

FlowWrite 是一款专为 AI 辅助文字创作而生的可视化工作流编辑器。本文档聚焦于产品核心功能设计，探索文本处理与创意写作的全新交互范式。

目录

产品愿景 .....	1
缘起 .....	1
理念 .....	1
核心功能 .....	1
workflow 功能模块的总体设计 .....	2
交互方式 .....	2
组件设计 .....	2

产品愿景

FlowWrite 致力于成为 AI 时代文字创作者的得力助手，通过直观的可视化工作流，让复杂的 AI 赋能文本处理变得简单而优雅。

缘起

在 AI 写作实践中，创作者往往需要在多个工具间频繁切换，prompt 的调试、文本的迭代、创意的碰撞都充满了重复性的复制粘贴操作。FlowWrite 诞生于这样的痛点：我们渴望一个场景覆盖足够大的创作范式，让文字工作者能够专注于内容本身。

理念

AI 可以大幅提高写作的效率，但真正的创作往往需要反复打磨和人工干预。我们借鉴 ComfyUI 的节点化思想，为文本创作领域打造一个灵活、可定制的可视化工作流系统。我们希望以这样的方式，让文字创作者享受“人剑合一”的创作体验。

核心功能

- **节点化文本处理**：将复杂的 AI 流程操作拆解为（低耦合的）基于节点的工作流
- **实时依赖解析**：智能处理节点间的数据流转与依赖关系
- **创作友好设计**：为文字工作者的使用习惯深度优化

# workflow 功能模块的总体设计

基于对交互方式的设想，我们发现具体组件的设计可以是低耦合而优雅的。

## 交互方式

一个节点代表一次 LLM 的 api 调用，也就是一次对话。一个节点的输入可以是多个，输出是唯一的，但可以作为多个节点的输入。那么一个节点的多个输入该如何被应用到这个节点的对话中？其实基于定制化的思路，基本上是这样形态：

[ 第一个输入 ]

请参照上面的要求，将以下三次 LLM 对话的输出总结成一个：

[ 第二个输入 ]

[ 第三个输入 ]

[ 第四个输入 ]

也就是说，某个节点的输出在还没有具体从一次 LLM 对话中产生的时候，就已经预订好了它在另一个节点的输入中所扮演的角色，这个设计催生了一些前期设想，如占位文本块，或是文本块引用的概念，前者使得一个节点可以不依赖于与其它节点的连接关系来定义，后者使得非节点输出（也就是不是从 AI 侧获取而是由用户自定义的文本）的文本块也可以共享状态（改一处其它地方也会改）

基于这类设计，工作流的运行主要是类似拓扑解析依赖的过程，当然后续可以加一些对有向环支持的花活。

注意到，前面提到的这两种设想似乎可以合并成一个虚拟文本块的概念。而且要加一个可以在工作流中暂时 freeze 一个虚拟文本块的功能，使得这个虚拟文本块在后续的使用中表现的就像一个普通文本块（这个可能会成为一个很常用的功能）。

不过普通文本块的引用功能还是禁掉吧，感觉没啥意义。

综合来看，状态设计不会很复杂。

## 组件设计