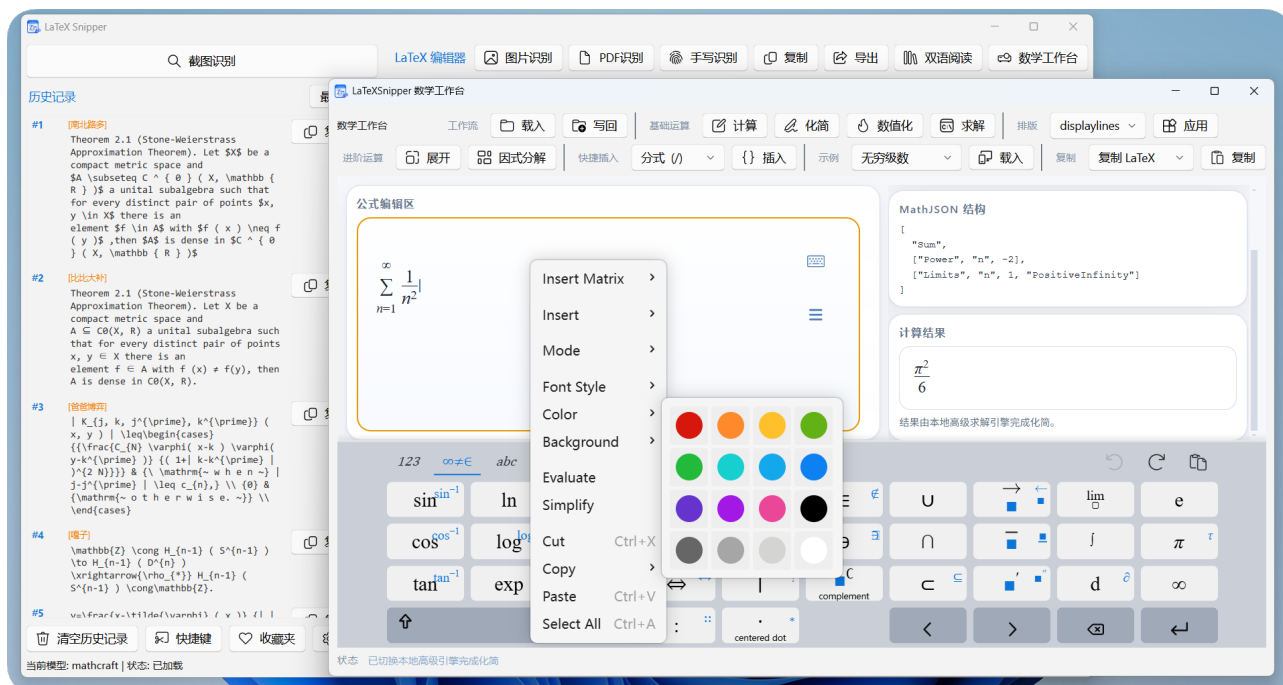


LaTeXSnipper

用户手册

适用于 v2.4.0-LTS | 长期支持版



目录

第一卷 • LaTeXSnipper 用户指南

- 开始使用前（必读）
- 主窗口功能入口与识别流程
- 本地模型（MathCraft ONNX）相关问题
- 外部模型相关问题
- 安装和环境问题
- 网络与更新问题
- PDF 与识别效果问题
- 使用技巧与功能问题
- 平台特定问题（Linux / macOS）
- 与其他软件冲突
- 从源码运行与开发者常见问题
- 如何有效反馈 Bug

第二卷 • Office 加载项指南

- 加载项介绍与系统要求
- 安装与注册
- Word 功能详解
- PowerPoint 功能详解
- 公式编辑器与侧边栏
- 设置与帮助

- 常见问题排查

第三卷 • 移动端使用指南

- LaTeXSnipper Mobile 介绍
- 功能与页面结构
- 与桌面版的差异
- OCR 识别与输入方式
- 公式编辑器与键盘策略
- 导出与分享
- 移动端常见问题

第四卷 • MathCraft 内部模型介绍

- MathCraft OCR 项目介绍
- 环境变量设置指南

第一卷 • LaTeXSnippet 用户指南

开始使用前（必读）

安装后第一步：运行依赖向导

LaTeXSnippet 首次启动或检测到关键依赖缺失时会弹出“依赖向导”。这不是可有可无的步骤——向导负责安装和配置内置 MathCraft OCR、PDF/Pandoc 等运行依赖层；外部模型服务本身仍需要用户另行部署或配置。

⚠ 常见错误：跳过向导后直接使用内置识别

如果依赖层未完整安装，内置 MathCraft 截图识别、PDF 识别、手写识别和 Pandoc 导出可能不可用，或在首次调用时进入修复流程。如果只使用外部模型，可以跳过安装并进入应用，但仍需要在设置页完成外部服务配置并通过连接测试。向导只管理 Python 依赖层，不会安装或卸载系统软件包（如 apt/brew 等）。

如果向导本身运行失败：

- Windows：确保以普通用户（非管理员）运行，且杀毒软件未拦截
- Linux：确保 `python3 -m venv` 可用（Debian/Ubuntu 通常需要 `python3-venv`）
- macOS：确保有可用的 Python $\geq 3.10, < 3.13$ ，推荐 Homebrew Python 或 python.org 官方 3.11/3.12 安装包

软件的基本 workflow

！了解这个流程可以避免 80% 的困惑

截图 → 识别 → 编辑/预览 → 导出，具体来说：

1. **主窗口截图**：按下截图快捷键（默认见设置），框选公式区域
2. **识别结果**：自动出现在主窗口列表，双击可查看 LaTeX 源码和渲染预览
3. **编辑区**：主窗口右侧是 LaTeX 编辑器，支持实时预览渲染
4. **数学工作台**：点击“数学工作台”按钮打开独立窗口，会自动载入主编辑器中的公式。在工作台中可以编辑公式、进行数值计算或表达式化简，完成后可写回主编辑器
5. **手写识别**：从主窗口打开手写窗口，内置识别固定使用 MathCraft 混合模式，手写文字和公式会一起识别
6. **导出**：右键或菜单可选择 20 种桌面端导出格式（部分需要 Pandoc）

日志文件在哪里（关键!）

遇到任何问题时，第一步永远是 **查看日志**：

- **Windows**： `%USERPROFILE%\latexsnippet\logs\`
- **Linux**： `~/.latexsnippet/logs/`
- **macOS**： `~/Library/Logs/LaTeXSnippet/`
- **崩溃日志**：同目录下的 `crash-native.log`

✓ 建议

反馈 Bug 时把整个 logs 目录打包发过来，不要只截图报错弹窗。

主窗口功能入口与识别流程

主窗口按钮分别做什么

当前主窗口左侧是截图和历史区，右侧是 LaTeX 编辑、预览和工具入口：

- **截图识别**：使用全局快捷键或按钮进入框选截图识别。默认快捷键 Windows/Linux 为 `Ctrl+F`，macOS 为 `Command+F`，可在主窗口“快捷键”按钮中修改。
- **图片识别**：选择本地图片文件并走当前识别模型。
- **PDF 识别**：选择 PDF 后按页渲染识别，可输出 Markdown 或 LaTeX 文档。
- **手写识别**：打开独立手写画布，写完后自动触发识别，结果可复制或写回主编辑器。
- **数学工作台**：将主编辑器中的公式载入 MathLive 工作台，进行计算、化简、展开、因式分解、求解等操作。
- **复制 / 导出**：复制当前编辑器内容，或通过统一导出菜单导出为 LaTeX、Markdown、MathML、HTML、OMML、SVG 以及 Pandoc 扩展格式 (Word、ODT、PowerPoint、EPUB、PDF、HTML 独立页、Typst、纯文本)。
- **收藏夹 / 历史记录**：历史项和收藏项都支持复制、编辑、导出、重命名和删除。

桌面端当前共有 **20 种导出格式**：

- **内置格式 (无需 Pandoc)**：LaTeX 行内、LaTeX display、LaTeX equation、Markdown 行内、Markdown 块级、MathML、MathML .mml、MathML <m>、MathML 属性形式、HTML、Word OMML、SVG Code
- **Pandoc 扩展格式**：Word .docx、ODT .odt、PowerPoint .pptx、EPUB .epub、PDF .pdf、HTML 独立页 .html、Typst .typ、纯文本 .txt

托盘或菜单栏状态菜单还提供 **截图屏幕模式**：自动模式会按鼠标释放点选择屏幕，也可以固定到某一块显示器。多屏截图位置不对时，优先检查这里。

设置页当前业务逻辑

设置页的“选择识别模型”只有两类入口：

- **内置模型**：使用 MathCraft OCR。本地识别类型可选 **公式、混合 (文字+公式)、纯文字**。
- **外部模型**：连接 OpenAI-compatible、Ollama 或 MinerU 服务。推荐预设包括 GLM-OCR、PaddleOCR-VL、Qwen2.5/Qwen3-VL、Ollama Vision 和 MinerU Native。

外部模型的 **输出偏好** 只影响普通图片、截图和手写识别；PDF 入口会单独询问导出格式和 DPI。填写 **自定义提示词** 后，自定义提示词优先级最高，会覆盖图片、截图、手写以及 OpenAI-compatible / Ollama PDF 识别的默认模板。

设置页还包含外观主题、公式渲染引擎、LaTeX 路径验证、更新检查、启动时显示日志窗口、依赖管理向导、打开 MathCraft 缓存目录等入口。MathCraft 依赖统一由主依赖环境管理，设置页不再提供单独的模型下载按钮。

PDF 识别流程

点击 **PDF 识别** 后，程序会按当前模型执行：

- 当前为 **内置模型** 且不是混合模式时，会提示切换到 MathCraft 混合识别后继续。PDF 文档识别需要混合模式做文本和公式整理。
- 当前为 **外部模型** 时，必须先设置在设置中完成外部模型配置并通过连接测试。
- MinerU 原生协议会走文档解析模式，输出固定为 Markdown；其他模型会先询问输出 Markdown 还是 LaTeX。
- 程序会询问识别页码或连续范围，例如 5、3-7；默认先填入 1-5，避免大 PDF 一次性耗时过长。
- 程序会询问 PDF 渲染 DPI，范围为 90-300。外部模型默认 150 DPI，内置模型默认 200 DPI。

- 识别结果会打开独立的 PDF 结果窗口，可编辑、复制或保存。Markdown 文档保存时，如果结构化结果包含图片资源，程序会尽量把相关 assets 一起复制到保存目录。

本地模型（MathCraft ONNX）相关问题

✔ MathCraft ONNX 用户提示

应用会自动下载并 `warmup()`，如果 `warmup` 失败会尝试重新下载。但如果网络慢，下载模型可能很久。可以先用 `python -m mathcraft_ocr models check` 检查缓存状态。

模型下载失败（网络问题 / 代理 / 防火墙）

现象： 启动或首次使用时卡在 “model xxx downloading”，然后失败。

原因：

- 模型托管在 GitHub Releases 等境外服务器，国内用户可能无法直接访问
- 公司/学校网络有防火墙拦截
- 代理设置不正确导致 `urllib` 无法连接

解决：

- 检查网络是否能访问 GitHub
- 设置环境变量 `HTTP_PROXY` / `HTTPS_PROXY`（如果使用代理）
- 尝试切换网络（如手机热点）
- 删除 MathCraft 模型缓存目录下残留的不完整下载后重试：
 - ▶ Windows: `%APPDATA%\MathCraft\models\`
 - ▶ Linux: `${XDG_DATA_HOME:-~/.local/share}/LaTeXSnippet/MathCraft/models/`
 - ▶ macOS: `~/Library/Application Support/LaTeXSnippet/MathCraft/models/`

✔ 命令行诊断

运行 `python -m mathcraft_ocr models check` 可以查看模型缓存状态。运行 `python -m mathcraft_ocr warmup --profile mixed --provider auto` 可以预热混合识别链路；如果模型缺失或损坏，运行时会自动下载或修复缓存。

onnxruntime 安装不完整或版本不对

现象： 启动报 `failed to import onnxruntime` 或 `missing get_available_providers`。

原因：

- 可能安装了 `onnxruntime` 的命名空间包而非完整包
- `pip` 安装时网络中断导致包不完整
- CPU 版和 GPU 版冲突

解决：

- `pip uninstall onnxruntime onnxruntime-gpu` 全部卸载后重新安装
- 如需 GPU 加速: `pip install onnxruntime-gpu`
- 仅 CPU: `pip install onnxruntime`

CUDA / GPU 相关错误

现象： `warmup` 时报 CUDA 相关错误，如 `cuda wasn't able to be loaded`、`cublas` 错误等。

原因：

- 安装了 `onnxruntime-gpu` 但没装 CUDA / cuDNN

- CUDA 版本与 onnxruntime-gpu 不匹配
- 显卡驱动太旧
- 多 GPU 环境下默认选错了设备

解决：

- **如果不需要 GPU：** 设置环境变量 MATHCRAFT_FORCE_ORT_CPU=1 强制 CPU 模式
- **如果需要 GPU：** 确认 CUDA Toolkit 和 cuDNN 版本与 onnxruntime-gpu 匹配（查 onnxruntime 官方兼容表）
- 更新显卡驱动

模型缓存损坏（下载一半断电 / 杀进程）

现象： warmup 报 invalid protobuf、failed to load model、sha256 mismatch。

原因： 下载过程中断导致模型文件不完整；或者磁盘故障导致文件损坏。

解决：

- 删除对应模型目录：Windows 为 %APPDATA%\MathCraft\models\<model_id>\, Linux 为 \${XDG_DATA_HOME:- ~/.local/share}/LaTeXSnippet/MathCraft/models/<model_id>/, macOS 为 ~/Library/Application Support/LaTeXSnippet/MathCraft/models/<model_id>/
- 重启应用，会自动重新下载
- 或用命令行：python -m mathcraft_ocr models check 检查缓存状态

显存不足导致 ONNX 模型加载失败

现象： warmup 报 CUDA out of memory 或直接崩溃。

原因： GPU 显存不够。应用会根据显存选择批大小，但如果其他程序（浏览器、游戏）占了显存，剩余不够。

解决：

- 关闭其他占用 GPU 的程序
- 设置 MATHCRAFT_FORCE_ORT_CPU=1 强制使用 CPU

外部模型相关问题

根本没配置外部模型，直接点识别然后报错

现象：截图后识别失败，提示“模型名为空”或“外部模型地址为空”。

✗ 错误原因

外部模型客户端 `_validate_config()` 要求 provider 非 mineru 时 `model_name` 和 `base_url` 不能为空。
默认 `model_name` 就是空字符串，你需要手动填写。

解决：先去 **设置** → **外部模型**，至少填上 Base URL 和模型名（或选择预设），点“测试连接”通过后再使用。

Base URL 应该填服务根地址

现象：服务本身正常，但测试连接访问了重复路径，或提示接口路径不存在。

原因：LaTeXSnipper 会按协议自动拼接接口路径：

- Ollama: `<Base URL>/api/tags` 测试模型列表，`<Base URL>/api/chat` 执行识别
- OpenAI-compatible: `<Base URL>/v1/models` 测试模型列表，`<Base URL>/v1/chat/completions` 执行识别
- MinerU: `<Base URL>/<健康检查路径>` 测试连接，`<Base URL>/<解析接口路径>` 执行解析

解决：Base URL 只填服务根地址，例如 `http://127.0.0.1:11434`、`http://127.0.0.1:8000` 或服务商 HTTPS 根地址；不要手动追加 `/v1/`、`/api/chat/`、`/v1/chat/completions` 或 `/file_parse`。

模型没预热（冷启动），第一次调用巨慢或超时

现象：首次 OCR 识别等待很久（可能超过 60 秒），然后报超时。

原因：vLLM / Ollama / 本地 ONNX 模型首次加载需要将模型读入内存/显存，尤其是大模型（7B+）。默认超时仅 60 秒。

解决：

- **方案 A：**首次使用前先手动调用一次预热（用 curl 或 Ollama 的 API 测试页面）
- **方案 B：**在设置里把超时提高到 120-180 秒
- **方案 C：**Ollama 用户可以先 `ollama run <model_name>` 确保模型已在内存中

Ollama 已启动但 Base URL 填错

现象：测试连接报“无法连接到 127.0.0.1:11434”。

常犯错误：

- 填了 `http://localhost:11434/v1`（Ollama **不需要** `/v1` 后缀）
- 填了 `https://` 而不是 `http://`（本地 Ollama 默认不开 HTTPS）
- Ollama 服务实际监听在其他端口，但用户没改端口号
- Ollama 根本没启动（`ollama serve` 没运行）

✓ 快速验证

先在浏览器访问 `http://127.0.0.1:11434/api/tags`，如果返回 JSON 就说明正常。

模型名拼写错误 / 大小写不对

现象：测试连接提示“模型名不存在: xxx”，然后列出可用模型。

原因：模型名必须完全一致，包括大小写。

解决：

- Ollama 用户执行 `ollama list` 查看确切模型名

- OpenAI-compatible 用户查服务文档或 `/v1/models` 接口返回值

线上 API 用了但没填 API Key

现象：测试连接或识别时报 401 “接口认证失败”。

原因：线上服务（硅基流动、DeepSeek、OpenAI 等）必须提供 API Key。本地 Ollama 通常不需要。

解决：在设置的 API Key 栏填入正确的密钥。

⚠ 安全提醒

API Key 是敏感信息，请勿分享给他人或上传到公开仓库。LaTeXSnipper 不会将你的 API Key 发送到除你指定的 Base URL 以外的任何地方。

MinerU 协议选了但是服务端没配好

现象：测试连接报 404 或 409。

原因：MinerU 的默认测试端点是 `/health`，默认解析端点是 `/file_parse`。不同版本可能用不同路径。

解决：

- 先确认 MinerU 服务是否在运行：浏览器访问 `http://127.0.0.1:8000/health`
- 如果路径不对，在设置中修改“解析接口路径”和“健康检查路径”
- MinerU 409 错误通常表示解析任务失败，查看 MinerU 服务端日志

选了 OpenAI-compatible 协议但服务是 Ollama

现象：测试连接时访问 `/v1/models` 返回 404。

原因：Ollama 的模型列表接口是 `/api/tags`，不是 `/v1/models`。

解决：把协议从“OpenAI-compatible”改成“Ollama”。

! 协议选择速查

- Ollama 服务 → 选 **Ollama**
- 硅基流动 / DeepSeek / OpenAI / Groq 等 → 选 **OpenAI-compatible**
- MinerU 服务 → 选 **MinerU**
- 本地 MathCraft ONNX → 在“选择识别模型”中选 **内置模型**

自定义提示词把系统提示覆盖了导致输出格式错乱

现象：填写了自定义提示词后，输出变成了自然语言解释而不是 LaTeX 代码。

原因：自定义提示词优先级最高，会完全覆盖模板提示词。如果你写的提示词没有强调“只输出 LaTeX”，模型可能会自由发挥。

解决：如果不确定怎么写，清空自定义提示词，只用模板。

安装和环境问题

Python 版本不对

现象： 安装或运行时出现语法错误或不兼容提示。

原因： 不同场景的 Python 要求不同，不能混用。

解决：

- **普通 Windows 安装包用户：** 不需要单独安装 Python，安装包自带规范化的 Python 3.11 模板环境。
- **Linux/macOS 安装包用户：** 需要系统中有可用 Python $\geq 3.10, < 3.13$ ，仅用于创建用户目录下的隔离依赖环境。
- **源码运行/开发者：** 推荐 Python 3.11，与当前 Windows 打包环境保持一致。

```
# Python 3.11 创建虚拟环境
python3.11 -m venv .venv

# Windows 激活
.\.venv\Scripts\activate

# Linux/macOS 激活
source .venv/bin/activate

# Windows 安装公共依赖
pip install -r requirements.txt

# Linux 安装公共依赖和 Linux 快捷键依赖
pip install -r requirements-linux.txt

# macOS 安装公共依赖
pip install -r requirements-macos.txt
```

PyInstaller 打包版和开发版行为不同

现象： 从 GitHub Release 下载的 exe 能跑，但 git clone 后用源码跑不了。

原因： 打包版和源码版的环境边界不同。Windows 安装包包含规范化的 Python 3.11 模板环境，默认依赖根是安装目录下的 `_internal\deps`；Linux/macOS 安装包只包含 PyInstaller 应用本体，默认依赖根在用户可写目录中创建。源码运行则完全依赖你自己的开发环境。

解决：

- **如果是用户：** 优先使用 Release 版本，不要自己从源码跑
- **如果是开发者：** Windows 使用 `requirements.txt`；Linux/macOS 分别使用 `requirements-linux.txt` / `requirements-macos.txt`；需要打包或构建资源时再安装 `requirements-build.txt`

依赖环境创建在哪里

LaTeXSnippet 使用 `install_base_dir` 作为活动 Python 依赖根。用户在依赖向导或设置中切换依赖目录后，只有主 Python 依赖环境跟随新的依赖根；Pandoc 固定在应用状态目录的共享 `tools` 目录中，部署一次即可复用。

```
<安装目录>\_internal\deps
```

Linux/macOS 安装包不会把构建机的 `tools/deps/python311` 或任意虚拟环境打进安装包。默认依赖根是用户可写目录：

```
Linux: ~/.latexsnipper/deps
macOS: ~/Library/Application Support/LaTeXSnipper/deps
```

依赖根内的常见子目录如下：

```
<依赖根>/python311          主依赖 Python/venv
```

共享工具目录如下：

```
<应用状态目录>/tools/pandoc      可选 Pandoc 二进制目录
```

！为什么这样设计

Linux/macOS 的安装目录通常位于 `/usr/lib/latexsnipper` 或 `.app` 包内部，普通用户没有写权限。将依赖层放在用户可写依赖根可以避免权限错误，也避免把开发者机器上的 `venv`、`pyvenv.cfg`、`CI` 路径打进安装包。

Linux `.deb` 会声明 `python3` 和 `python3-venv` 依赖；macOS 没有 `.deb` 这种系统依赖声明机制，如果找不到可用 `python3`，请先安装：

```
# Homebrew
brew install python

# 或使用 python.org 官方 macOS 安装包
https://www.python.org/downloads/macos/
```

pip 依赖安装失败（特别是 pywin32 / PyQt6）

现象： `pip install -r requirements.txt` 报错。

依赖向导会先显示 UI；`ensurepip`、`pip` 升级以及 `setuptools` / `wheel` 修复只会在点击下载/安装后执行。若所选目录已有可用 Python 环境，向导会直接使用该环境。Windows 安装包默认使用内置 `python311` 模板；若用户切换到没有可复用 Python 的目录，三端都会使用系统 Python（3.10 到 3.12）创建隔离环境。

原因：

- `pywin32` 需要编译或特定 `wheel`
- `PyQt6` 体积大约多
- 某些包在国内 PyPI 镜像可能没有及时同步

解决：

- 使用清华/阿里云 PyPI 镜像：

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple -r requirements.txt
```

- Linux/macOS 源码运行请改用对应平台文件：`requirements-linux.txt` 或 `requirements-macos.txt`
- `pywin32` 只在 Windows 平台安装；如果 Windows `pip` 装不上，去 PyPI 手动下载匹配 Python 版本的 `wheel` 安装

权限问题（Program Files / 系统目录）

现象： 安装在 `C:\Program Files\` 下运行报权限错误。

原因： 应用需要在用户目录（`%APPDATA%`、`~/.latexsnipper/`）写入日志、配置、模型缓存。如果应用本身放在受保护目录，某些操作可能被 UAC 拦截。

解决：

- GitHub 安装包默认安装到 `%LOCALAPPDATA%\LaTeXSnipper`，普通用户权限即可运行
- 如果手动放入 `C:\Program Files\` 等受保护目录，不要把依赖目录、模型缓存或日志目录指向安装目录；保持默认用户目录通常没有问题

卸载后如何清理用户数据和模型缓存

LaTeXSnipper 卸载默认保留用户数据，方便升级或重装后继续使用原配置、历史记录、用户切换过的依赖环境、共享工具和 MathCraft 模型缓存。需要彻底清理时：

- **Windows:** 运行 Windows 安装包自带卸载程序时，会先出现可选清理窗口，随后仍会显示 Inno 标准卸载确认；标准确认通过后，卸载器会关闭正在运行的 LaTeXSnipper，并按勾选项删除用户数据/日志/临时文件、LaTeXSnipper 管理的共享工具目录、MathCraft 模型权重。安装目录内的 `_internal` 会随主程序卸载删除；用户切换到外部的完整 Python 或 venv 不会被卸载器删除。
- **Linux .deb:** 包管理器卸载不会自动删除 home 目录数据。卸载前可运行 `latexsnipper-clean-user-data`，按提示删除当前用户的数据、共享工具和模型缓存；脚本不会读取或删除用户切换过的外部 Python 依赖根。
- **macOS .dmg / .app.zip:** 删除 .app 只会移除应用本体。需要清理数据时，运行 .dmg 中的 `Uninstall User Data.command`，或运行 app 包内 `Contents/Resources/Uninstall User Data.command`。

如果你显式设置过 `MATHCRAFT_HOME` 指向自定义目录，卸载和清理脚本都不会自动删除该目录，避免误删用户指定的外部数据位置。

中文路径 / 用户名包含特殊字符

现象：启动崩溃或模型加载失败。

原因：Windows 用户名含中文、空格、特殊符号时，`%APPDATA%` 路径可能包含非 ASCII 字符，某些 C 库或 ONNX 底层可能无法正确处理。

解决：设置环境变量 `MATHCRAFT_HOME` 指向一个纯 ASCII 路径，例如 `C:\mathcraft_data`。

网络与更新问题

Windows 防火墙拦截本地服务

现象： Ollama / MinerU 等服务已启动，但 LaTeXSnippet 连不上。

原因： Windows Defender 防火墙可能拦截了本地回环连接。

解决：

- 检查防火墙设置，允许 Python / Ollama 通过
- 临时关闭防火墙测试（仅用于排查）
- 确认 Ollama 监听在 0.0.0.0 还是 127.0.0.1

更新检查失败

现象： 检查更新时一直转圈或报错。

原因： 更新检查访问 `api.github.com`，国内可能被墙或很慢。

解决：

- 设置代理后重试
- 直接去 GitHub Releases 手动下载最新版
- 查看 `%USERPROFILE%\.latexsnipper/logs/` 下的日志了解具体错误

杀毒软件误报 / 拦截

现象： exe 被删除、隔离或阻止运行。

原因： PyInstaller 打包的 exe 有时被误报为木马。

解决：

- 将安装目录加入杀毒软件白名单
- **确保从 GitHub Releases 下载**（优先使用 SignPath 签名安装包；签名不可用时官方 Release 可能提供同名未签名回退包）

PDF 与识别效果问题

PDF 识别结果很乱

现象： 用内置模型或外部模型识别 PDF 效果很差。

原因：

- DPI 设得太低 (< 100) 或太高 (> 200)
- 扫描件与文字型 PDF 处理方式不同
- 提示词模板不匹配 (用了公式模板去识别文档)

解决：

- 文字型 PDF：尝试 140-170 DPI
- 扫描件：尝试 200-300 DPI
- 外部模型普通图片/手写识别：确认设置中“输出偏好”和提示词模板与任务匹配
- PDF 识别：在 PDF 入口单独选择 Markdown/LaTeX 与 DPI；MinerU 文档解析固定输出 Markdown

切换了输出模式但结果没变

现象： 设置里选了 LaTeX 输出，但返回的还是 Markdown。

原因： 自定义提示词会覆盖输出模式设置；PDF 入口的格式选择独立于外部模型设置页的“输出偏好”；MinerU 协议走原生接口，不受输出偏好影响。

解决： 清空自定义提示词、确认协议类型；如果是 PDF，请重新从 PDF 入口选择导出格式。

大图片 / 高分辨率截图识别失败

现象： 截了 4K 屏幕的图，识别直接报错或超时。

原因： 图片太大，Base64 编码后体积膨胀约 33%，可能超过服务端请求体大小限制，或传输太慢导致超时。

解决：

- 缩小截图范围
- 降低截图分辨率
- 提高超时时间

使用技巧与功能问题

截图快捷键不生效

现象：按下载图快捷键后没有任何反应。

可能原因与解决：

- **快捷键被其他软件占用：**输入法、词典取词、录屏软件可能占用全局快捷键。暂时关闭这些软件，或在设置中更换截图快捷键
- **权限不足 (Linux/macOS)：**Linux Wayland 可能限制全局快捷键或截图；macOS 截图需要屏幕录制权限，参阅平台特定章节
- **应用在后台运行？**Windows 和有托盘的 Linux 会隐藏到系统托盘；macOS 会保持应用运行，可从 Dock 或菜单栏状态项重新打开

导出功能不可用 (Pandoc 相关)

现象：导出 Word / PowerPoint / EPUB / PDF / Typst 等格式时报错或选项灰色。

原因：这些格式依赖 Pandoc。Pandoc 是可选的外部工具；PDF 导出还需要系统中可用的 LaTeX PDF 引擎 (XeLaTeX、LuaLaTeX 或 pdfLaTeX)。

解决：

- 打开依赖向导，安装 “PANDOC” 层
- 或手动安装 Pandoc (<https://pandoc.org>) 并确保在 PATH 中
- 核心功能 (LaTeX / Markdown / MathML / HTML / OMML / SVG 导出) 不需要 Pandoc
- PDF 导出如果继续失败，请确认已安装 TeX Live / MiKTeX 等 LaTeX 发行版，并且 xelatex、lualatex 或 pdflatex 可在 PATH 中找到

手写识别不触发 / 延迟太高

现象：在手写窗口写完公式后没反应，或延迟很长才识别。

原因：

- OCR 服务未就绪 (内置 MathCraft 混合模式未加载，或外部模型未连接)
- 网络延迟 (使用远程外部模型时)
- 触控笔驱动问题

解决：

- 打开手写窗口后，程序会后台预热 MathCraft 混合模式；首次使用可能需要等待模型加载
- 手写窗口右下角有状态指示，确认显示 “就绪”
- 如果使用远程模型，检查网络延迟

手写识别为什么不跟随主窗口的公式/文字模式

说明：手写窗口的内置识别固定使用 MathCraft 混合模式。这样可以同时保留普通文字、中文/英文标签和公式，后续 “自动排版” 生成文档时不会因为只走公式模式而丢失文字。

影响：

- 主窗口截图识别仍然可以选择公式、文字、混合或外部模型
- 手写窗口使用内置模型时固定为混合模式，并会按混合模式预热
- 如果主窗口选择了外部模型，手写窗口仍会使用外部模型，但默认采用手写混合 OCR 提示词，以 Markdown + LaTeX 形式保留文字和公式

手写自动排版后文字、中文或换行不符合预期

现象： 自动排版后普通文字被省略，中文编译异常，或多行文字在 PDF 中挤到同一行。

当前策略：

- 自动排版会生成完整 XeLaTeX 文档，并使用 ctexart 文档类
- 导言区会补齐常用数学和表格宏包，如 amsmath、amssymb、mathtools、geometry 等
- 普通文字行会按段落保留，连续文字行会自动分段，避免 PDF 中被合并到同一行
- 如果外部模型排版时吞掉了识别草稿里的普通文字，程序会尝试把这些文字合并回文档正文

数学工作台计算结果不对

现象： 输入表达式后计算返回错误结果或超时。

原因：

- 表达式语法问题（如隐式乘法未加 *）
- Compute Engine 无法处理的复杂表达式

解决：

- 检查表达式语法：2x 需要写成 2*x
- 计算结果以数学工作台内置前端计算引擎为准
- 对于超长运行的计算，耐心等待回退引擎结果

深色/浅色主题切换后显示异常

现象： 切换主题后某些窗口颜色错乱或文字不可见。

解决：

- 重启应用确保所有窗口正确应用主题
- 如果问题持续，在设置中明确选择“浅色”或“深色”而非“跟随系统”

Pandoc 导出中文乱码

现象： 导出 .docx 或 .epub 后中文显示为乱码。

原因： Pandoc 默认可能使用非 UTF-8 编码，或缺少中文字体。

解决：

- 确保 Pandoc 版本 ≥ 3.0
- 导出 LaTeX PDF 时确保安装了 ctex 宏包或中文字体
- Word 导出后在 Word 中手动设置字体

平台特定问题（Linux / macOS）

三端主要差异速查

LaTeXSnippet 的主流程在 Windows、Linux、macOS 上保持一致：截图识别、图片识别、PDF 识别、手写识别、导出、历史记录、收藏夹和数学工作台使用同一套业务逻辑。

主要差异集中在系统集成层：

- **截图快捷键：** Windows 使用 Win32 原生全局快捷键；Linux 使用 pynput 全局快捷键，X11 最稳定，Wayland 可能受桌面环境策略限制；macOS 使用 Carbon 原生全局快捷键。
- **可设置快捷键：** Windows/Linux 支持 Ctrl+字母 或 Ctrl+Shift+字母；macOS 支持非保留 Command+字母 / Command+Shift+字母，也支持 Option+Command 组合。
- **默认快捷键：** Windows/Linux 为 Ctrl+F；macOS 为 Command+F。

- **截图实现：** Windows 使用 Qt 框选截图；Linux 先走 Qt，失败后可尝试 `grim`、`maim`、`gnome-screenshot` 等系统工具或 `portal` 回退；macOS 先走 Qt，可回退到系统 `screencapture`，首次使用可能弹出屏幕录制权限。
- **关闭窗口 / 后台常驻：** Windows 关闭主窗口会隐藏到系统托盘，托盘菜单“退出”才真正退出；Linux 有系统托盘时关闭主窗口会隐藏到托盘，没有托盘时会询问是否退出；macOS 关闭主窗口会最小化并保持应用运行，Dock 或菜单栏“退出”才真正退出。
- **权限要求：** Windows 普通截图路径不需要额外系统权限；Linux Wayland 可能限制截图和全局快捷键；macOS 截图需要屏幕录制权限，Carbon 全局快捷键通常不需要辅助功能权限。
- **依赖环境：** Windows 默认依赖根为 `<安装目录>\_internal\deps`，可切换；Linux 默认依赖根为 `~/.latexsnipper/deps`，可切换；macOS 默认依赖根为 `~/Library/Application Support/LaTeXSnipper/deps`，可切换。Linux/macOS 需要系统 Python ≥ 3.10 , < 3.13 创建 `venv`。
- **安装包：** Windows 使用 Inno 安装包；GitHub Release 优先发布签名安装包，签名不可用时发布同名未签名回退包；Linux 使用 Debian/Ubuntu `.deb`；macOS 使用 `.dmg` 或 `.app.zip`。

！快捷键兼容性

快捷键设置入口使用平台主修饰键：Windows/Linux 接受 `Ctrl+字母` 或 `Ctrl+Shift+字母`。macOS 接受非系统保留的 `Command+字母` / `Command+Shift+字母`，并支持 `Option+Command` 组合；`Command+Q/H/M/W/A/C/V/X/Z/Space/Tab` 和系统截图键会被拒绝。若 Linux Wayland 环境下快捷键或截图无响应，通常是桌面环境权限或全局快捷键策略限制，优先参考下方 Wayland 说明。macOS 使用 Carbon 注册全局快捷键，通常不需要开启“辅助功能”权限；截图失败时应优先检查“屏幕录制”权限。

Linux: Wayland 下截图功能异常

现象： 在 Wayland 会话下截图黑屏、部分窗口无法截取或快捷键无响应。

原因： Wayland 的安全模型限制了应用间的屏幕捕获，Qt 的截图机制可能需要额外配置。

解决：

- 尝试安装并配置 `grim + slurp` (wlroots 系) 或 `gnome-screenshot` (GNOME)
- 应用会自动检测这些工具作为回退方案
- 某些发行版需要在设置中允许屏幕共享权限

！Linux 依赖提示

依赖向导只管理 Python 依赖层。`grim`、`maim`、`gnome-screenshot` 等系统工具需要用户自行通过包管理器安装。这些工具是可选的回退方案——应用在 Qt 原生截图失败时自动尝试它们。

Linux: 虚拟机或 Wayland 下启动后直接中止

现象： 命令行出现 `EGL not available`、`No physical devices`、`GL0zone not found`、`Failed to get system egl display`，随后程序中止。

原因： 这通常不是 MathCraft GPU 推理问题，而是 Qt WebEngine / Chromium 在 Wayland、虚拟机、WSL 或没有可用 DRI render 节点时无法初始化 EGL/GPU 图形后端。

当前策略： LaTeXSnipper 会在高风险 Linux 图形环境中自动启用软件渲染兜底。正常 X11 + 真实 GPU + 可用 `/dev/dri/renderD*` 的用户仍走原生路径，不会被强制降级。

手动开关：

```
# 强制启用 Linux 图形兜底
LATEXSNIPPER_FORCE_LINUX_GRAPHICS_FALLBACKS=1 latexsnipper

# 禁用 Linux 图形兜底，尝试原生 Qt/GPU 路径
LATEXSNIPPER_DISABLE_LINUX_GRAPHICS_FALLBACKS=1 latexsnipper
```

如果仍然无法启动，通常是系统缺少 XWayland/XCB/QtWebEngine 运行库。Debian/Ubuntu 用户可先确认这些包是否存在：

```
sudo apt install xwayland libxcb-cursor0 libxcb-icccm4 libxcb-image0 \
libxcb-keysyms1 libxcb-render-util0 libxcb-xinerama0 libxkbcommon-x11-0
```

macOS: 首次启动被阻止

现象： macOS 提示“无法验证开发者”或“已损坏”。

原因： 应用未经过 Apple 公证 (notarization)。

解决：

- 前往 系统设置 → 隐私与安全性 → 点击“仍要打开”
- 或终端执行：xattr -cr /Applications/LaTeXSnippet.app

macOS: 截图权限弹窗

现象： 首次截图时系统弹窗要求授予屏幕录制权限。

解决：

- 在系统设置 → 隐私与安全性 → 屏幕录制中勾选 LaTeXSnippet
- 授权后如果仍无法截图，请完全退出并重新打开应用

macOS: 全局快捷键是否需要辅助功能权限

结论： 通常不需要。

LaTeXSnippet 在 macOS 上使用 Carbon 原生全局快捷键注册，不使用 pynput 或键盘事件监听线程。截图快捷键不生效时，优先检查快捷键是否被系统或其他软件占用，以及应用是否仍在运行。只有涉及低层键盘监听、模拟键鼠控制或其他辅助控制能力的软件通常才需要“辅助功能”权限。

macOS: 没有可用 pip 或 pip 版本太低

现象： 依赖向导提示找不到 pip，或安装依赖时出现 pip / setuptools / wheel 版本过低。

原因： macOS 安装包不内置完整 Python 依赖环境，需要借助系统中可用的 Python $\geq 3.10, < 3.13$ 在活动依赖根下创建 python311。默认依赖根是 ~/Library/Application Support/LaTeXSnippet/deps。如果系统 Python 没有 venv / ensurepip / pip，依赖层无法自动初始化。

解决：

- 推荐安装 Homebrew Python: brew install python
- 或安装 python.org 官方 macOS Python 包
- 重新启动 LaTeXSnippet 后再运行依赖向导

各平台数据目录速查

配置文件 & 日志：

```
Windows: %USERPROFILE%\latexsnipper\
Linux:   ~/.latexsnipper/
macOS:   ~/Library/Application Support/LaTeXSnippet/
```

依赖根：

```
Windows: <安装目录>\_internal\deps (默认，可在依赖向导/设置中切换)
Linux:   ~/.latexsnipper/deps (默认，可切换)
macOS:   ~/Library/Application Support/LaTeXSnippet/deps (默认，可切换)
```

依赖根内：

```
python311          主依赖 Python/venv
```

共享工具目录：

```
<应用状态目录>/tools/pandoc          可选 Pandoc 二进制目录
```

模型缓存 (MathCraft ONNX):

Windows: %APPDATA%\MathCraft\models\

Linux: \${XDG_DATA_HOME:- ~/.local/share}/LaTeXSnipper/MathCraft/models/

macOS: ~/Library/Application Support/LaTeXSnipper/MathCraft/models/

与其他软件冲突

多版本 Python 共存导致调用错误

现象： 命令行能跑但 GUI 跑不了，或反之。

原因： 系统中安装了多个 Python 版本，PATH 中执行的是错误的 Python。

解决：

- where python (Windows) / which python (Linux/macOS) 检查当前使用的 Python 路径
- 确保虚拟环境已激活
- 使用绝对路径运行

输入法 / 屏幕取词软件干扰截图

现象： 截图快捷键无效或截到的画面不对。

原因： 某些输入法、词典取词、录屏软件占用全局快捷键或 hook 了屏幕捕获。

解决： 暂时关闭冲突软件的屏幕相关功能，或在 LaTeXSnippet 设置中修改截图快捷键。

如何有效反馈 Bug

X 先说最重要的

只说“报错了”“用不了”“闪退”而不提供日志的 issue，不会被处理。没有日志 = 无法定位 = 无法修复。

反馈 Bug 时请务必附带以下 **全部** 信息：

必须提供的信息

1. 日志文件

位于 `%USERPROFILE%\latexsnipper\logs\`，把 **整个 logs 目录打包** 发过来。命令行运行时 `stderr` 的输出也要一并附上。

2. 崩溃日志

`crash-native.log`（在日志目录下），如果有的话。

3. 运行环境

操作系统版本、安装方式（exe 还是源码）、Python 版本（源码运行时）。

4. 外部模型配置

协议类型、模型名、是否本地部署、是否使用代理。

5. 复现步骤

你点了什么、填了什么配置、什么时候报的错。越详细越好。

6. 截图

错误提示的 **完整截图**（不要裁剪，要包含窗口标题栏）。

去哪里反馈

- **GitHub Issues:** <https://github.com/SakuraMathcraft/LaTeXSnipper/issues>
- **GitHub Discussions:** <https://github.com/SakuraMathcraft/LaTeXSnipper/discussions>（一般使用问题请发这里）

从源码运行与开发者常见问题

README Quick Start 步骤验证

README 中提供了三种平台的源码运行步骤，经验证均正确可行：

✓ 已验证的步骤

- **Windows:** `python -m venv .venv → pip install -r requirements.txt → python src/main.py`
- **Linux:** `python3 -m venv .venv → pip install -r requirements-linux.txt → python src/main.py`
- **macOS:** `python3 -m venv .venv → pip install -r requirements-macos.txt → python src/main.py`
- `requirements-linux.txt` 和 `requirements-macos.txt` 通过 `-r requirements.txt` 自动包含公共依赖；Linux 额外使用 `pynput` 支持全局快捷键，macOS 使用原生全局快捷键实现。

开发环境路径不要混用

现象：本地存在 `python311/` 和 `tools/deps/python311/` 两套 Python，不确定应该用哪一个。

规则：

- `python311/` 是 Windows 安装包的内置 Python 模板，只用于打包复制，不要安装开发依赖，不要执行 `ruff/pyright/pytest`。
- `tools/deps/python311/` 是 IDE、开发检查和 Actions 打包使用的项目环境，可以安装 `requirements*.txt`、`ruff`、`pytest`、`pyright`。
- Linux/macOS 安装包运行时不会携带 `tools/deps/python311`。Linux 默认依赖根是 `~/.latexsnipper/deps`，macOS 默认依赖根是 `~/Library/Application Support/LaTeXSnipper/deps`；用户切换依赖目录后，以配置中的 `install_base_dir` 为准。

Windows 开发者初始化示例：

```
py -3.11 -m venv tools\deps\python311
.\tools\deps\python311\python.exe -m pip install --upgrade pip wheel setuptools
.\tools\deps\python311\python.exe -m pip install -r requirements.txt
.\tools\deps\python311\python.exe -m pip install -r requirements-build.txt
.\tools\deps\python311\python.exe -m pip install ruff pytest pyright
```

! 路径速查

- 仓库根目录 `python311/` — Windows 模板环境，不要污染。
- `tools/deps/python311/` — 开发、检查、构建环境。
- `~/.latexsnipper/deps` — Linux 默认运行时依赖根。
- `~/Library/Application Support/LaTeXSnipper/deps` — macOS 默认运行时依赖根。
- `<依赖根>/python311` — 程序创建的主 Python 依赖环境。
- `<应用状态目录>/tools/pandoc` — 程序创建的共享工具目录，切换依赖根后继续复用。

开发者验证命令注意事项

常用验证命令：

```
.\tools\deps\python311\python.exe -m ruff check .
.\tools\deps\python311\python.exe -m pytest test
.\tools\deps\python311\python.exe -m pyright
.\tools\deps\python311\python.exe -m compileall -q src mathcraft_ocr test
```

⚠️ 运行前必做

ruff、pytest、pyright 不在任何 requirements*.txt 中。首次运行验证前需要手动安装：

```
.\tools\deps\python311\python.exe -m pip install ruff pytest pyright
```

pyrightconfig.json 中的路径

类型检查应使用 tools/deps/python311 的第三方包解析环境，根目录 python311/ 只作为 Windows 模板排除在检查外。不要为了让 pyright 通过而向根目录模板安装开发包。

✓ 提示

如果 tools/deps/python311 不存在，先创建开发环境；不要改用根目录 python311 作为替代。

加载项介绍与系统要求

什么是 Office 加载项

LaTeXSnipper Office 加载项是一个 Windows 原生 VSTO 插件，安装后会在 Word 和 PowerPoint 的功能区（Ribbon）中添加 LaTeXSnipper 专用标签页。OLE 公式对象和 Word OMML 公式都由本地组件处理；本机 Bridge 仅用于与 LaTeXSnipper 桌面端通信并读取截图 OCR 识别结果。

! 与桌面端的关系

OLE 公式对象和 Word OMML 公式不依赖桌面端 Bridge。只有截图 OCR 需要本机运行的 LaTeXSnipper 桌面端提供识别结果；使用截图识别前，请确保桌面端已启动并开启了 Office 插件功能。

系统要求

- **Windows 版本：**Windows 10 22H2 或更高版本 / Windows 11
- **Office 版本：**Microsoft 365 应用（当前通道或月度企业通道）、Office 2024 / 2021 / 2019 零售版或批量许可版、Office LTSC 2024 / 2021
- **运行时：**.NET Framework 4.8；Microsoft Edge WebView2 Runtime（Windows 11 已内置；M365 会自动安装）
- **Office 体系：**仅支持 32 位和 64 位桌面版 Office；不支持网页版、移动版和 macOS 版 Office
- **LaTeXSnipper 桌面端：**只有截图 OCR 需要本机运行并开启 Office 插件功能；普通公式插入、加载、更新、转换、编号和引用均由插件本地完成

安装前检查

⚠ 首次安装必读

- 安装前请关闭所有正在运行的 Word 和 PowerPoint 窗口。Office 进程会锁定加载项 DLL，安装程序无法覆盖。
- 如果此前安装过开发版（通过脚本手动注册），请先使用注册脚本的反注册功能清理，或运行安装包执行升级安装。
- 安装程序需要管理员权限，用于写入 HKLM 注册表和受信任的发布者证书存储。

安装与注册

安装流程

安装程序会自动完成以下步骤：

1. 将 Word 和 PowerPoint 加载项文件复制到安装目录
2. 将代码签名证书安装到受信任的发布者存储
3. 写入 HKLM 注册表键值（同时覆盖 32 位和 64 位 Office 路径）
4. 调用 VSTO 安装程序静默注册加载项
5. 清理 Office 禁用项和崩溃记录

自定义安装路径

安装程序支持自定义安装路径，VSTO 注册和文件引用会自动适配所选路径。不建议将加载项安装到受保护的临时目录或网络路径。

卸载

通过 Windows “设置 → 应用 → 已安装的应用” 卸载，或运行安装目录下的 `unins000.exe`。卸载程序会自动：

1. 删除所有安装文件
2. 移除 HKLM 注册表键值
3. 调用 VSTO 安装程序反注册加载项
4. 清理 Office 禁用项和崩溃记录

命令行静默安装

安装包支持 Inno Setup 标准静默参数：

```
# 静默安装（显示进度条）
OfficePluginSetup-2.4.0.exe /silent

# 完全静默（无界面）
OfficePluginSetup-2.4.0.exe /verysilent

# 自定义安装目录
OfficePluginSetup-2.4.0.exe /dir="D:\Tools\LaTeXSnipper"
```

Word 功能详解

功能区概览

安装后，Word 功能区的最后位置会出现 **LaTeXSnippet** 标签页，包含 8 个功能组：

- **公式组**：行内公式、行间公式、带编号公式、截图识别
- **编辑组**：加载所选、删除所选
- **转换组**：转为 OLE、转为 Word
- **引用组**：插入引用
- **编号组**：添加编号、重编号
- **边界组**：插入章分隔符、插入节边界
- **格式化组**：格式化所选、格式化全部
- **工具组**：状态窗格、设置、帮助

插入 OMML 公式

Word 加载项通过本地 OMML 转换引擎将 LaTeX 转换为 Word 原生 OMML (Office Math Markup Language) 公式，不依赖桌面端 Bridge。OMML 公式是 Word 原生格式，可以像手动插入的公式一样编辑。

! OMML 公式

OMML 公式是 Word 原生格式，卸载加载项后公式仍然可以正常显示和编辑。加载项插入的公式会附带 LaTeXSnippet 管理元数据 (LaTeX 源码、显示模式、编号模式)，用于后续加载、更新和重编号。普通 Word 公式没有 LaTeXSnippet 元数据，插件不会猜测其 LaTeX 来源。

三种插入模式

- **行内公式**：插入正文中的行内公式，跟随文字排版
- **行间公式**：插入居中显示的独立公式
- **带编号公式**：插入自动编号的行间公式，编号位置、外框、章/节层级和层级分隔符可在设置中配置

加载、更新和删除

选中由 LaTeXSnippet 插入的公式后：

- **加载所选**：将公式的完整 LaTeX 源码加载到公式编辑器，可以修改后更新原公式
- **更新**：在编辑器中修改后点击更新，新公式会替换原公式，保留元数据
- **删除所选**：删除公式及对应的元数据

添加编号与重编号

Word 支持将未编号的 LaTeXSnippet 公式转换为自动编号公式，并按文档顺序维护自动编号：

- **添加编号**：为选中的行间公式添加自动编号。如果公式已有编号，不会重复添加
- **重编号全部**：按文档顺序更新所有自动编号公式的 SEQ 字段、编号书签和引用字段，不重新渲染公式内容；缺失元数据或编号字段的异常公式会被跳过并计数

- **编号位置：** 设置中可选择右编号或左编号；该设置只影响新插入的编号公式和之后被自动编号的普通公式，不改变已有公式的编号属性

章/节分隔符与层级编号

支持多层级公式编号体系。在文档中插入章或节分隔符后，自动编号会按层级生成（例如 1.1、1.2）。

- **插入章分隔符：** 标记新章开始，后续自动编号重置为当前章号
- **插入节边界：** 标记新节开始，后续自动编号追加当前节号
- **包含章/节编号：** 设置中可控制编号是否包含章或节的层级部分
- **分隔符样式：** 层级分隔符可在设置中选择（.、-、·、:、/）
- **隐藏分隔符：** 设置中可隐藏插入文档中的分隔符标记

交叉引用

支持在文档正文中插入对公式编号的交叉引用：

- **插入引用：** 在光标位置插入引用占位符，然后选中目标公式或目标公式所在段落，自动生成交叉引用
- 交叉引用与 Word 原生交叉引用系统集成，可通过 Word 的“更新域”刷新

公式转换

支持在已有的管理公式之间转换格式，也支持把选中的 Word 原生 OMML 公式显式转换为 LaTeXSnipper OLE：

- **转为 OLE：** 将选中的 OMML 管理公式转为 OLE 公式对象（支持更多渲染选项）
- **转为 Word：** 将 OLE 公式转为 Word 原生 OMML 公式（可像 Word 公式一样直接编辑）
- **Word 原生 OMML 转 OLE：** 选中非 LaTeXSnipper 托管的 Word 原生公式后执行“转为 OLE”，插件会读取原生 OMML，转换为 MathML，并把 MathML 作为源码存入 OLE 元数据。这类公式可加载和继续转换，但不应执行“格式化所选”，因为格式化链路按 LaTeX 宏重写源码，MathML 源码会被破坏。

格式重置

- **格式化所选：** 将选中的公式恢复为默认字体、颜色和自然尺寸
- **格式化全部：** 将所有尺寸被修改过的公式恢复为自然尺寸

性能与复杂度参考

下面的复杂度用于估算长文档中的响应时间。设 n 为参与操作的公式数， m 为文档中 LaTeXSnipper 管理公式总数， c 为文档中的内容控件、图片/OLE 对象或形状数量， f 为 Word 域数量， p 为当前段落内公式对象数量， s 为单个公式源码或 OMML/MathML XML 的长度， r 为一次 MathJax 渲染或图片/OLE 生成成本。

操作	Word 复杂度	PowerPoint 复杂度	主要耗时来源
插入 OLE 公式	$O(s + r + ole)$	$O(s + r + ole)$	MathJax 渲染、EMF/OLE 对象创建、Office COM 插入

插入 Word OMML	$O(s + r + \text{xsl} + \text{xml})$	不适用	MathJax 转 MathML、MML2OMML XSL、Word InsertXML
插入 PowerPoint PNG	不适用	$O(s + r + \text{png})$	MathJax 渲染、SVG/PNG 光栅化、形状插入
插入带编号公式	$O(s + r + \text{ole/xml} + c)$	不适用	公式插入、SEQ 编号域、书签、居中/右对齐制表位布局
加载所选	$O(c)$ 最坏，通常接近 $O(1)$	$O(c)$ 最坏，通常接近 $O(1)$	从选区附近定位元数据，必要时扫描对象
更新公式	$O(s + r + \text{replace})$	$O(s + r + \text{replace})$	重新渲染和替换原对象；OLE 会保留用户缩放
删除所选	$O(c + f + n \log n)$	$O(c + n)$	定位并按文档位置删除公式、边界控件、引用域或引用占位符
添加编号	$O(s + r + \text{replace} + \text{before})$	不适用	将普通行间公式改为编号布局，只扫描当前位置前的编号时间线
重编号全部	$O(c + f + m \log m + \text{refs})$	不适用	收集并排序自动编号公式和章/节边界，一次建立 SEQ 字段映射，更新编号书签和 LaTeXSnipper 引用域
插入章/节分隔符	$O(c + m \log m + \text{refs})$	不适用	插入边界控件后立即重编号全部自动编号公式
插入交叉引用	$O(1)$ 插入占位，选择目标时最坏 $O(c + f)$	不适用	查找目标公式、定位编号书签并创建 Word REF 域
转 OLE (插件 OMML)	$O(n * (s + r + \text{ole}))$	$O(n * (s + r + \text{ole}))$ (PNG → OLE)	渲染、OLE 创建、逐个 COM 替换
Word 原生 OMath 转 OLE	$O(n * (s + \text{xsl} + r + \text{ole}))$	不适用	OMML2MML XSL、MathJax 按 MathML 渲染、OLE 创建
转 Word OMML / PNG	$O(n * (s + r + \text{xsl} + \text{xml}))$	$O(n * (s + r + \text{png}))$	OLE/PNG 重新生成和 Office 对象替换
格式化所选	$O(n * (s + r + \text{replace}) + p)$	$O(n * (s + r + \text{replace}))$	只有属性不同或尺寸被改过时才重新渲染/重置；Word 行内基线只扫描当前段落
格式化全部	$O(m)$	$O(m)$	扫描全文公式并恢复被用户缩放过的自然尺寸
截图识别	$O(\text{ocr} + s + r)$	$O(\text{ocr} + s + r)$	桌面端 OCR 和后续公式插入

多选转换和多选格式化所选会先收集稳定快照（公式 ID、起始位置、渲染类型和必要元数据），不跨批持有 live COM object。转换和涉及替换的格式化按文档位置倒序分批执行，每批 5 个公式，每批更新一次状态窗格；Word 每批使用短 undo record。长公式或大量公式时，主要耗时仍来自 r （MathJax 渲染/图片生成）和 ole/xml/png （Office COM 插入与对象替换）。建议长文档批量清理时分段执行并保存文档。

Word 原生 OMath 转 OLE 保存的是 MathML 源码，不是 LaTeX 源码；它用于保真转换，不进入 LaTeX 格式化链路。原生行内公式由 Word 自己维护外层公式输入框，少数单选场景下外层占位框可能由 Word 保留；继续转为 LaTeXSnipper 管理的 OMML 后会由插件接管并清理为插件格式。

截图识别

点击“截图识别”后，加载项等待桌面端完成下一次截图 OCR。若需取消，可点击功能区“取消识别”按钮或在等待过程中再次点击“截图识别”按钮。OCR 识别结果会自动填入公式编辑器。

PowerPoint 功能详解

功能区概览

PowerPoint 的 LaTeXSnipper 标签页结构与 Word 类似，但功能更精简：

- **公式组**：插入公式、截图识别
- **编辑组**：加载所选、删除所选
- **转换组**：转为 OLE、转为 PNG
- **格式化组**：格式化所选、格式化全文
- **工具组**：状态窗格、设置、帮助

插入公式图片

PowerPoint 加载项支持两种渲染方式：

- **OLE 公式对象**：将 LaTeX 交由本机 OLE 公式对象处理器渲染，以 OLE 对象形式嵌入幻灯片，保留原始版式
- **高 DPI PNG 图片**：由内置 MathJax 渲染为 SVG 后光栅化为高 DPI PNG 图片插入幻灯片

! 与 Word 的区别

- 没有行内/行间区别——PowerPoint 幻灯片上的公式始终是独立对象
- 不支持自动编号和重编号（这些是 Word 文档专属功能）

加载与删除

- **加载所选**：选中幻灯片上的公式图片，点击加载所选，公式的 LaTeX 源码会载入独立公式编辑器。修改后重新插入时，新图片会替换原位置旧图片
- **删除所选**：删除选中的公式图片及元数据

公式转换与格式重置

- **转为 OLE**：将选中 PNG 格式的公式图片转换为 OLE 公式对象
- **转为 PNG**：将 OLE 公式转换为 PNG 图片
- **格式化所选**：将选中的公式恢复为默认字体、颜色和自然尺寸
- **格式化全文**：将所有尺寸被修改过的公式恢复为自然尺寸

✓ 提示

如果你在幻灯片上找不到公式图片，可以检查形状的替代文本（Alt Text）。所有 LaTeXSnipper 公式图片的替代文本均以 “LaTeXSnipper formula” 开头。

截图识别

与 Word 相同的截图 OCR 流程。识别结果会自动填入公式编辑器，可直接插入或修改后插入。

公式编辑器与侧边栏

侧边栏（任务窗格）

侧边栏是加载项的常驻面板，通过功能区“状态窗格”按钮或自动弹出显示：

- **MathLive 编辑器：** 侧边栏顶部是可视化公式编辑器，支持所见即所得的公式编辑
- **LaTeX 源码：** 编辑器下方是 LaTeX 源码输入区，编辑器和源码区双向同步
- **连接按钮：** 测试与桌面端 Bridge 的连通性
- **截图识别按钮：** 触发截图 OCR 等待状态
- **插入按钮：** 将当前侧边栏公式直接插入文档/幻灯片

独立公式编辑器

通过功能区“行内公式”“行间公式”“带编号公式”按钮或“加载所选”打开的独立公式编辑器窗口，搭载 MathLive 可视化公式引擎和完整数学符号系统。

- **数学编辑区：** 全尺寸 MathLive 所见即所得编辑区，支持键盘快捷键（如 $+$ $_$ $\sqrt{\quad}$ 等标准 LaTeX 缩写自动识别）、光标导航和即时预览
- **LaTeX 源码区：** 编辑区下方的 LaTeX 源码输入区，与可视化编辑区实时双向同步，可直接粘贴或编辑 LaTeX 代码
- **符号面板（右侧）：** 内置 **18 分类、2121 个数学符号与公式模板**，覆盖从基础算术到前沿研究的全部符号体系：
 - **基础（4 类 171 条）：** 希腊字母（含变体和希伯来字母）、结构模板（矩阵/根式/行列式等，支持自选行列数）、分隔符（含大尺寸变体和装饰框线）、集合与逻辑符号
 - **代数（1 类 174 条）：** 线性代数（矩阵运算/特征系统）/抽象代数（群环域/同调代数/李理论/代数几何）/算子代数（ C^* 代数/von Neumann/K-理论）
 - **几何（1 类 133 条）：** 微分几何/黎曼曲率/陈类/Chern-Simons/Atiyah-Singer 指标/辛几何
 - **数论（1 类 166 条）：** 同余/二次互反律/模形式/L 函数/丢番图方程/解析数论
 - **分析（1 类 210 条）：** 微积分/实复分析/傅里叶分析/PDE/泛函分析/不等式/估计理论
 - **拓扑（1 类 166 条）：** 一般拓扑/同调论/同伦论/示性类/莫尔斯理论/谱序列/K 理论
 - **关系与箭头（4 类 264 条）：** 比较关系（含各种否定变体）/二元运算（含环论算子）/大型运算符/箭头符号
 - **函数（1 类 131 条）：** 三角函数/反三角函数/双曲函数/对数/极值/实虚部/误差函数/积分正弦余弦
 - **概率统计（1 类 170 条）：** 常见分布/随机过程/鞅/信息论/极限定理/不等式
 - **物理（1 类 251 条）：** 力学/电磁学/热力学/量子力学/相对论/标准模型/量子场论/弦论
 - **化学（1 类 229 条）：** 分子式/有机官能团/化学反应/物化常数/光谱学/电化学
 - **其他（1 类 56 条）：** 杂项符号
- **标签内搜索：** 每个符号面板顶部设有搜索框，支持按中文名称或 LaTeX 源码实时过滤当前标签内的符号
- **全局搜索：** 标签列顶部设有全局搜索框，输入关键词后跨所有 18 个标签检索，按组归类展示结果
- **底部工具栏：** 基础运算（计算/化简/数值化/求解/展开/因式分解）、多行排版布局、快捷插入片段、示例载入、复制（LaTeX / MathJSON）
- **插入 / 更新：** 底部按钮根据当前模式显示“插入”或“更新”

- **取消：** 关闭编辑器，不应用更改

✓ 编辑器快捷键

- Enter：新建数学行
- Shift+Enter：插入或更新当前公式
- Esc：只收回 MathLive 虚拟键盘，不关闭编辑器

平台界面差异

Word 侧边栏包含行间公式、自动编号、自定义编号输入框等 Word 专属控件。PowerPoint 侧边栏不包含这些选项，界面更简洁。

设置与帮助

设置窗口

通过功能区“设置”按钮打开。设置窗口支持完整的公式参数配置：

- **公式插入方式：**OLE 对象（默认，不依赖 Bridge）或 Word OMML（本地转换，不依赖 Bridge）
- **编号位置：**右编号（默认）或左编号
- **外框：**无、圆括号 (())、方括号 ([])、花括号 ({})
- **层级分隔符：**、 -、·、:、/
- **包含章编号 / 包含节编号：**控制编号是否包含层级部分
- **隐藏章分隔符 / 隐藏节分隔符：**控制文档中是否显示分隔符标记
- **字体颜色：**拾色器选择新插入公式的默认字体颜色，支持恢复为黑色或白色
- **默认字体：**TeX 原生字体、罗马正体、粗体、斜体、黑板粗体、花体、哥特体、无衬线、等宽等论文常用数学字体

编辑器键盘行为

快捷键	功能
Enter	新建数学行
Shift + Enter	插入或更新当前公式
Ctrl + F	插入分式
Ctrl + R	插入根号
Ctrl + H	插入上标
Ctrl + L	插入下标
Ctrl + J	插入上下标
Esc	收回 MathLive 虚拟键盘

帮助窗口

通过功能区“帮助”按钮打开，根据当前宿主（Word 或 PowerPoint）显示对应内容：

- 系统要求详情
- Word / PowerPoint 功能列表
- 功能区按钮说明
- 插件架构介绍
- 使用边界说明

常见问题排查

加载项未出现在功能区

现象：安装后打开 Word 或 PowerPoint，功能区没有 LaTeXSnippet 标签页。

排查步骤：

1. 打开 Word / PowerPoint → 文件 → 选项 → 加载项
2. 在“管理”下拉列表中选择“COM 加载项”，点击“转到”
3. 检查列表中是否有 **LaTeXSnippet**
4. 如果存在但未勾选，勾选后确定
5. 如果存在但被禁用，在“禁用的应用程序加载项”中重新启用

如果列表中完全没有 LaTeXSnippet：

1. 确认安装程序以管理员权限运行
2. 检查杀毒软件是否拦截了安装

连接桌面端失败

现象：使用截图识别或 OMML 公式插入时提示“无法连接到 LaTeXSnippet”（OLE 公式对象渲染不依赖 Bridge 连接）。

解决：

1. 确认 LaTeXSnippet 桌面端正在运行
2. 在桌面端设置中确认“Office 插件功能”已开启
3. 检查防火墙是否拦截了 127.0.0.1:28765 端口
4. 如果桌面端改变了 Bridge 端口，设置环境变量 LATEXSNIPPER_OFFICE_BRIDGE_URL 指向正确地址

公式编辑器加载失败

现象：侧边栏或独立编辑器显示空白或加载错误。

原因：公式编辑器（包括 MathLive 可视化编辑器和符号面板）依赖 Microsoft Edge WebView2 Runtime。如果系统中未安装或版本过旧，编辑器可能无法加载。

解决：

1. 从 <https://go.microsoft.com/fwlink/p/?LinkId=2124703> 下载并安装 WebView2 Runtime
2. Windows 11 和 Microsoft 365 通常已内置 WebView2

截图 OCR 提示“正忙”

现象：点击截图识别后很快提示错误“截图识别正忙，请稍后再试”。

原因：桌面端上一次 OCR 请求尚未完成或未正确取消。加载项会自动取消旧请求并重试一次。如果重试仍然失败，可能桌面端正在处理其他任务。

解决：稍等片刻后重试。如果持续出现，关闭并重新打开桌面端的 Office 插件功能。

加载项被 Office 禁用

现象：加载项之前正常使用，某次打开 Office 后消失了。

原因：Office 检测到加载项启动耗时过长或崩溃，将其自动禁用。

解决：

1. 文件 → 选项 → 加载项 → 管理“禁用的应用程序加载项” → 转到
2. 找到 LaTeXSnippet，重新启用
3. 如果频繁被禁用，查看 Office 加载项 Resiliency 注册表
(HKCU\Software\Microsoft\Office\{App}\Resiliency) 并清理对应条目

安装后出现两个无图标程序条目

现象： Windows “设置 → 应用 → 已安装的应用” 中出现 LaTeXSnipper.OfficePlugin.WordVstoAddIn 和 LaTeXSnipper.OfficePlugin.PowerPointVstoAddIn 无图标条目。

说明： 这些是 VSTO ClickOnce 部署的注册痕迹，属于正常现象。加载项安装程序已将其标记为系统组件，通常不会在用户界面上显示。如果仍然可见，不影响加载项的正常使用。

LaTeXSnippet Mobile 介绍

LaTeXSnippet Mobile 是面向 Android 和 iOS 的移动端公式 OCR 识别应用，基于 Capacitor + Vite 构建，前端为纯 Web 技术栈 (HTML/CSS/JS)，后端使用 Java ONNX Runtime 进行本地离线推理。

项目地址与版本

- GitHub: https://github.com/strangelion/LaTeXSnippet_mobile
- 当前版本: **v1.2.2**
- 构建方式: Vite 8 + Capacitor 8 (Android/iOS)
- 许可证: Apache License 2.0

与桌面版的关系

! 全平台生态

LaTeXSnippet Mobile 是独立的应用，不依赖桌面版 LaTeXSnippet。两者共享相同的 MathCraft OCR 模型体系，但移动版使用 Java ONNX Runtime (而非 Python)，所有推理在手机本地完成。桌面版的 Office 插件、Pandoc 导出高级功能、数学工作台等不在移动版中提供。

系统要求

- Android 10+ (推荐 Android 12+)
- 存储空间: 安装包约 120MB，解压后约 350MB (含全部 ONNX 模型)
- RAM: 推荐 6GB 以上 (低端设备可能出现模型加载缓慢或 OOM)
- 可选: 网络连接 (仅用于自动更新检查、AI 整理和 Pandoc WASM 首次加载)

功能与页面结构

移动端采用单页面 SPA 架构，底部导航栏包含 4 个主页面：

识别页面 (OCR)

移动端的核心功能页面，顶部有三个识别模式标签：

- **公式模式**：仅检测和识别数学公式，输出 LaTeX 代码
- **混合模式**：同时识别文本和公式，输出 Markdown + LaTeX 混合文档
- **文字模式**：纯文本 OCR，不检测公式

识别图片的来源有 5 种方式：

1. **拍照识别**：打开系统相机拍摄，支持矩形框选和套索裁剪，支持闪光灯、四角把手拖拽透视矫正、旋转滑块
2. **图片选择**：从系统相册选取图片，自动识别
3. **PDF 识别**：选择 PDF 文件，使用 pdfjs 逐页渲染为图片后识别
4. **手写识别**：打开独立 Canvas 手写画板（支持压感、墨迹平滑、笔/橡皮擦、撤销/重做）
5. **文件拖放**：将图片或 PDF 文件拖入识别区域

编辑器页面

MathLive 所见即所得公式编辑器，支持中文界面和本地化：

- **MathLive 编辑区**：所见即所得数学公式编辑，开启 smartMode 自动识别数学/文本模式，smartFence 自动闭合括号
- **LaTeX 源码区**：编辑器下方显示当前公式的 LaTeX 源码，与编辑区双向同步
- **快速符号工具栏**：希腊字母 (α β π θ ω)、运算符 ($\sqrt{\quad}$ $\int$ $\sum$ $\prod$)、关系符 ($\leq$ $\geq$ $\rightarrow$ $\Rightarrow$ ∞)
- **计算工具栏**：求值、退格、常用 LaTeX 片段插入
- **三态键盘切换**：详见“公式编辑器与键盘策略”章节
- **KaTeX 实时预览**：编辑器内容通过 KaTeX 渲染为公式图片
- **导出菜单**：支持 9 种导出格式

历史记录页面

基于 IndexedDB 的本地存储，支持：

- **滑动操作**：右滑删除、左滑分享/复制/收藏（收藏触发阈值为 50% 位置）
- **收藏筛选**：过滤仅显示收藏项；收藏模式下清空只删除收藏条目
- **点击载入**：点击历史条目自动填入公式编辑器
- **来源标记**：每条历史记录标注来源（拍照/图片/PDF/手写）

设置页面

- **识别引擎**：内置 MathCraft ONNX（默认）或外部 API（OpenAI-compatible）
- **加速模式**：ONNX Runtime 推理加速选项（CPU / NNAPI / GPU）
- **渲染引擎**：KaTeX（轻量快速）或 MathJax（兼容性更好），完全离线
- **视觉皮肤**：7 套视觉主题（Material 蓝 / MIUI 渐变 / iOS 蓝 / 樱花和风 / 黑客矩阵 / 暖咖纸墨 / 极简纤白）
- **AI 整理**：DeepSeek 兼容 API 配置，用于识别结果纠错和格式化
- **多语言**：简体中文 / 繁体中文 / 英文 / 日文 / 韩文
- **开发者面板**：日志查看、版本信息、更新检查

与桌面版的差异

！移动端不是桌面版的简化版

LaTeXSnipper Mobile 基于完全不同的技术栈实现（Java ONNX Runtime + WebView），与桌面版 Python/PyQt 共享模型体系但不共享代码。功能取舍上更侧重移动场景，部分桌面功能不提供，也有桌面版没有的移动专属特性。

移动端没有的桌面功能

- **Office 加载项：** Word / PowerPoint 插件只支持桌面 Windows，移动端不支持
- **数学工作台：** 符号计算、表达式化简、方程求解等桌面专属功能
- **全局截图快捷键：** 移动端不存在桌面端的操作系统级截图快捷方式
- **Pandoc 全格式导出：** AsciiDoc、reStructuredText、OPML 等格式在移动端已移除，保留 9 种常用格式
- **自定义环境变量：** 桌面端通过 MATHCRAFT_* 等环境变量配置模型路径、加速模式等，移动端通过设置 UI 配置
- **多窗口管理：** 桌面端有主窗口、手写窗口、数学工作台等多个独立窗口，移动端为单页面应用

移动端独有的桌面没有的功能

- **相机拍照识别：** 移动端利用手机摄像头拍摄文档和公式
- **套索裁剪：** 拍照后支持自由四边形框选和透视矫正
- **手写画板：** 移动端提供触控手写板（桌面端也有手写窗口，但移动端支持压感触控笔）
- **更多视觉皮肤：** 7 套皮肤（桌面端仅有浅色/深色主题）
- **自动更新检查：** 启动时检测 GitHub Release，移动端弹出系统更新提示
- **离线 PWA：** 可通过 Service Worker 缓存，支持离线使用
- **系统分享集成：** 结果可通过 Android/iOS 系统分享菜单发送到其他应用
- **三态键盘：** MathLive 虚拟键盘 / 系统原生键盘 / 关闭键盘的灵活切换

技术栈差异

桌面端

- Python / PyQt6 / PyInstaller
- Python ONNX Runtime（公式/文字/混合识别）
- Qt WebEngine（编辑器、公式渲染）
- Pandoc 系统命令（文件格式转换）
- Win32 / Carbon / pynput（全局快捷键）

移动端

- Vite 8 / Capacitor 8 / WebView
- Java ONNX Runtime Android（公式/文字/混合识别）
- KaTeX / MathJax（公式渲染）
- Pandoc WASM（格式转换，内嵌在 APK 中）

- pdfjs-dist 4.67 (PDF 渲染)

OCR 识别与输入方式

识别引擎架构

移动端使用纯 Java ONNX Runtime 引擎，所有推理在手机本地完成：

- **公式检测 (YOLOv8)**: mathcraft-mfd.onnx, 输入 [1,3,768,768], 768×768 letterbox
- **公式识别 (TrOCR)**: encoder_model.onnx (DeiT 编码器) + decoder_model.onnx (束搜索解码, beam=3)
- **文字检测 (DBNet)**: ppocrv5_mobile_det.onnx, 最长边 960, stride32 对齐
- **文字识别 (CRNN)**: ppocrv5_mobile_rec.onnx, BGR 48×320 输入, CTC 解码
- **方向检测**: pplcnets_doc_ori.onnx, 0°/90°/180°/270° 自动校正

识别流程：

图片 → 自动旋转 (EXIF + ONNX 方向检测)
→ 模式选择：
公式模式: YOLOv8 检测 → TrOCR 识别 → LaTeX 输出
文字模式: DBNet 检测 → CRNN 识别 → 文本输出
混合模式: YOLOv8 检测 + DBNet 检测 → 区域裁剪合并
→ 公式段: 行分割 → TrOCR 逐行识别 → {aligned} 重组
→ 文字段: CRNN 识别 → 版面输出 (inline: \$... \$, display: \$\$... \$\$)

拍照识别

移动端全屏相机支持：

1. 点击识别页相机按钮启动系统相机，自动对焦
2. 拍照后进入裁剪模式：矩形框选或自由套索（四角把手拖拽）
3. 支持旋转滑块 (0°-360°)
4. 透视矫正：四边形扭曲自动矫正为矩形
5. 确认后自动进入识别
6. 大图自动压缩：>500KB 压缩到最长边 1920px

手写识别

移动端提供触控手写板（独立于桌面端的手写窗口）：

- **工具**: 笔 / 橡皮擦
- **墨迹处理**: 墨迹平滑 (SMOOTH_WINDOW=3)、压感支持 (触控笔)
- **撤销/重做**: 栈式管理，最多保留 60 笔
- **画布调整**: 根据屏幕自动适配
- **颜色跟随**: 笔颜色自动跟随深色/浅色主题
- **识别**: 点击“识别”按钮，导出为 PNG 后走当前识别引擎
- **默认模式**: 从图片切换到手写时自动设为混合模式

外部 API 识别

移动端支持连接外部 OCR API (OpenAI-compatible 协议)：

- **内置预设**: PaddleOCR-VL、硅基流动 Qwen2.5-VL、Gemini 2.5 Flash、MinerU 原生
- **连接测试**: 设置页中测试配置的有效性
- **AI 整理**: 配置 DeepSeek 兼容 API，对识别结果进行纠错和格式化

公式编辑器与键盘策略

MathLive 编辑器配置

移动端使用 MathLive 0.104 作为所见即所得公式编辑器，核心配置：

- `mathVirtualKeyboardPolicy = 'manual'` — 不自动弹出虚拟键盘
- `smartMode = true` — 自动识别输入为数学模式还是文本模式
- `smartFence = true` — 自动闭合未配对的括号、花括号、方括号
- 中文本地化：通过 `MathfieldElement.strings` 注入完整中文 UI 文本
- 通过 `new MathfieldElement()` 创建而非 `<mathlive-field>` 标签（避免部分 WebView 不注册自定义元素）

三态键盘切换

针对移动端输入场景的特殊设计：

编辑器顶部的键盘按钮支持三种模式循环切换：

1. **关闭（默认）**：不弹出任何键盘，适合纯符号工具栏输入
2. **MathLive 虚拟键盘**：弹出 MathLive 内置数学符号键盘，包含希腊字母、运算符、关系符等专用键位；通过 `toggleVirtualKeyboard` 命令唤起
3. **系统原生键盘**：使用 `<textarea>` 代理接收焦点并触发 IME，输入事件转发到 MathLive 的 `insert()` 方法。系统键盘弹出时自动隐藏底部公式符号工具栏，腾出编辑空间

！系统键盘实现机制

MathLive v0.104 中 `sandboxed` 策略被映射为 `manual`，无法在 Android WebView 中触发系统 IME。移动版采用 `offscreen <textarea>` 代理方案：

- 代理以 `position: fixed` 覆盖编辑区接收焦点，触发系统键盘
- `input` 事件 → `mathField.insert()` 转发输入
- Shadow DOM 的 `contentEditable` 在系统键盘模式下被禁用，配合 `capture-phase focus guard` 防止 MathLive 抢焦点
- 零宽空格占位防止 Android 在内容清空时自动隐藏 IME
- 处理 Backspace、Enter、方向键和 CJK 输入法合成事件

符号面板

编辑器底部提供快速符号工具栏：

- **希腊字母**： $\alpha \beta \gamma \pi \theta \mu \sigma \omega \delta \epsilon \varphi$
- **运算符**： $\sqrt{\quad} \int \sum \prod \partial \infty$
- **关系符**： $\leq \geq \neq \approx \rightarrow \Rightarrow \leftrightarrow$
- 点击符号直接在光标位置插入对应 LaTeX

渲染引擎

- **KaTeX（默认）**：轻量快速（265KB），HTML 输出模式原生支持中文混合显示，无中文字体渲染问题，资源完全离线
- **MathJax（可选）**：兼容性更好（1.1MB + 23 字体文件），启动阶段 splash 预加载，完全离线（无 CDN 依赖）

导出与分享

9 种导出格式

格式	转换路径	说明
PNG	KaTeX → SVG → Canvas	高清公式图片，含中文字体回退
SVG	KaTeX → SVG	矢量公式图片
LaTeX	Pandoc WASM	.tex 文件格式
MathML	Pandoc WASM	数学标记语言
Markdown	Pandoc WASM	markdown+tex_math_dollars
HTML	Pandoc WASM	网页格式
Typst	纯 JS 转换器	200+ 符号映射 + 结构转换，不依赖 Pandoc
Word	Pandoc WASM	.docx 格式
Plain Text	Pandoc WASM	纯文本

! Typst 导出说明

Typst 导出使用纯 JS 转换器，包含 200+ LaTeX→Typst 符号映射（希腊字母、运算符、箭头、关系符、函数名）和结构转换（分数、根式、矩阵、cases、text 等）。不经过 Pandoc WASM，因为 pandoc-wasm 的 WASM 二进制对完整 LaTeX math mode 支持受限。

分享功能

- **系统分享：**通过 Capacitor Share API 将识别结果分享到其他应用
- **文件分享降级：**当 Capacitor Share 在某些 Android 版本上传 base64 文件失败时，自动触发下载而非弹系统分享（避免“没有应用可执行此操作”的错误）
- **复制到剪贴板：**一键复制 LaTeX 源码
- **发送到编辑器：**将识别结果填入公式编辑器，修改后再导出

Pandoc WASM 说明

⚠ 移动端 Pandoc 导出注意事项

Pandoc WASM 二进制（约 56MB）内嵌在 APK 中，首次导出时需加载 WASM 模块。Android WebView 运行时有点限制：

- wasi_snapshot_preview1 模块在部分 WebView 中缺失，已通过 @bjorn3/browser_wasi_shim 垫片修复
- WASM 加载路径已处理为 Capacitor 兼容路径，绕开 vite-plugin-wasm 生成的路径问题
- Pandoc WASM 不需要网络连接（WASM 二进制已内嵌）

移动端常见问题

模型加载失败或 OOM（内存不足）

现象：启动时进度条卡住，或识别时报“内存不足”直接闪退。

原因：全部 ONNX 模型约 350MB 解压后大小，`encoder_model.onnx` 一个文件就 87MB。低端设备（4GB 以下 RAM）可能在同时加载多个模型时 OOM。

解决：

- 在设置页 **加速模式** 中切换到较低加速级别
- 避免在识别时同时运行大型应用（如游戏、视频编辑）
- 如果频繁 OOM，尝试减少识别模式切换频率（避免频繁加载/卸载不同模型）
- 应用已启用 `largeHeap` (AndroidManifest.xml)，但仍有物理限制
- 模型加载失败时会自动清理 session 并重试

拍照识别卡顿或无法对焦

现象：相机启动慢、对焦困难或拍照后裁剪界面卡顿。

原因：移动端相机使用系统 WebView 的 `getUserMedia` API，其性能和对焦能力取决于 Android WebView 实现和系统相机 HAL。

解决：

- 优先使用系统自带相机应用拍照，然后从相册选择图片识别
- 确保手机有足够光线的环境
- 拍照时保持手机稳定
- 大图自动压缩机制会处理 >500KB 的图片，无需手动压缩

手写板延迟或误触

现象：手写识别延迟高，或手指/触控笔划线时出现误触。

原因：Canvas 触控事件在 WebView 中可能不如原生应用灵敏，某些廉价的触控屏或贴膜也可能导致触控不准。

解决：

- 使用触控笔而非手指书写，效果更佳
- 手写板支持墨迹平滑 (`SMOOTH_WINDOW=3`)，轻微抖动会被自动过滤
- 如果橡皮擦不方便，可使用“清空”按钮一键清除
- 手写完成后点击“识别”按钮，不走自动识别（减少误触发）

导出失败（Pandoc WASM 相关问题）

现象：导出 Markdown/HTML/Word 等格式时卡住或报错。

原因：Pandoc WASM 模块加载失败，或 WASM 二进制损坏/缺失。

解决：

- PNG、SVG、Typst 三种格式 **不依赖 Pandoc WASM**，可直接导出
- 如果 Pandoc 依赖的格式导出失败，先尝试使用不依赖 Pandoc 的格式
- 重启应用后重试
- 如果问题持续，尝试清除应用数据后重新安装 APK
- 检查开发者日志面板是否有 WASM 相关错误信息

AI 整理连接失败

现象：点击“AI 整理”后提示连接失败或超时。

原因：AI 整理需要连接 DeepSeek 兼容 API，未配置正确的 API Key 或 Base URL，或网络不可用。

解决：

- 前往设置页 **AI 整理** 配置 API Endpoint 和 API Key
- 确认手机网络连接正常
- 如果使用自定义 API，确保接口兼容 DeepSeek 的聊天补全格式
- AI 整理为可选功能，不配置不影响识别和导出

更新检查失败

现象： 启动时检查更新失败或不弹更新提示。

原因： 更新检查访问 GitHub Releases API，国内用户可能无法直接访问。

解决：

- 确保手机网络可访问 GitHub
- 如果使用代理/VPN，确认已正确配置
- 更新检查失败不影响核心识别功能
- 手动去 GitHub Releases 页面下载最新 APK

外部 API 识别配置问题

现象： 切换到外部引擎后识别失败或提示配置有误。

原因： 外部 API 需要正确的协议类型、Base URL、模型名和可选的 API Key。

解决：

- 设置页提供预设 (PaddleOCR-VL、硅基流动、Gemini、MinerU 原生)，一键填写常用配置
- 使用预设后再根据需要微调 Base URL 和模型名
- 配置完成后点击 **测试连接** 确认配置有效
- 移动端的外部 API 设置与桌面端类似，但预设列表可能不同

KaTeX / MathJax 渲染异常

现象： 公式在预览中显示为源码或渲染错乱。

原因： 渲染引擎切换后未完全加载，或公式包含特定语法导致渲染失败。

解决：

- 在设置页切换 **公式渲染引擎** 尝试另一种引擎
- KaTeX 为默认引擎，轻量快速，绝大多数场景足够
- 如果公式含有 KaTeX 不支持的语法 (如某些 LaTeX 宏包)，切换到 MathJax 通常可以解决
- renderBlock 智能判断显示/内联模式：多行环境 ($\begin{aligned}$ 、 $\$$....\$$) 按逻辑块分组渲染，不逐行切分
- 不含 $\backslash$ 的表达式 (如 $x^2 + y^2 = z^2$) 会被正确渲染 (兜底逻辑)

历史记录丢失

现象： 重启应用后历史记录消失。

原因： 历史记录存储在 IndexedDB 中，清除应用数据或存储空间不足时可能丢失。

解决：

- IndexedDB 数据与应用绑定，卸载应用会清除所有历史记录
- 重要的识别结果建议提前复制到其他地方保存
- 不要通过系统设置清除应用数据 (会一并删除 IndexedDB)
- 如果存储空间极度不足，Android 系统可能自动清除 IndexedDB

PDF 识别页数限制

现象： 选择 PDF 后只识别了前面部分页面。

原因： 移动端限制 PDF 最多识别 100 页，识别时逐页渲染处理较慢。

解决：

- 超过 100 页的 PDF 建议分卷处理或将长文档分段导出
- PDF 使用 pdfjs-dist 4.67 在 WebView 中逐页渲染为图片后识别，渲染目标 768px
- 处理器密集，长文档建议在充电状态下操作

移动端特定兼容性问题

X WebView 兼容性注意事项

以下问题已在开发中解决，但如果遇到异常，可参考：

- **MathLive 自定义元素：** `<mathlive-field>` 在部分 WebView 中不注册，改用 `new MathfieldElement()` 创建
- **虚拟键盘策略：** 设置为 'manual'，不自动弹出；通过键盘按钮手动唤起
- **按钮事件：** 必须用 `pointerdown + stopPropagation`，`click` 在 WebView 中不可靠
- **COOP/COEP 头：** 已通过 Capacitor 和 Vite 配置，确保 WASM 加载正确
- **Service Worker：** 注册 `/sw.js` 用于离线缓存
- **PWA 安装：** 支持 `beforeinstallprompt` 事件，可添加到主屏幕
- **中文渲染：** SVG `foreignObject` 显式指定 "Noto Sans CJK SC", "Microsoft YaHei" 中文字体回退，避免中文方框问题

移动端数据目录

! 关注存储占用

移动端所有文件（模型、日志、历史记录）均存储在应用内部目录。ONNX 模型约 220MB，Pandoc WASM 约 56MB，MathJax 字体约 20MB。强烈不建议通过系统设置清除应用数据，否则所有模型和离线功能需要重新下载。

应用数据目录位置

- **ONNX 模型**: `public/models/`（内嵌在 APK 中，无需下载）
- **Pandoc WASM**: `public/vendor/pandoc/`（内嵌在 APK 中）
- **历史记录**: IndexedDB（应用内部存储）
- **日志**: `localStorage`（开发者日志面板查看）
- **设置**: `localStorage` + `Android SharedPreferences`

APK 体积组成

- Pandoc WASM: 56MB
- ONNX 模型: 220MB（APK 内压缩）
- KaTeX + MathLive + PDF.js: 15MB
- MathJax 可选: 20MB
- APK 总大小约 280MB，安装解压后约 400MB

第四卷 • MathCraft 内部模型介绍

MathCraft OCR 项目介绍

MathCraft OCR 是 LaTeXSnippet 内置的本地公式、文本与混合文档识别引擎，基于 ONNX Runtime 实现本地推理。模型缓存完整后，截图、图片和 PDF 页面识别都可以离线完成。

项目地址：

- GitHub: <https://github.com/SakuraMathcraft/MathCraft-Models>
- PyPI: <https://pypi.org/project/mathcraft-ocr/>

主要特点：

- **本地离线识别：** 模型安装完成后无需网络连接，识别在本地完成
- **ONNX 运行时：** 基于 ONNX Runtime，支持 CPU 与 CUDA GPU 路径
- **多任务识别：** 支持公式、纯文字、混合文档与 PDF 文档识别
- **自动缓存修复：** 模型文件首次使用时自动下载并缓存；缺失或损坏时会按需修复

安装方式：

```
pip install "mathcraft-ocr[cpu]"
# 或 GPU 环境:
pip install "mathcraft-ocr[gpu]"
```

单独使用 PyPI 包时，请在干净环境中只选择一种 ONNX Runtime 后端；不要同时安装 onnxruntime 和 onnxruntime-gpu。在 LaTeXSnippet 中使用时，通常通过依赖向导自动安装。

命令行使用：

```
# 检查模型缓存状态
python -m mathcraft_ocr models check

# 检查运行环境并预热模型
python -m mathcraft_ocr doctor --provider auto
python -m mathcraft_ocr warmup --profile mixed --provider auto

# 识别图片中的公式
python -m mathcraft_ocr ocr formula.png --profile formula --provider auto --json

# 混合文档识别并输出 Markdown
python -m mathcraft_ocr ocr page.png --profile mixed --provider auto --output result.md
```

在 LaTeXSnippet 中使用： 在设置页的“选择识别模型”中选择 **内置模型** 即可启用本地 MathCraft OCR；识别类型可选公式、混合（文字+公式）或纯文字。如需强制使用 CPU 推理（例如避免 GPU 兼容性问题），请参考下方的环境变量设置。

模型集与识别配置

当前 mathcraft-ocr PyPI 包版本为 0.2.4。模型权重使用 MathCraft Models v1.0.0 发布集，包含 **4 个 ONNX 模型**：

- mathcraft-formula-det — 数学公式区域检测 (Formula Detection)
- mathcraft-formula-rec — 公式到 LaTeX 识别 (Formula Recognition)
- mathcraft-text-det — 快速多语言文本检测 (Text Detection)
- mathcraft-text-rec — 快速多语言文本识别 (Text Recognition)

！识别流程

MathCraft 采用 **先检测后识别** 的流水线架构：图片输入 → 公式区域检测 → 公式识别 + 文本检测 → 文本识别 → 结构化输出。这种设计确保公式区域的识别优先级高于普通文本，避免公式被误判为正文。

三种识别模式 (Profiles):

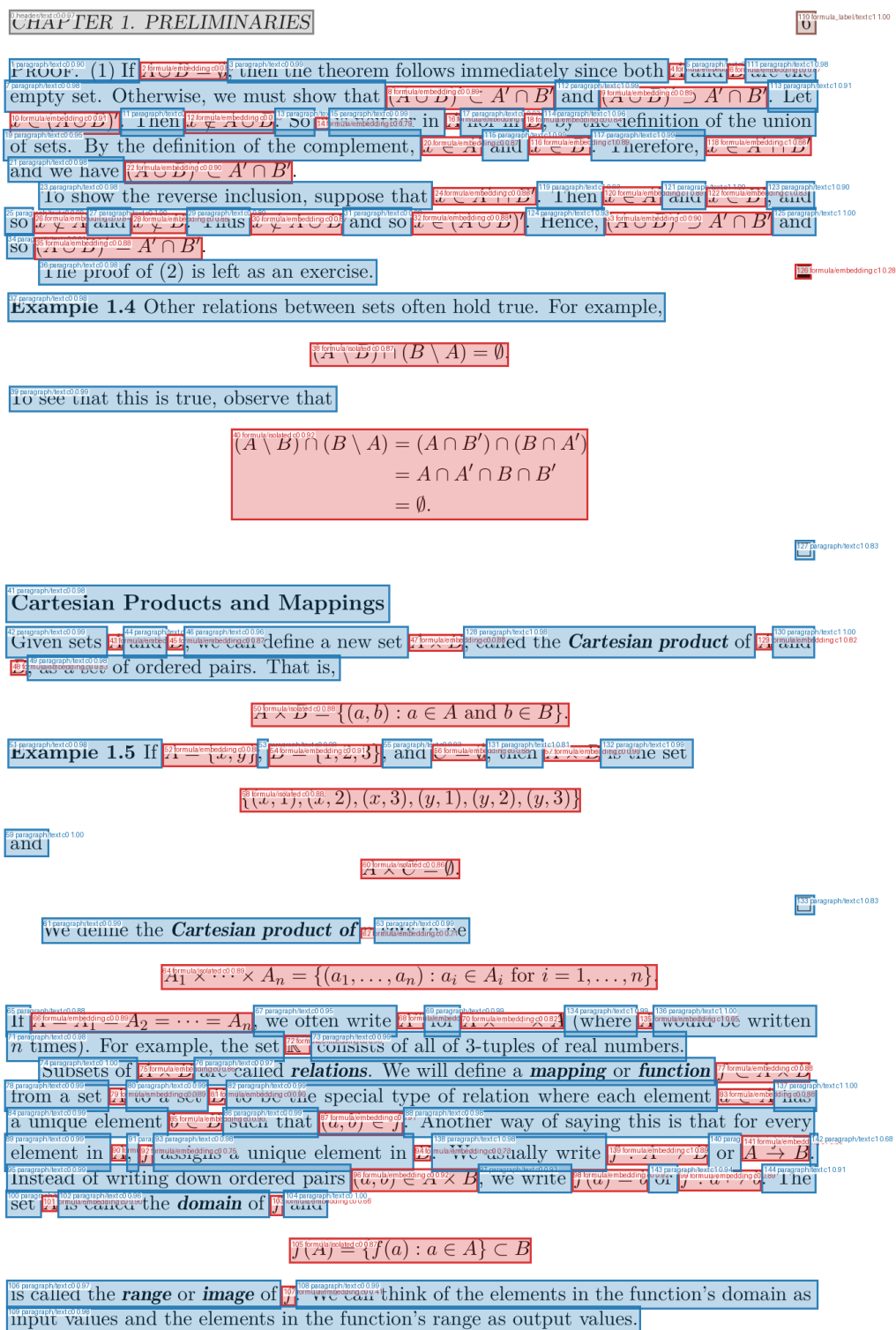
- `formula` — 公式截图 → LaTeX 公式文本
- `text` — 纯文本 OCR → 文本
- `mixed` — 文本+公式混合文档 → Markdown 结构化文本

识别效果展示

以下示例来自 MathCraft 的结构化块输出。彩色框标注了检测到的角色、顺序、分栏信息和置信度分数。

英文数学论文: **Abstract Algebra (p.18)**

公式密集的英文数学论文页面，包含大量行内公式和展示公式。



英文期刊: Dynamics Journal (p.5)

以展示公式为主的期刊页面, 包含公式编号、标签、页眉和页码。

where

$$\begin{aligned} \hat{F}_{11}(u_n, v_n, \delta) &= -\frac{(1+b)x^*\delta}{2a} - \frac{a \sin x^* u_n^2}{2} - \frac{(1+b)x^*\delta^2}{4a^2} \\ &\quad - \frac{a \cos x^* u_n^3}{6} - \frac{(1+b)x^*\delta^3}{8a^3} + O(\|(u_n, v_n, \delta)\|^4), \\ \hat{F}_{12}(u_n, v_n, \delta) &= \frac{(-x^*b^2 + 2a \sin x^* - 2x^*b - x^*)\delta}{2a} \\ &\quad + \frac{a(b-1) \sin x^* u_n^2}{2} + \frac{(1+b)^2 x^* \delta^2}{4a^2} - \cos x^* \delta u_n \\ &\quad + \frac{a(b-1) \cos x^* u_n^3}{6} + \frac{(1+b)^2 x^* \delta^3}{8a^3} + \frac{\sin x^* \delta u_n^2}{2} \\ &\quad + O(\|(u_n, v_n, \delta)\|^4). \end{aligned}$$

Consider the following system

$$\begin{pmatrix} u_{n+1} \\ v_{n+1} \\ \delta_{n+1} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_n \\ v_n \\ \delta_n \end{pmatrix} + \begin{pmatrix} \hat{F}_{11}(u_n, v_n, \delta_n) \\ \hat{F}_{12}(u_n, v_n, \delta_n) \\ 0 \end{pmatrix}, \quad (15)$$

where

$$\begin{aligned} a_{11} &= -1 + a \cos x^*, & a_{12} &= 1, \\ a_{13} &= -\frac{(1+b)x^*}{2a}, & a_{21} &= a(1-b) \cos x^*, \\ a_{22} &= -b, & a_{23} &= \frac{-2a \sin x^* + x^*(b+1)^2}{2a}. \end{aligned}$$

$$\begin{aligned} \hat{F}_{11}(u_n, v_n, \delta_n) &= -\frac{a \sin x^*}{2} u_n^2 - \frac{(1+b)x^*\delta_n^2}{4a^2} \\ &\quad - \frac{a \cos x^*}{6} u_n^3 - \frac{(1+b)x^*\delta_n^3}{8a^3} \\ &\quad + O(\|(u_n, v_n, \delta_n)\|^4), \\ \hat{F}_{12}(u_n, v_n, \delta_n) &= \frac{a(b-1) \sin x^*}{2} u_n^2 + \frac{(1+b)^2 x^* \delta_n^2}{4a^2} \\ &\quad - \cos x^* \delta_n u_n + \frac{a(b-1) \cos x^*}{6} u_n^3 \\ &\quad + \frac{(1+b)^2 x^* \delta_n^3}{8a^3} + \frac{\sin x^*}{2} \delta_n u_n^2 \\ &\quad + O(\|(u_n, v_n, \delta_n)\|^4). \end{aligned}$$

We construct an invertible matrix

$$T = \begin{pmatrix} -\frac{a \sec x^*}{1} & -\frac{1}{b-1} & \frac{\sin x^*}{2a \cos x^* - 2b - 2} \\ a & 1 & e \\ 0 & 0 & 1 \end{pmatrix},$$

where

$$e = \frac{-a(a \sin x^* - (1+b)x^*) \cos x^* + 2a \sin x^* - x^*(1+b)^2}{2a(-b-1+a \cos x^*)},$$

and use the translation

$$\begin{pmatrix} u_n \\ v_n \\ \delta_n \end{pmatrix} = T \begin{pmatrix} \tilde{x}_n \\ \tilde{y}_n \\ \tilde{\delta}_n \end{pmatrix}.$$

then the system (15) becomes

$$\begin{pmatrix} \tilde{x}_{n+1} \\ \tilde{y}_{n+1} \\ \tilde{\delta}_{n+1} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -b + a \cos x^* & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{x}_n \\ \tilde{y}_n \\ \tilde{\delta}_n \end{pmatrix} + \begin{pmatrix} \tilde{F}_{11}(\tilde{x}_n, \tilde{y}_n, \tilde{\delta}_n) \\ \tilde{F}_{12}(\tilde{x}_n, \tilde{y}_n, \tilde{\delta}_n) \\ 0 \end{pmatrix}, \quad (16)$$

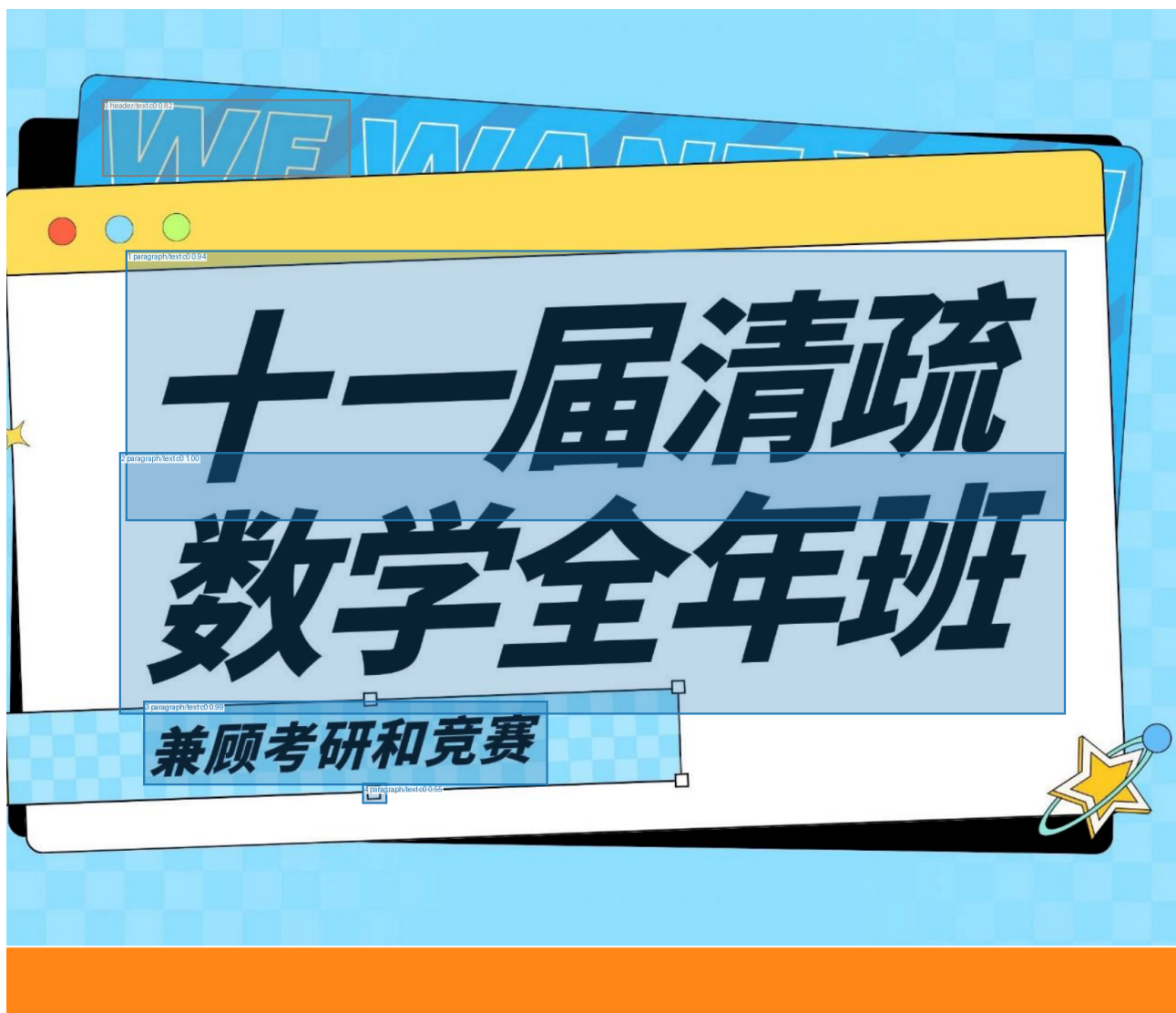
where

$$\begin{aligned} \tilde{F}_{11}(\tilde{x}_n, \tilde{y}_n, \tilde{\delta}_n) &= -\frac{\tan x^* \sec x^* \tilde{x}_n^2}{2a} - \frac{\tan x^* \tilde{x}_n \tilde{y}_n}{b-1} + \frac{\sin x^* \tan x^* \tilde{\delta}_n \tilde{x}_n}{2 \cos x^* (a-2b-2)} \\ &\quad - \frac{a \sin x^* \tilde{y}_n^2}{2(b-1)^2} + \frac{a \sin^2 x^* \tilde{y}_n \tilde{\delta}_n}{2(b-1)(a \cos x^* - b - 1)} \\ &\quad - \frac{\tilde{\delta}_n^2}{8(a \cos x^* - b - 1)^2 a^2} \left((2x^*(b+1)^3 + a^3 \sin^3 x^* \right. \\ &\quad \left. - 4ax^*(b+1)^2 \cos x^* + 2a^2 x^*(b+1) \cos^2 x^*) \right) \\ &\quad + \frac{\sec^2 x^* \tilde{x}_n^3}{6a^2} + \frac{\sec x^* \tilde{x}_n^2 \tilde{y}_n}{2a(b-1)} - \frac{\tan x^* \tilde{x}_n^2 \tilde{\delta}_n}{4a(a \cos x^* - b - 1)} \\ &\quad + \frac{\tilde{x}_n \tilde{y}_n^2}{2(b-1)^2} - \frac{\sin x^* \tilde{x}_n \tilde{y}_n \tilde{\delta}_n}{2(b-1)(a \cos x^* - b - 1)} + \frac{a \cos x^* \tilde{y}_n^3}{6(b-1)^3} \\ &\quad - \frac{a \cos x^* \sin x^* \tilde{y}_n^2 \tilde{\delta}_n}{4(b-1)^2(a \cos x^* - b - 1)} + \frac{a \cos x^* \sin^2 x^* \tilde{y}_n \tilde{\delta}_n^2}{8(b-1)(a \cos x^* - b - 1)^2} \\ &\quad - \frac{\tilde{\delta}_n^3}{48(a \cos x^* - b - 1)^3 a^3} \left(6a^3 x^*(b+1) \cos^3 x^* \right. \\ &\quad \left. - 18a^2 x^*(b+1)^2 \cos^2 x^* - 6x^*(b+1)^4 \right. \\ &\quad \left. + a(a^3 \sin^3 x^* + 18x^*(b+1)^3 \cos x^*) \right) \\ &\quad + O(\|(\tilde{x}_n, \tilde{y}_n, \tilde{\delta}_n)\|^4), \end{aligned}$$

$$\begin{aligned} \tilde{F}_{12}(\tilde{x}_n, \tilde{y}_n, \tilde{\delta}_n) &= \frac{(b-1) \tan x^* \sec x^* \tilde{x}_n^2}{2a} + \tan x^* \tilde{x}_n \tilde{y}_n \\ &\quad - \frac{\tilde{\delta}_n \tilde{x}_n}{2a(a \cos x^* - b - 1)} (ab \sin x^* \tan x^* - a \cos x^* \\ &\quad + 2b + 2 - a \sec x^*) \\ &\quad + \frac{a \sin x^* \tilde{y}_n^2}{2b-2} - \frac{\tilde{x}_n \tilde{y}_n^2}{2b-2} - \frac{a \cos x^* \tilde{y}_n^3}{6(b-1)^2} \\ &\quad - \frac{\tilde{\delta}_n \tilde{y}_n}{2(b-1)(a \cos x^* - b - 1)} \left(-a \cos^2 x^* + (2b+2) \cos x^* \right. \\ &\quad \left. + a(b \sin^2 x^* - 1) \right) \\ &\quad + \frac{\tilde{\delta}_n^2}{8a^2(a \cos x^* - b - 1)^2} \left((a^3 b \sin^3 x^* + 2x^*(b+1)^4 \right. \\ &\quad \left. + 4(a \sin x^* - x^*(b+1)^2)a(b+1) \cos x^* - a^3 \sin x^* \right. \\ &\quad \left. - 3(a \sin x^* - \frac{2x^*(b+1)^2}{3})a^2 \cos^2 x^* \right) \end{aligned}$$

中文讲义：Lecture Note (p.1)

中文数学文档页面，包含文本与公式混合排版。



第十一届清疏大学生数学竞赛班讲义

非数学类 & 数学类共用

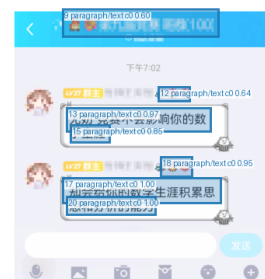
作者：清疏数学

组织：清疏大学

时间：July 7, 2024

版本：2.0

作者联系方式：QQ:937821418



本讲义将自动生成和 cctalk 用户绑定的 ID, 如果你的 ID 多次出现在公开环境, 我们会追究法律责任

仕。

极限与级数：Limits and Series (p.1)

稀疏的标题/封面式页面，用于验证版面分析稳定性。

1 paragraph/text c0 0.99

Problem Books in Mathematics

1 paragraph/text c0 0.99

Ovidiu Furdui

2 paragraph/text c0 1.00

3 paragraph/text c0 1.00

4 paragraph/text c0 1.00

5 paragraph/text c0 1.00

6 paragraph/text c0 1.00

7 paragraph/text c0 1.00

Limits, Series,
and Fractional
Part Integrals

8 paragraph/text c0 0.98

Problems in Mathematical Analysis



10 paragraph/text c0 0.94

Springer

性能参考

以下为本地 block_layout_regression_v4 基准测试数据 (CUDA 环境):

- 测试页数: 10
- 总块数: 495
- 总字符数: 21,417
- Markdown 行数: 304
- **平均每页耗时: 8.34 秒**
- 最快页面: 1.33 秒
- 最慢页面: 18.53 秒

! 稳定性保障

MathCraft OCR 的设计原则:

- 仅依赖 ONNX Runtime, 无 PyTorch 推理依赖
- 基于清单 (manifest) 的文件校验与缓存修复
- 断点续传支持 (适用于慢速/不稳定网络)
- 公式检测优先于文本 OCR
- 结构化块输出: 标题、段落、展示公式、页眉、页码、分栏

环境变量设置指南

许多解决方案需要设置环境变量 (如 MATHCRAFT_FORCE_ORT_CPU、MATHCRAFT_HOME 等):

Windows (永久生效):

1. Win+R → 输入 `sysdm.cpl` → 高级 → 环境变量
2. 在"用户变量"中新建, 变量名填入上面给出的名称, 值填入 `1` 或路径

Windows (仅当前终端):

在 PowerShell 中执行: `$env:MATHCRAFT_FORCE_ORT_CPU = "1"`

Linux / macOS (永久生效):

在 `~/.bashrc` 或 `~/.zshrc` 末尾添加:
`export MATHCRAFT_FORCE_ORT_CPU=1`
然后执行 `source ~/.bashrc`

Linux / macOS (仅当前终端):

`export MATHCRAFT_FORCE_ORT_CPU=1`

常用环境变量一览:

- MATHCRAFT_FORCE_ORT_CPU — 设为 1 强制使用 CPU 推理 (禁用 GPU)
- MATHCRAFT_HOME — 指定 MathCraft 数据目录 (模型缓存、配置等)
- HTTP_PROXY / HTTPS_PROXY — 设置代理服务器地址 (用于模型下载)
- MATHCRAFT_LOG_LEVEL — 日志级别 (DEBUG / INFO / WARNING / ERROR)
- LATEXSNIPPER_FORCE_LINUX_GRAPHICS_FALLBACKS — Linux 下强制启用 Qt/WebEngine 软件渲染兜底
- LATEXSNIPPER_DISABLE_LINUX_GRAPHICS_FALLBACKS — Linux 下禁用图形兜底, 尝试原生 Qt/GPU 路径
- LATEXSNIPPER_SHOW_CONSOLE — Windows 打包版调试时显示或隐藏运行日志窗口