：不足扩写、超出直出整理（_tidy_direct）
2.3.0	Author 整篇自主分章分节（一节可含多页），每页标注章节
2.4.0	本报告（技术分析/流程/格式/预处理）随安装包分发

附注：本报告所有字段与格式均来自 pptx-wzq 真实产物（visual_blocks.json / binding / textbook.md / captions.md / sources/），非虚构。安装后可通过 `python -c "import pptx_wzq; print(pptx_wzq.__version__)"` 确认版本。