华泰证券 HUATAI SECURITIES

大模型辅助编程手册

华泰研究

研究员 林晓明
SAC No. S0570516010001 linxiaoming@htsc.com
SFC No. BPY421 +(86) 755 8208 0134

研究员 何康，PhD
SAC No. S0570520080004 hekang@htsc.com
SFC No. BRB318 +(86) 21 2897 2202

联系人 沈洋
SAC No. S0570123070271 shenyang023029@htsc.com
+(86) 21 2897 2228

2025 年 2 月 26 日 | 中国内地 深度研究

人工智能 87：大模型辅助编程手册

本文是大模型辅助编程的使用指南，详细介绍大模型在日常编程与项目开发过程中的辅助编程应用及其相应部署流程。本文从大模型嵌入开发环境的程度将不同辅助编程工具分为两类：插件类以及 IDE 类，以 Github Copilot、Cline、CodeGPT、MarsCode 和 Codeium，与 Cursor、Windsurf 和 MarsCode IDE 分别为例，介绍两类工具的部署流程及具体功能。此外，本文以 Cursor 的 Composer 功能为代表，展示其在量化策略编写方面的实践案例。

插件类辅助编程工具：以 Github Copilot 为代表的代码补全智能助手

Github Copilot 由微软与 OpenAI 联手开发，是一款老牌代码智能补全插件。Copilot 支持多终端部署，依托 Github 优质的编程生态，辅助编程功能丰富且稳定，是代码智能补全的代表工具。此类工具还包括 CodeGPT、MarsCode 和 Codeium，其中，CodeGPT 核心特色是多模型支持，支持 DeepSeek 等主流大模型 API 以及本地 Ollama 部署模型的接入；Codeium 支持包含 VSCode 和 PyCharm 在内的 17 款编辑器，功能逻辑简单清晰；MarsCode 以豆包代码模型为基座，提供智能代码补全、代码生成等丰富功能，界面简洁直观。

插件类辅助编程工具：独树一帜的自动化编程插件 Cline

Cline 是插件类工具中为数不多可实现完全自动化编程的工具。不同于传统编程助手仅提供代码补全功能，Cline 通过源代码 AST（抽象语法树）分析和正则表达式搜索实现项目级代码重构，能主动创建和编辑文件、探索大型项目以及执行终端命令等。Cline 较为擅长处理复杂任务，例如根据自然语言指令创建符合项目结构的 Vue 组件或 Python 脚本，具备精准的上下文理解能力。此外，相比其他插件，Cline 在国产化适配（原生支持 DeepSeek 等国产模型）和企业级扩展（MCP 私有化部署）方面具有独特优势。

IDE 类辅助编程工具：扩展功能边界，“超越”辅助编程插件局限

Cursor 是 IDE 类辅助编程工具的代表，旨在“超越”插件类工具的局限。Cursor 支持对整个项目代码库的深度理解和索引，能够基于全局上下文提供代码建议、优化和重构。Cursor 还支持多文件编辑和全局重构，能够自动识别相关文件并提供优化建议。除 Cursor 外，Windsurf 和 MarsCode 分别代表了 AI 编程工具另外两个探索方向。Windsurf 以其 Agent 驱动的智能编程模式为核心，支持多步骤、多工具协同工作，能够自动维护上下文状态并智能规划任务。相比之下，MarsCode 则是一款轻量化的智能编程工具，仅支持云端浏览器使用，界面简洁直观，使用门槛低。

推荐应用：VSCode+Cline+DeepSeek 与 Cursor

综合功能与实际体验，VSCode+Cline+DeepSeek 与 Cursor 是较为推荐的辅助编程应用配置。相比于其他工具，Cline 在多方面占据优势，包括用户自行嵌入模型 API、辅助编程高度自动化、外部工具调用等。对于 Cursor 而言，以 IDE 为产品形态更好地扩充了大模型辅助编程的权限，展现出“全局视角”下的智能交互，这使得 Cursor 不再仅是一个代码生成工具，更是一个能够深度融入编码工作流的全能编程助手。

风险提示：大模型是海量数据训练获得的产物，输出准确性可能存在风险；不同大模型辅助编程工具效果存在差距，对于大模型生成的代码，需要谨慎参考；大模型辅助编程工具功能及稳定性可能受到版本切换影响。

正文目录

引言 **5**
推荐应用一：VSCode + Cline + DeepSeek 6
推荐应用二：Cursor 7
插件类 **8**
Github Copilot：老牌代码智能补全插件 8
部署 9
基础功能：代码补全、代码解释、生成测试等 10
核心功能：联想编程 13
CodeGPT：多模型及开源模型支持 14
部署 14
基础功能：对话、代码补全、代码生成、即时检索等 15
MarsCode：轻量化编程助手插件 18
部署 18
基础功能：代码补全、代码预测推荐、智能错误修正等 18
Codeium：多编辑器支持编程助手插件 19
部署 19
基础功能 20
Cline：新兴开源自动化编程插件 21
部署 21
基础功能：运行终端命令、创建修改文件、使用浏览器等 22
核心功能：自动化编程（Task）与 MCP 24
插件类工具对比 25
IDE 类 **26**
Cursor：辅助编程工具的集大成者 26
基础功能：代码补全、行内编辑、对话 27
核心功能：Composer 自动化编程 27
Windsurf：AI Flow 范式下的新兴 IDE 29
MarsCode：轻量化云端 IDE 30
大模型辅助编程实践 **32**
以 Cursor 为例 32
总结 **36**
风险提示 36

图表目录

图表 1： 代码生成大模型发布时间线 5
图表 2： OpenRouter 大模型应用排行榜 6

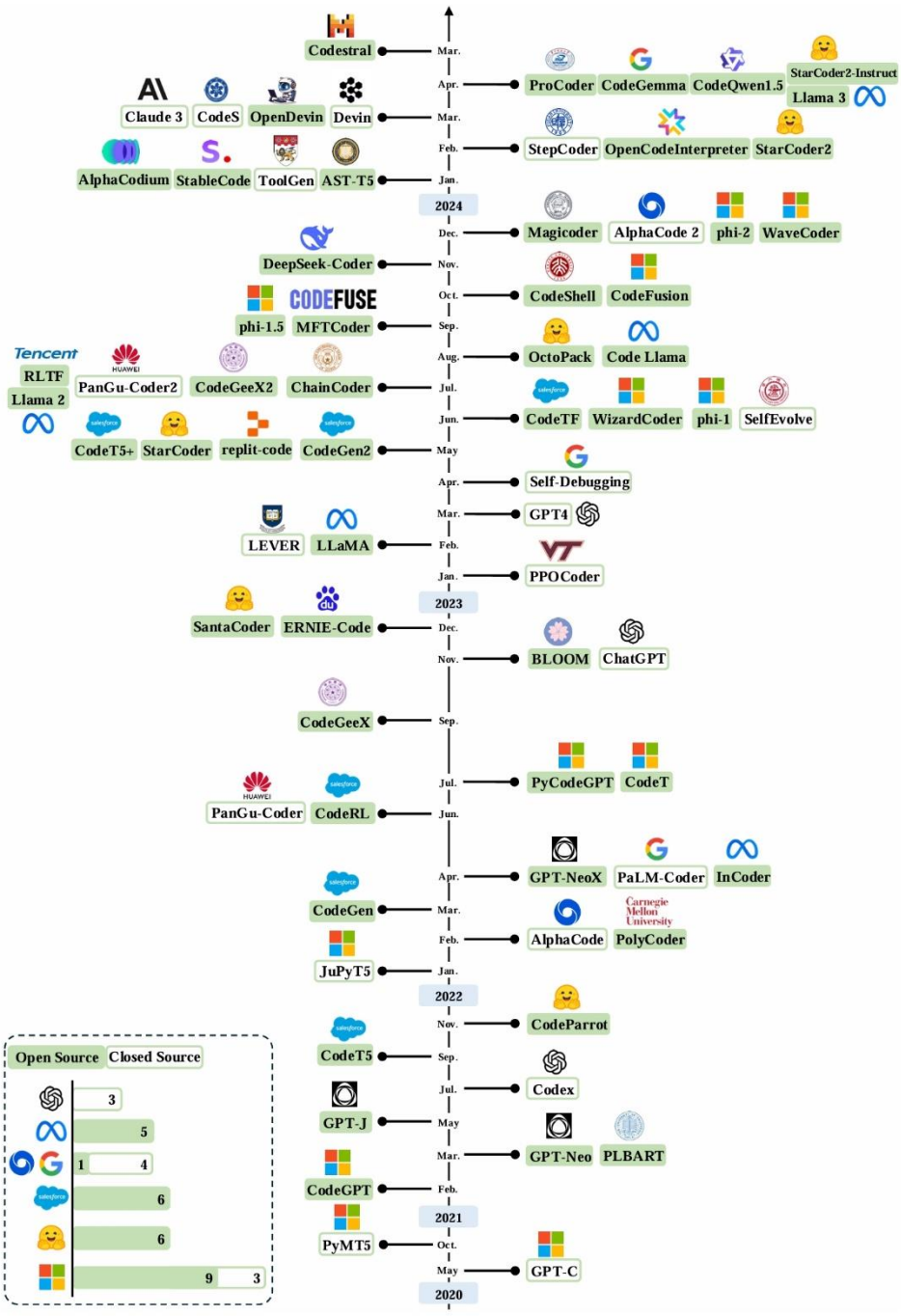
图表 3： DeepSeek-R1 对比其他模型效果 7
图表 4： Cursor 的 Tab 键功能 7
图表 5： 插件类大模型辅助编程工具基础信息汇总 8
图表 6： Github Copilot 插件页面展示 9
图表 7： Github Copilot 不同部署方式对比汇总 9
图表 8： Github Copilot 浏览器部署 9
图表 9： Github Copilot 终端部署 10
图表 10： Github Copilot 根据注释自动生成代码 10
图表 11： Github Copilot 解释代码功能展示 11
图表 12： Github Copilot 生成测试功能展示 11
图表 13： Github Copilot 行内编辑 12
图表 14： Github Copilot q/a 功能展示 12
图表 15： Github Copilot 快捷键汇总（以 windows + VSCode 为例） 13
图表 16： Github Copilot 行内对话快捷指令汇总（以 windows + VSCode 为例） 13
图表 17： Github Copilot Slide 对话快捷指令汇总（以 windows + VSCode 为例） 13
图表 18： CodeGPT 插件页面展示 14
图表 19： CodeGPT 接入本地 Ollama 模型进行代码辅助编写 15
图表 20： CodeGPT 对话框页面展示 15
图表 21： CodeGPT 对话快捷指令汇总（以 windows+VSCode 为例） 16
图表 22： CodeGPT CodeBuilder 功能展示 16
图表 23： CodeGPT Stack Overflow 功能展示 17
图表 24： MarsCode 功能概述 18
图表 25： MarsCode 插件页面展示 18
图表 26： MarsCode 快捷键汇总（支持自定义） 19
图表 27： Codeium 插件页面展示 20
图表 28： Cline 插件页面展示 21
图表 29： Cline 支持的 API 格式类型 21
图表 30： Cline 执行终端命令图示 22
图表 31： Cline 文件修改记录图示 22
图表 32： Cline 增加上下文图示 23
图表 33： Cline 请求使用浏览器图示 23
图表 34： Cline 自动生成代码文件图示 24
图表 35： Cline MCP Server 界面 24
图表 36： 插件类大模型辅助编程工具对比 25
图表 37： Cursor 界面展示 27
图表 38： Cursor Composer 功能展示 28
图表 39： Cursor Composer Agent 模式 28
图表 40： Windsurf 的 AI Flow 范式 29
图表 41： Windsurf 页面展示 30
图表 42： MarsCode IDE 页面展示 31

图表 43： MarsCode 快捷键汇总（以 windows 系统为例） 31
图表 44： Chat 生成项目结构对话提示词 32
图表 45： Cursor Chat 生成项目结构对话结果 33
图表 46： Cursor Composer Agent 编写 SMA 策略过程页面展示 34
图表 47： Cursor Composer Agent 编写 SMA 策略运行过程日志 35
图表 48： Cursor Composer Agent 编写 SMA 权益曲线可视化 35
图表 49： Cursor Composer Agent 编写 SMA 策略回测结果 35

引言

在当今的大模型时代中，大语言模型（LLMs）正以其强大的信息处理能力和广泛的应用前景，成为推动各垂直领域工具革新的关键力量。随着代码生成大模型的不断演进，大模型在提高编程效率、优化代码质量以及创新开发流程中的作用日益凸显。

图表1：代码生成大模型发布时间线



资料来源：Jiang et al.（2024）A Survey on Large Language Models for Code Generation，华泰研究

随着大模型技术的不断发展，其在金融投研领域的应用之广泛，进一步凸显了大模型辅助投研的重要价值。华泰金工在前期已发布多篇关于大模型的应用报告，深入探究大模型在量化研究与日常工作的辅助作用，例如《GPT 因子工厂 2.0：基本面与高频因子挖掘》（20240926）和《大模型本地部署手册》（20241007）等。

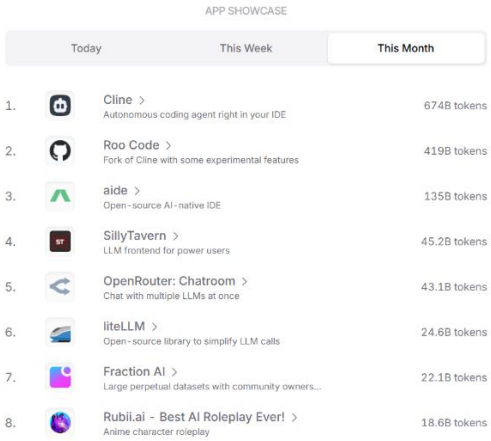
本文尝试从大模型辅助编程视角，为金融投研领域的开发者提供一个全面的指南，详细介绍各类辅助编程工具特点和特色，以及利用此类工具提升编程工作效率和质量的技巧。本文从大模型嵌入开发环境的程度出发，将大模型辅助编程工具分为两类：插件类和 IDE 类，并通过整理 Github Copilot、CodeGPT、MarsCode、Codeium 和 Cline，以及 Cursor、Windsurf 和 MarsCode IDE 等工具的具体功能和部署流程，为读者提供了一个可实操的应用指南。

此外，本文不仅详细介绍了各类大模型辅助编程工具的特点，还探讨了如何通过这些工具实现更高级的编程实践，如量化策略编写。通过 Cursor Composer 等工具的应用实例，本文展示了大模型辅助编程工具在实际量化策略开发中的灵活性和潜力。最后，综合考虑各类辅助编程应用性能及实际体验，本文推荐以下应用，仅供读者参考。

推荐应用一：VSCode + Cline + DeepSeek

作为 OpenRouter（大模型应用排行榜）排名第一的应用，Cline 在多方面占据优势。首先，Cline 支持用户自行嵌入模型 API，便于用户使用开源、低成本或有特定维度优势的模型，在此基础上，Cline 拥有高度自动化的辅助编程功能，能够自行创建、编辑文件，以及调用外部工具，从而给予用户极佳的使用体验。

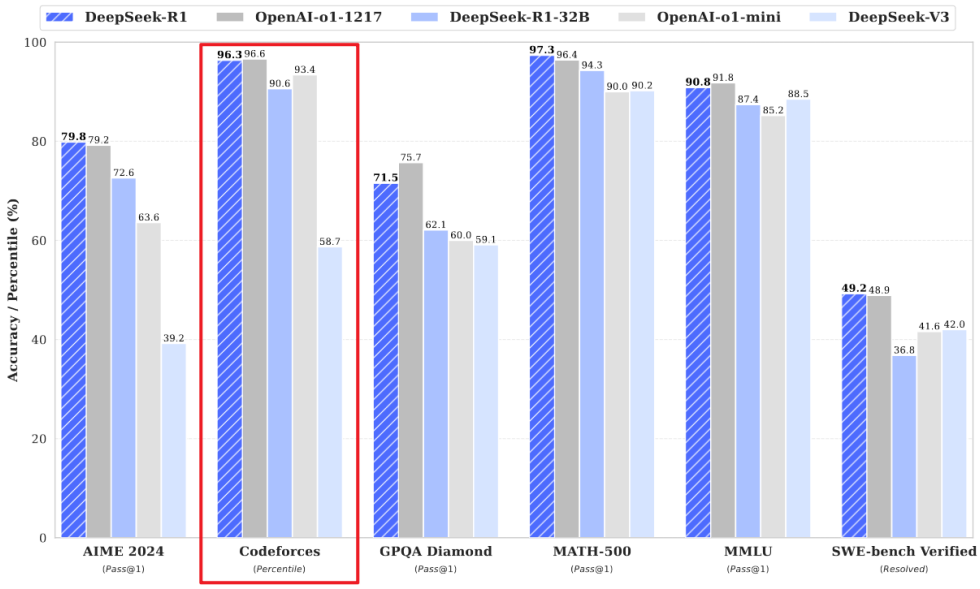
图表2：OpenRouter 大模型应用排行榜



注：排行榜数据截至 2025 年 2 月 23 日
资料来源：openrouter.ai，华泰研究

自 DeepSeek-R1 发布以来，DeepSeek 相关模型及应用持续火热。以 Codeforces 数据集（编程能力测试数据集）为对比标准，DeepSeek-R1 与 OpenAI-o1-1217 几近持平，展现出其优异编程能力。然而，DeepSeek-R1 的使用成本是 OpenAI-o1-1217 的十分之一乃至更低，且 DeepSeek-R1 开源可做本地化部署进一步降低成本。DeepSeek-R1 是大模型辅助编程“物美价廉”的模型选择，而在当前可实现自动化编程的工具中，例如 Cursor、Windsurf、Cline 等，Cline 是为数不多较好支持 DeepSeek 系列模型的应用。

图表3：DeepSeek-R1 对比其他模型效果



资料来源：DeepSeek-R1 技术报告，华泰研究

推荐应用二：Cursor

相比于 Cline 这类插件类辅助编程工具，Cursor 作为完整的 IDE 更好地扩充了大模型辅助编程的权限。除却自动化编程这一功能，Cursor 同时拥有代码补全、行内编辑和对话功能，形成了 Cline + Github Copilot 的功能叠加，这使得 Cursor 在仅需细微代码补充的场景中更具优势。从这一角度上看，Cursor 是辅助编程工具的集大成者，基于其优异能力，Cursor 获得 Product Hunt 2024 年度最佳产品奖项。

与此同时，相比其他产品，Cursor 拥有更优秀的智能化设计。以 Tab 键功能为例，Cursor 的多行编辑功能，可以一次性提供多个编辑建议，帮助用户节省时间；智能重写功能，用户随意或不小心输入之后，Cursor 会自动修正错误；光标预测功能，Cursor 能预测用户的下一个光标位置，以便可以无缝导航代码。

图表4：Cursor 的 Tab 键功能

资料来源：Cursor 官网，华泰研究

插件类

插件类大模型辅助编程工具本质上是一种深度集成到开发环境（IDE）中的智能助手系统，它通过实时分析开发者的编程上下文、项目结构和编码意图，提供智能化的编程辅助服务。这类工具最显著的特征是其“插件化”设计，使其能够无缝融入开发者熟悉的 IDE 环境中，成为编程工作流的自然延伸。这类工具通常以 IDE 插件的形式存在，可以无缝嵌入到开发者日常使用的编辑器中，如 VS Code、IntelliJ IDEA 等。它们能够理解当前代码文件的结构、上下文关系以及项目依赖，在开发者编写代码时提供实时的智能建议。

以 GitHub Copilot、CodeGPT、MarsCode 和 Codeium 为代表的插件类辅助编程工具都具备代码自动补全、代码生成、注释生成、代码解释等核心功能，其中 GitHub Copilot 作为较早且成熟的产品，在代码生成的准确性和连贯性方面表现出色；CodeGPT 则更专注于代码解释和文档生成；MarsCode 在中文编程场景下有其独特优势；而 Codeium 则提供了免费且功能完整的替代方案。Cline 作为插件类编程工具的例外，具有全自动化的代码辅助生成功能，将大模型能力深度集成至开发流程，可以运行终端命令、创建修改文件、使用浏览器等。

插件类辅助编程工具适用于日常编程过程中的代码编写、调试和重构场景，能够显著提升开发效率，尤其适合处理重复性的编程任务、API 调用示例编写、单元测试生成等场景。工具背后的大模型通过学习和理解大量开源代码，能够为开发者提供符合编程规范和最佳实践的代码建议，有效降低编程过程中的认知负荷。

图表5：插件类大模型辅助编程工具基础信息汇总

	插件官网	供应商	模型支持	开发平台支持
Github Copilot	https://github.com/features/copilot	Github（微软）OpenAI 联合开发	GPT、Claude	VSCode，JetBrains，Vim/Neovim
CodeGPT	https://codegpt.co/	Judini	GPT、Claude、DeepSeek 等主流 Api 接口以及本地 Ollama 本地部署模型	VSCode，JetBrains
MarsCode	https://www.marscode.cn	字节跳动	豆包代码模型	VSCode，JetBrains
Codeium	https://codeium.com/	Exafunction	GPT、Claude、Llama	VSCode，JetBrains 等 17 多种编辑器
Cline	https://cline.bot/	开源	GPT、DeepSeek 等主流 API 接口及本地部署模型	VSCode

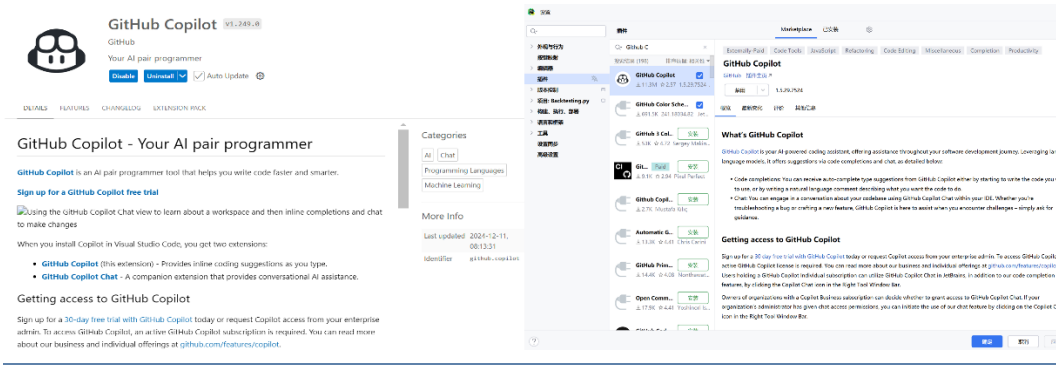
资料来源：各插件官网，华泰研究

Github Copilot：老牌代码智能补全插件

GitHub Copilot 于 2021 年 6 月推出，是由 GitHub 和 OpenAI 合作开发的一款大模型编程助手，旨在通过提供代码建议和自动完成功能来提高开发者的编程效率。Copilot 基于 OpenAI 的 Codex 模型，该模型在大量公开的代码库上训练得来，能够理解多种编程语言，并生成符合上下文的代码片段，为开发人员提供实时的代码提示和生成功能。Copilot 不仅能够补全代码行，还能生成整个函数、循环、条件语句等复杂结构，同时可以根据注释生成代码。

Github Copilot 的设计理念是让开发者专注于代码逻辑以及整体框架设计，而不是花费大量时间在编写重复性代码上。它支持多种集成开发环境（IDE），如 VSCode、JetBrains IDEs 等，使得开发者可以在熟悉的开发环境中无缝使用 Copilot 的功能，减少手动输入代码的时间，提高代码的质量和效率。同时它支持多种编程语言，如 Python、JavaScript、TypeScript、Ruby 等。Copilot 的关键技术包括自然语言处理、代码生成优化、上下文感知等，这些技术使得 Copilot 能够准确理解开发者的意图，并提供高质量的代码建议。

图表6：Github Copilot 插件页面展示



注：左图为 VSCode 平台，右图为 PyCharm 平台
资料来源：Github Copilot 平台，华泰研究

部署

目前，Github Copilot 支持三类部署方案：**IDE 类、浏览器以及终端**。首先，对于 IDE 类，如 Jetbrain 或 VSCode 开发环境中，可以在扩展中寻找 Github Copilot 插件/扩展进行安装，登录 Github 账号后，即可试用 Copilot 功能。以 VSCode 为例，在编辑器上点击 Extensions 图标，会看到 Marketplace 选项， 然后搜索 copilot 结果，安装对应两个插件，并进行登录即可。在最新 VSCode(1.96.1)上，可以按 Ctrl+Alt+I 或从标题栏的 Copilot 菜单中选择“Use AI Features with Copilot for Free...”，并在 Copilot Chat 窗口中，点击 “Sign in to Use Copilot for Free”按钮，登录到 GitHub 账户并注册 Copilot 免费计划，即可免费使用 Github Copilot。

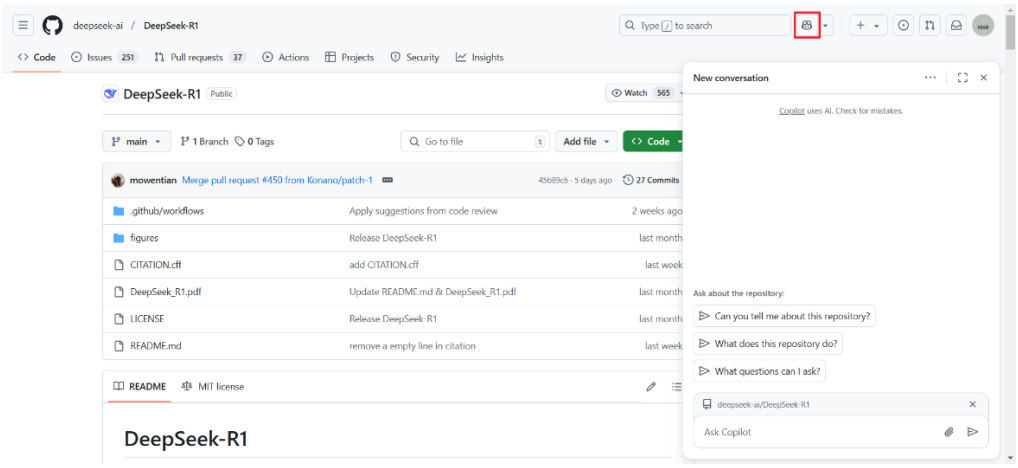
图表7：Github Copilot 不同部署方式对比汇总

部署方式	主要作用	支持的功能
浏览器	理解仓库的目的、检查文件并深入了解特定的代码行	直接对话，解释代码
终端	获取命令说明、获取代码建议	直接对话
VSCode	辅助代码编写（目前可以加入计划开发者免费使用 Github Copilot）	直接对话，解释、测试代码，行内编辑器、自动补全
PyCharm	辅助代码编写	直接对话，解释、测试代码，行内编辑器、自动补全

资料来源：Github Copilot 文档，华泰研究

其次，浏览器上同样支持与 Copilot 对话。具体而言，Copilot 功能嵌入 Github 网站中，只需要导航到 repository 并打开一个文件，并单击文件视图右上角的 Copilot 图标，即可与 Copilot 进行对话。

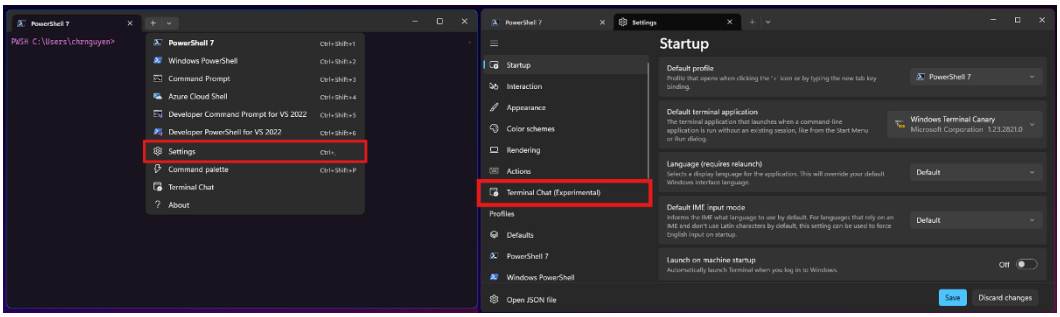
图表8：Github Copilot 浏览器部署



资料来源：Github Copilot 平台，华泰研究

此外，Github Copilot 还可以部署在 Windows 终端上，用户需要先安装 Windows Terminal Canary，在 Settings 中转到 Terminal Chat 设置，在 Service Providers 中选择 Github Copilot 并登录账号，即可在 Terminal Chat 中与 Copilot 进行对话。在模型方面，Github Copilot 支持 GPT-4o，o1-preview，o1-mini 与 Claude-sonnet-3.5，用户可以在对话编辑的时候自由选择模型。

图表9：Github Copilot 终端部署



资料来源：Github Copilot 平台，华泰研究

基础功能：代码补全、代码解释、生成测试等

代码自动补全（Autocompletion）

在项目中打开一个文件，开始编写代码或在代码旁边添加注释（建议先定义好新文件的内容格式或者是文件格式，补全会更准确）。GitHub Copilot 会根据代码上下文提供补全建议。

图表10：Github Copilot 根据注释自动生成代码

```
# 封装一个函数实现将数组内容顺序随机打乱重排
import random
def shuffle(arr):
    random.shuffle(arr)
    return arr
```

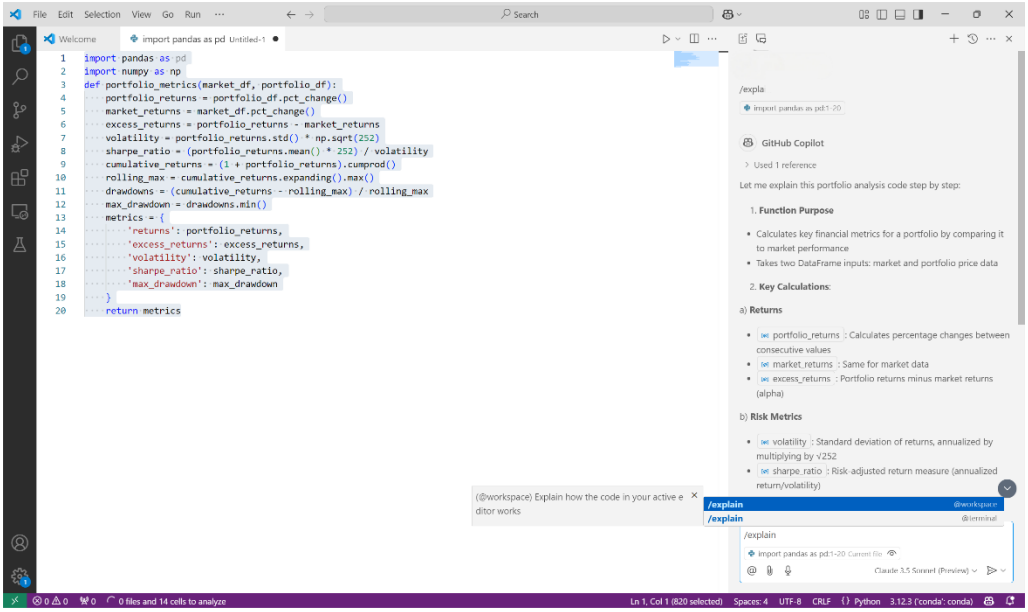
资料来源：Github Copilot 平台，华泰研究

可使用 Tab 键直接接受 GitHub Copilot 提供的代码建议。如需查看更多建议，可使用 Alt+]（Windows/Linux 系统）或 Option+]（macOS 系统）来切换到下一个建议，或者使用 Alt+[（Windows/Linux 系统）或 Option+[（macOS 系统）来切换到上一个建议。除 Tab 键，用户还可以使用其他组合快捷键来快速接受建议，例如，Ctrl+Enter（Windows/Linux 系统）或 Cmd+Enter（macOS 系统）将接受当前的代码建议并换行。

代码解释（Explain）

Github Copilot 提供代码注释功能，可提高代码可读性，方便后期维护与修改，首先选中要解释的代码行，在聊天框中输入/explain，或者右键点击代码行选择 Copilot > Explain this，如果需要中文解释，可以在聊天框中输入/explain 后跟中文描述。或者也可以在函数前加上注释符//，Copilot 会自动分析代码功能，并添加相应注释。采用这个功能，开发者可以快速理解代码的目的和工作方式，即使是在阅读别人的代码时也能提高效率。

图表11：Github Copilot 解释代码功能展示

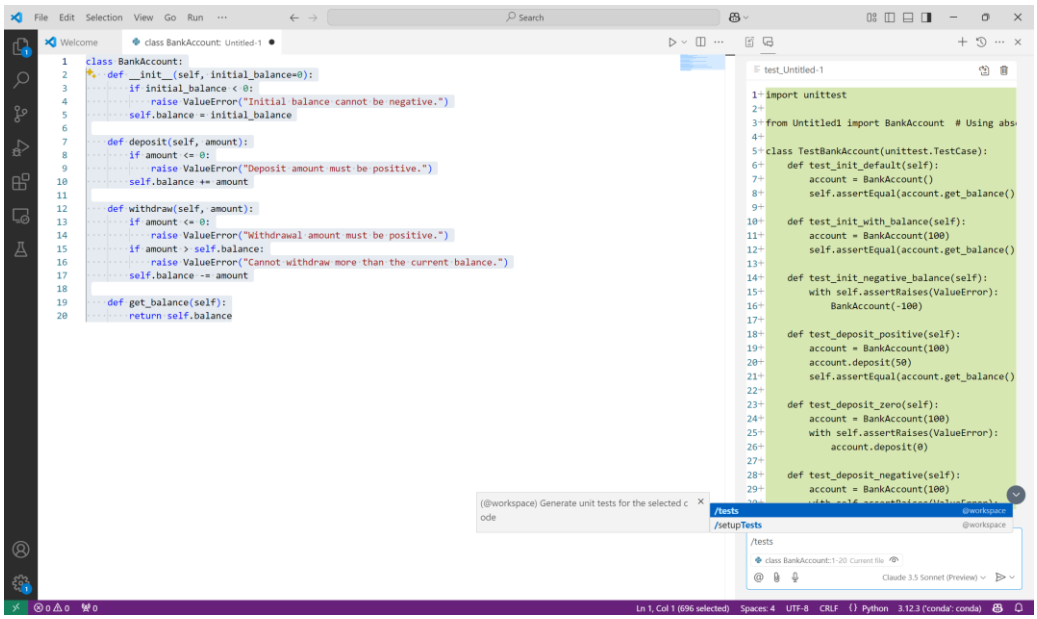


资料来源：Github Copilot，VSCode 平台，华泰研究

生成测试（Tests）

用户可以通过在 IDE 的当前选项卡上打开文件，并使用/tests 斜杠命令来生成一套完整的单元测试。此外，用户也可以使用 Chat 功能生成针对特定类的单元测试，例如可以创建一个 BankAccount 类，并请求 Copilot 生成涵盖多种场景的测试，包括边缘案例、异常处理和数据验证。

图表12：Github Copilot 生成测试功能展示

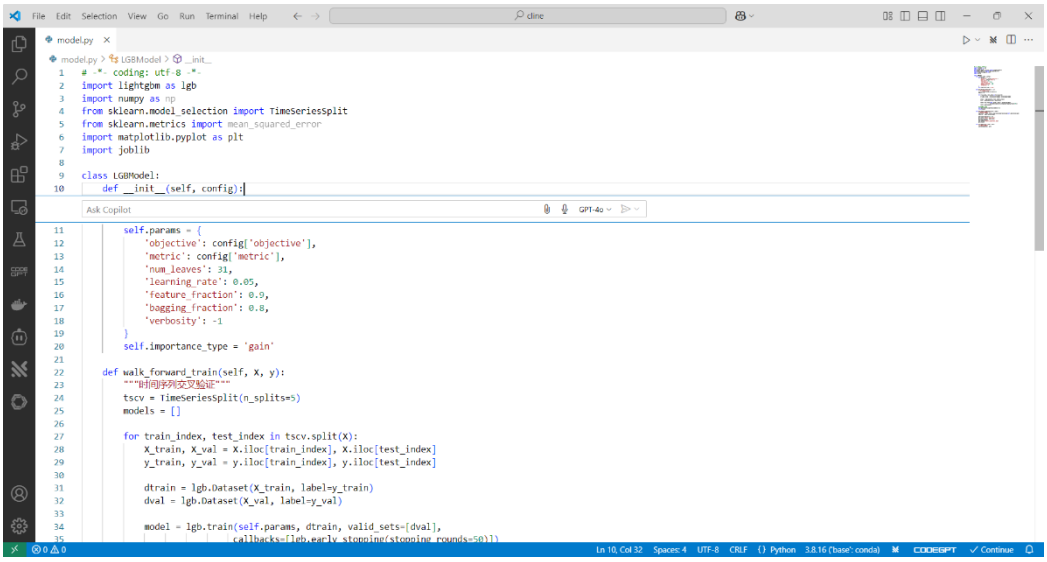


资料来源：Github Copilot，VSCode 平台，华泰研究

行内编辑（Inline Edit）& q/a 对话

在编辑器文件中，按 Cmd+I（在 macOS 上）或 Ctrl+I（在 Windows/Linux 上）进行行内操作（即行内编辑，Inline Edit），可以在当前光标所在的行展示一个输入框，用于提问或输入命令。

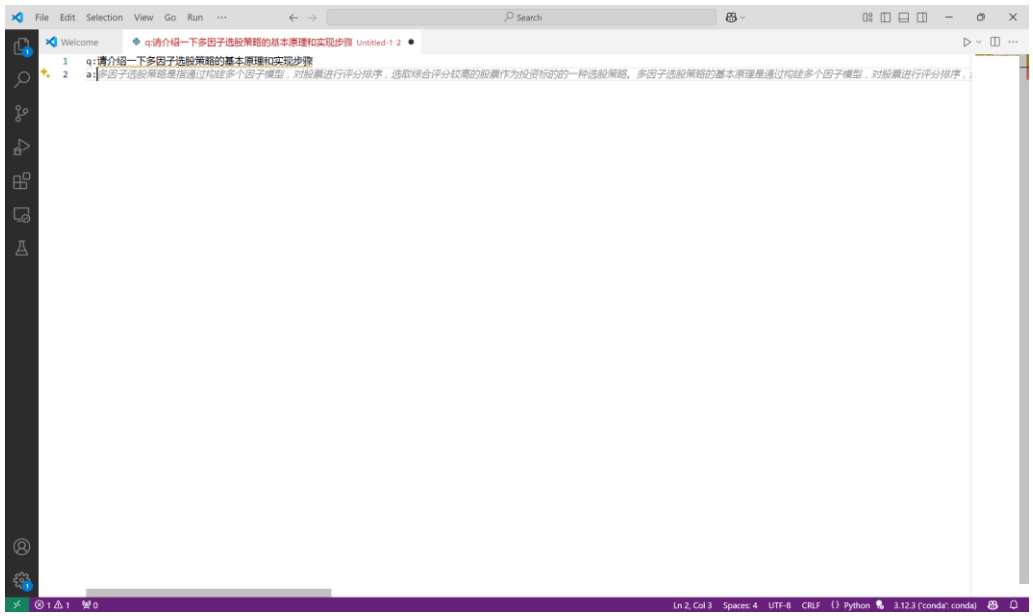
图表13：Github Copilot 行内编辑



资料来源：Github Copilot，VSCode 平台，华泰研究

在代码文件中输入注释符号后，输入 q:后跟一个问题，然后在下一行输入 a:，GitHub Copilot 会提供简短的回答。

图表14：Github Copilot q/a 功能展示



资料来源：Github Copilot，VSCode 平台，华泰研究

核心功能：联想编程

GitHub Copilot 的联想编程功能是一项基于大模型的代码辅助技术，它通过使用 OpenAI 的 Codex 模型，分析开发者输入的代码、注释和上下文信息来生成代码建议。此外，Copilot 还具备学习能力，能够根据用户的反馈不断学习，提供更加个性化的代码提示，帮助提升开发技能。GitHub Copilot 的个性化学习能力主要依托于其底层的 OpenAI Codex 大语言模型，通过本地和云端两个层面来实现。

在本地层面，它会实时分析用户的编码上下文、项目结构和编码风格，并在当前编辑会话中记住用户的选择和修改，具体而言包括用户的缩进、命名以及注释风格，偏好的架构设计模型、项目结构，框架、库的使用习惯以及常用的编程模式和解决方案。在云端层面，系统会收集用户对代码建议的接受（Accept）或拒绝（Discard）等反馈数据，分析用户的编码习惯和偏好，并可能为特定项目建立专门的建议模型。这种学习机制在确保用户隐私和数据安全的前提下，通过持续的优化和调整，使 Copilot 能够随着用户使用时间的增加，提供越来越符合个人或开发团队需求的代码建议。

图表15：Github Copilot 快捷键汇总（以 windows + VSCode 为例）

操作	快捷键
接受行内建议	Tab
拒绝行内建议	Esc
显示下一个行内建议	Alt+]
接受上一个行内建议	Alt+[
触发行内建议	Alt+\
打开 Github Copilot	Ctrl+Enter
打开行内操作	Ctrl+I

资料来源：Github Copilot 文档，华泰研究

图表16：Github Copilot 行内对话快捷指令汇总（以 windows + VSCode 为例）

操作	快捷指令
修复选中代码	/fix
添加文档注释	/doc
对选中代码进行解释	/explain
为选中代码生成单元测试	/tests

资料来源：Github Copilot 文档，华泰研究

图表17：Github Copilot Slide 对话快捷指令汇总（以 windows + VSCode 为例）

操作	快捷指令
快速定位代码	/search
设置断点并启动调试对话	/startDebugging
修复选中代码	/fix
修复测试失败问题	/fix TestFailure
开始新项目并建议目录结构	/new
创建新的 Jupyter Notebook	/newNotebook
设置测试环境	/setupTests
对选中代码进行解释	/explain
为选中代码生成单元测试	/tests
获取帮助信息	/help
清楚当前聊天会话或代码建议	/clear
询问有关命令行特定问题	@terminal
询问有关 VSCode 特定问题	@VSCode
处理与工作区相关任务	@workspace

资料来源：Github Copilot 文档，华泰研究

CodeGPT：多模型及开源模型支持

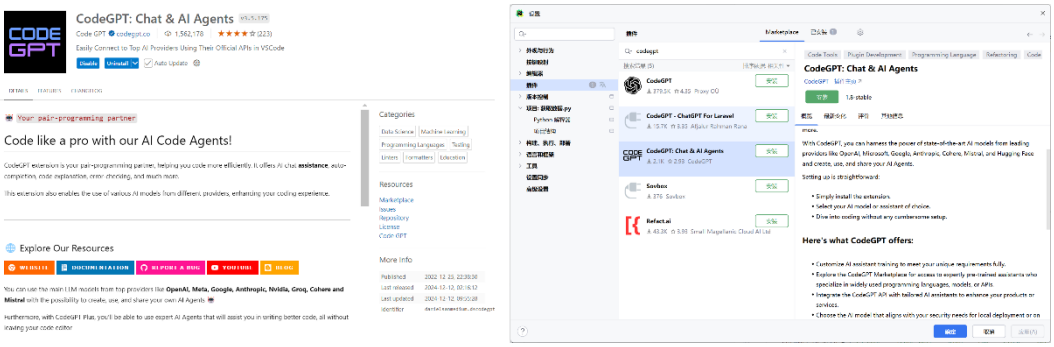
CodeGPT 是由 Judini 打造的一款专为开发者设计的生成式 AI 产品，它通过自然语言处理技术，能够自动生成程序源代码，并集成了多种功能以提高开发者的编程效率和代码质量。这款工具不仅能够回答开发者的技术问题，提供智能编码聊天功能，还能自动识别编程语言并生成代码解释，帮助开发者理解复杂的代码段。

在已有代码的基础上，CodeGPT 能够继续编写代码，实现更多功能或完成更复杂的任务。它还能一键生成方法注释及行间注释，提高代码的可读性和维护性。此外，CodeGPT 能够对代码片段进行静态分析，识别并重构不完整或冗余的代码，优化代码的可读性、性能和可维护性。它还能实时诊断代码，纠正潜在的语法错误和逻辑错误，提高代码的稳定性和质量。CodeGPT 还能自动生成测试代码，减轻开发者在测试环节的负担。

CodeGPT 支持多种集成开发环境，如 VSCode 和 JetBrains 系列 IDE，CodeGPT 能够在开发者熟悉的开发环境中无缝工作。其关键技术包括自然语言处理、代码生成优化和上下文感知等，使得 CodeGPT 能够准确理解开发者的意图，并提供高质量的代码建议。

此外 CodeGPT 也提供 API 接口，将大模型集成到开发工作流程中，使得开发者能够利用 CodeGPT 的能力来增强他们的应用程序和服务。通过这些 API 接口，开发者可以实现代码自动完成、生成、解释和重构，以及实时错误检查和调试。

图表18：CodeGPT 插件页面展示



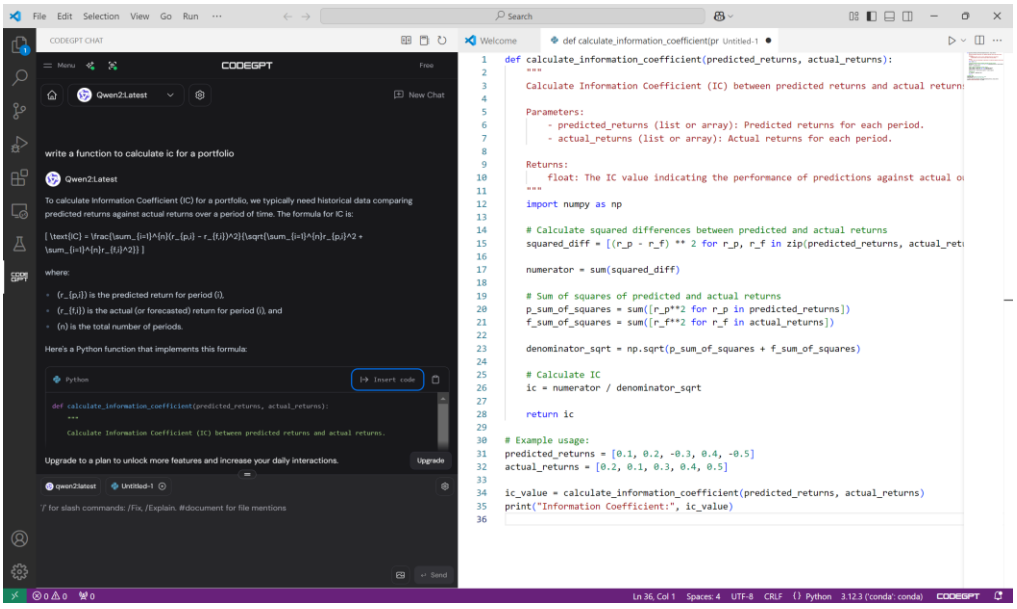
注：左图为 VSCode 平台，右图为 PyCharm 平台
资料来源：CodeGPT 平台，华泰研究

部署

目前 CodeGPT 支持在 VSCode，Cursor 和 Jetbrains 三个 IDE 开发环境上部署，与 Github Copilot 相同，可以在 Jetbrains—设置—插件，VSCode—扩展，Cursor—扩展中搜索 CodeGPT 安装并登录即可运行 CodeGPT。CodeGPT 在 VSCode 与 PyCharm 上的功能支持有所区别，在 VSCode 上不仅支持更多编程语言、流式响应，同时拥有更丰富的市场插件集成，即不同种类的预设代理，下面功能介绍中 CodeBuilder 与 Stack Overflow 均为预设的代理功能，相比之下 PyCharm 中的 CodeGPT 更专注于 Python 语言。

在安装部署后，用户需要设置对话模型及模型 API 才能开始对话，与 GitHub Copilot 不同，CodeGPT 支持更多模型自由选择，除去 GPT、Claude、DeepSeek 等主流模型，CodeGPT 还支持接入 Huggingface、Groq 等 API 接口，除此之外，还支持连接到本地的 Ollama 与 LM Studio 等大模型部署框架，一定程度上为不同用户提供更为自由的模型选择，降低实现成本，大模型本地部署方案可参考前期报告《大模型本地部署手册》(20241007)。使用本地大模型可实现响应迅速、低成本且无信息泄露风险的大模型辅助编程方案。

图表19： CodeGPT 接入本地 Ollama 模型进行代码辅助编写



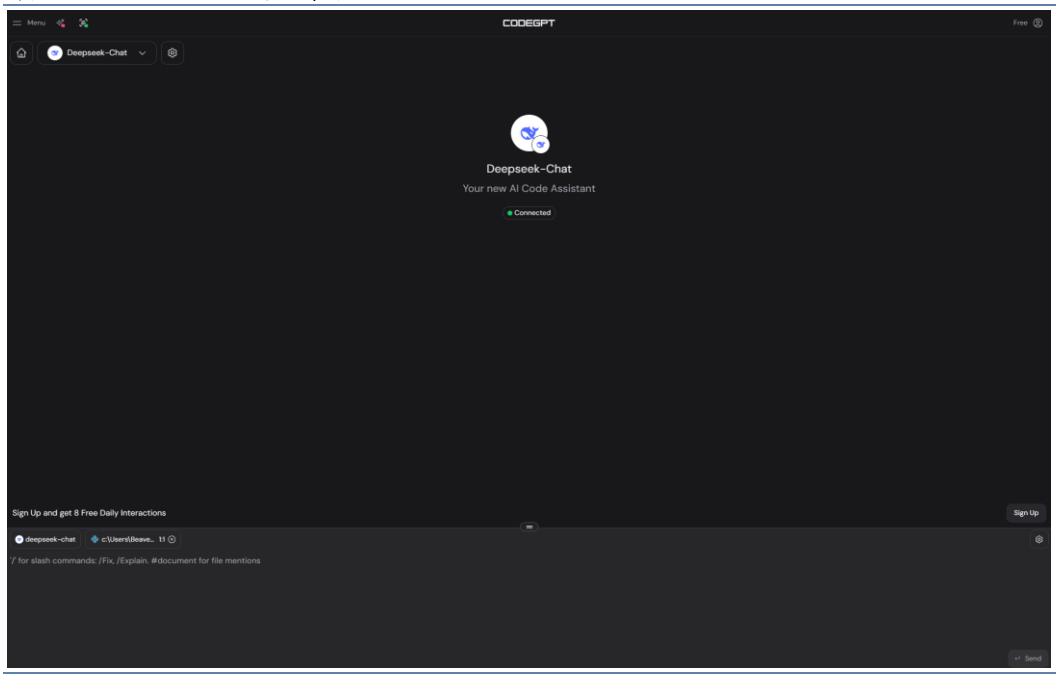
资料来源：CodeGPT，Qwen， VSCode 平台，华泰研究

基础功能：对话、代码补全、代码生成、即时检索等

对话（Chat）

点击左侧 GodeGPT 图表即可打开相应对话框，在对话框中可以自由选择对话模型，选中代码，点击输入框上方的 Import Selection 即可针对特定代码进行对话，与 Github Copilot 类似，CodeGPT 也提供部分快捷指令，方便用户在对话框中迅速实现编程常用功能。同时快捷键功能也同样可以通过选中代码片段，右键单击，选择对应功能实现。

图表20： CodeGPT 对话框页面展示



资料来源：CodeGPT 平台，DeepSeek，华泰研究

图表21：CodeGPT 对话快捷指令汇总（以 windows+VSCode 为例）

操作	快捷指令	功能描述
修复选中代码	/Fix	
解释选中代码	/Explain	帮助深入了解代码库
重构选中代码	/Refactor	优化代码库，提高可读性与效率
为选中代码创建文档	/Document	创建详细且信息丰富的文档
测试选中代码	/Unit Test	快速为代码生成单元测试
调试选中代码	/Debug	
查找问题	右键+CodeGPT: Find Problems	CodeGPT 主动识别和解决代码库中的问题

资料来源：CodeGPT 官方文档，华泰研究

代码自动补全（Autocompletion）

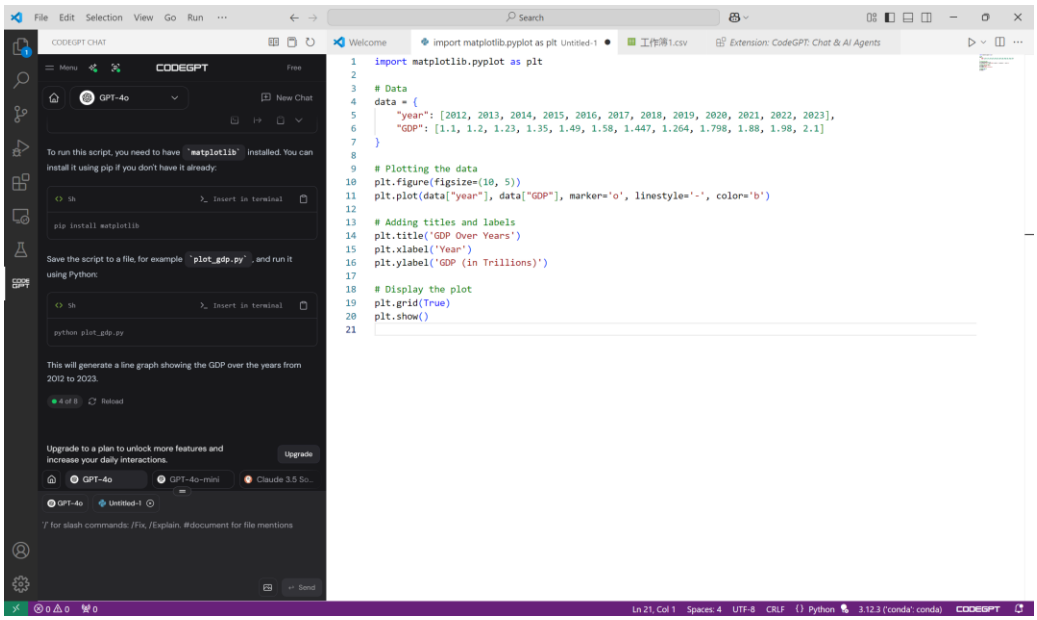
自动补全是插件类辅助编程工具的重要功能之一，对于 CodeGPT 而言，相比于 Chat 功能，Tab 自动补全可供选择的模型相对更少，仅包含 CodeGPT 订阅模型、Mistral 的 codestral 以及 Ollama 本地部署的部分 code 模型，可通过 Menu—Autocomplete—Status（Enable）来开启功能。值得注意的是，如果使用 Ollama 部署的模型，需要本地先运行 Ollama 服务器并在 CodeGPT 中正确配置。除此之外，CodeGPT 也支持针对仓库提交 Commit 信息的自动补全，当建议出现后，同样也只需要按 Tab 来接受。

代码生成器（CodeBuilder）

CodeBuilder 是 CodeGPT 插件中的一个功能强大的工具，它主要的作用是帮助开发者快速生成项目结构，包括文件夹和文件，可以理解为与下文 Cursor 的 Composer 功能的简略版。这个工具通过理解开发者的需求，从而自动创建出完整的项目结构，极大地提高了开发效率。CodeBuilder 支持多种流行的框架，如 Angular、Next.js 和 React，并且能够根据这些框架的最佳实践提供 AI 驱动的建议。使用 CodeBuilder 的过程也非常简单直观。用户首先需要登录其 CodeGPT 账户，然后在文本区域使用/CodeBuilder 命令，添加项目结构的上下文或规格，点击发送后，CodeBuilder 会提出一个项目结构的建议，最后点击'Create'即可生成项目结构。

需要注意的是，使用 CodeBuilder 功能需要将 CodeGPT 插件升级为 v3.5.13（pre-release）版本，且仅适用于 GPT-4o 模型。例如，可以选中数据的 csv 文件，并输入“/CodeBuilder Creat a Python file to graph this data”，CodeBuilder 会自动理解用户需求，创建文件，点击 create 即可接受。

图表22：CodeGPT CodeBuilder 功能展示



资料来源：CodeGPT 平台，华泰研究

行内编辑（Inline Edit）

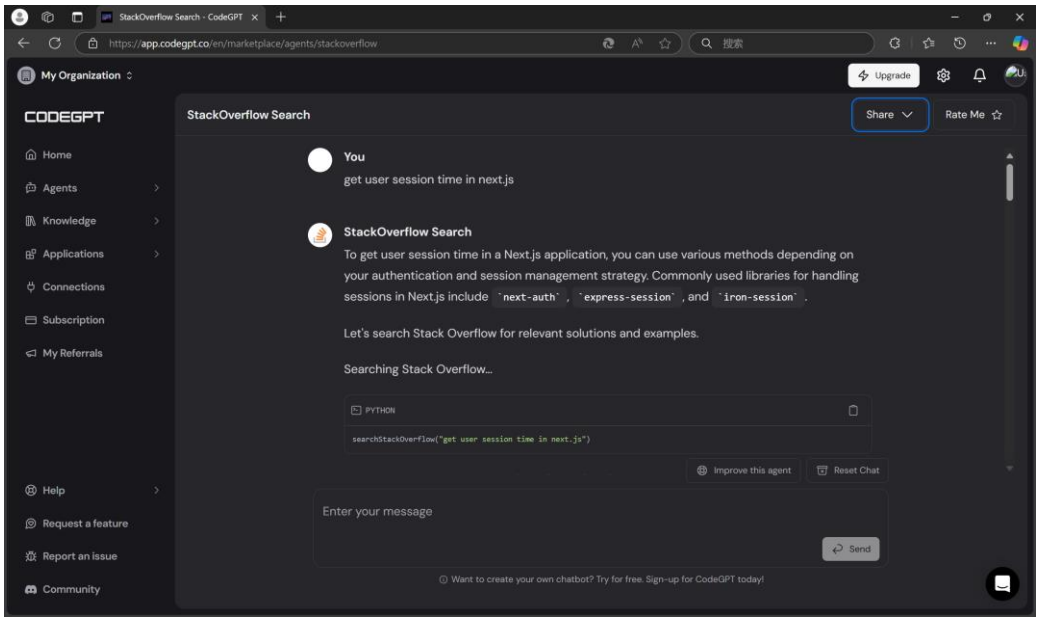
选择代码，按或右键单击所选代码，然后选择 CodeGPT：选择行内代码编辑，或使用快捷键 Cmd+shit+k （Ctrl+shift+k on Windows）呼出行内编辑器；对话后，原有内容和新内容将以绿色和红色突出显示，按接受建议（Tab）即可进行修改。

即时搜索（Stack Overflow）

CodeGPT 插件中的 Stack Overflow 功能为用户提供了一个直接在 VSCode 环境中访问 Stack Overflow 问答社区的便捷方式。这个功能使得用户在编码时遇到问题可以直接在 IDE 中搜索解决方案，而不必离开编辑器去浏览器中查找。CodeGPT 能够理解用户提出的问题，并从 Stack Overflow 中检索出最相关的回答，同时生成简洁的摘要，帮助用户迅速把握问题的核心和解决方案，节省用户在不同工具间切换的时间，同时提高解决问题的效率。

具体操作方面，用户只需要在 Chat 中输入/Stack Overflow+问题即可实现实时搜索，并解决问题。需要注意，与 CodeBuilder，CodeInterpreter（对话代码编辑器）两个功能类似，都需要选择 CodeGPT Plus—GPT-4o 模型，并在 Chat 页面安装对应扩展才能使用该功能。

图表23：CodeGPT Stack Overflow 功能展示



资料来源：CodeGPT 平台，华泰研究

API 接口

CodeGPT 的 API 提供了丰富的功能，主要围绕 AI 助手的管理和交互展开，相当于普通大模型 API 增加自动 RAG 和代码库管理功能。在 CodeGPT 系统中，AI 代理不仅仅是模型的简单实例，而是一个复杂的工具，利用检索增强生成（RAG）来提供结合上下文的详细准确的响应。用户可以创建代理，上传管理文件代码，当与代理交互时，它会自动根据输入和上传的文档执行语义搜索，检索最相关的文档，进行检索增强生成。

MarsCode：轻量化编程助手插件

豆包 MarsCode 编程助手是豆包旗下的 AI 编程助手，提供以智能代码补全为代表的 AI 功能，支持主流编程语言及 IDE，能在开发过程中提供单行或整个函数的编写建议，同时支持在用户开发过程中提供代码解释、代码审查、问题修复等辅助功能，提升开发效率与质量。

图表24：MarsCode 功能概述

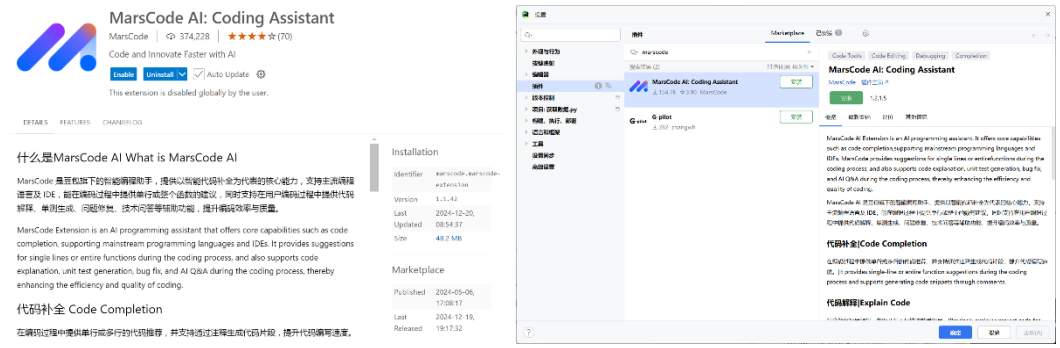
功能	说明
代码补全	阅读并理解当前代码，然后提供后续代码片段，也支持通过注释生成代码片段。
代码补全 Pro	基于上一次的编辑内容及代码情况，预测下一个改动点并提供推荐代码。
代码生成	理解自然语言并生成所需代码。
代码编辑	编辑指定代码，包括重构、优化、修改部分逻辑等
代码解释	精准解释项目代码，快速上手开发。
代码注释生成	生成函数级注释或更详细的行间注释。
单元测试生成	单元测试生成
智能修复	发现代码中的问题并修复。
智能问答	针对研发领域定向优化问答质量，提供更精准的问答结果。

资料来源：MarsCode 官方文档，华泰研究

部署

目前 MarsCode 支持部署在 VSCode 和 JetBrains IDE，部署方式与 Github Copilot 以及 CodeGPT 一致。

图表25：MarsCode 插件页面展示



注：左图为 VSCode 平台，右图为 PyCharm 平台
资料来源：MarsCode 平台，华泰研究

基础功能：代码补全、代码预测推荐、智能错误修正等

豆包 MarsCode 提供了一系列的 AI 功能，旨在提升开发效率和代码质量。自动代码补全功能允许开发者在编写代码时，通过敲击回车键换行来唤起编程助手，该助手会阅读并理解当前代码上下文，然后自动补全后续代码。此外，如果开发者在代码中添加注释并敲击回车键换行，MarsCode 会根据注释内容生成相应的代码，可以通过 Tab 键接受全部补全，或 Ctrl+→ 逐步接受。

MarsCode 还支持代码预测与编辑，能够基于上一次的编辑内容及代码情况，预测下一个改动点并给出代码推荐。开发者可以通过快捷键 Ctrl+Shift+Enter 主动触发推荐，并使用 Tab 键采纳推荐内容。采纳推荐后，MarsCode 会预测下一个改动点，并继续提供推荐。代码生成功能允许开发者用自然语言描述需求，在 MarsCode 的输入框中点击发送或敲击回车键，MarsCode 会根据描述生成相应的代码片段。

MarsCode 还提供了代码编辑与优化功能，开发者选中代码片段后，可以用自然语言描述代码编辑需求，如“优化选中的代码片段”，然后点击发送或敲击回车键，MarsCode 会根据描述进行代码的重构、优化或逻辑修改，提供代码改进建议，帮助提升代码质量，优化性能。

代码解释功能使开发者可以通过自然语言描述、使用/explain 指令、右键菜单选择 MarsCode>Explain Code 或点击界面上的 Explain 按钮，向 MarsCode 发送代码解释指令。这一功能帮助开发者理解复杂的代码逻辑，促进团队间的代码理解和协作。

单元测试生成功能允许开发者通过自然语言描述、使用/test 指令、右键菜单选择 MarsCode>Generate Test 或点击界面上的 Test 按钮，向 MarsCode 发送单元测试生成指令。MarsCode 会自动为代码生成单元测试，提高测试覆盖率，确保代码质量。

代码注释生成功能使开发者可以通过自然语言描述、使用/doc 指令或点击界面上的 Doc 按钮，向 MarsCode 发送代码注释生成指令。MarsCode 会快速生成代码注释，提高代码的可读性和维护性。

智能错误修复功能在代码中存在错误时，会在代码文件名称旁提示错误数量。开发者选中有问题的代码片段后，输入自然语言描述或使用/fix 指令，MarsCode 会提供智能修复方案，快速定位并修复代码中的错误，减少调试时间。

图表26：MarsCode 快捷键汇总（支持自定义）

行动	快捷键（VSCode）	快捷键（JetBrains）
采用行内自动补全的代码	Tab	Tab
逐字采用行内自动补全的代码	macOS：Command + →	macOS：Command + →
	Windows：Ctrl + →	Windows：Ctrl + →
舍弃代码建议	Esc	Esc
打开侧边 AI 对话框	macOS：Command + U	Ctrl + U
	Windows：Ctrl + U	
展示下一行自动补全的代码	macOS：opt +]	macOS：opt +]
	Windows：alt +]	Windows：alt +]
展示前一行自动补全的代码	macOS：opt + [	macOS：opt + [
	Windows：alt + [	Windows：alt + [

资料来源：MarsCode 官方文档，华泰研究

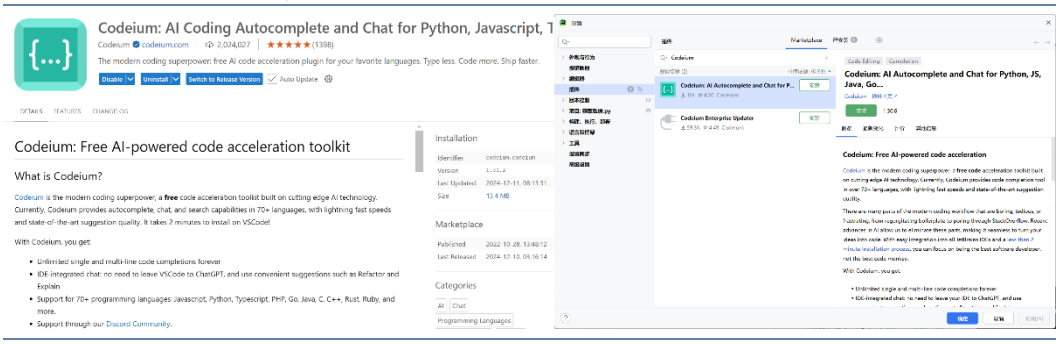
Codeium：多编辑器支持编程助手插件

Codeium 是一家位于美国硅谷的人工智能公司，致力于为开发者提供更智能、高效的编程体验。而 Codeium 是 Codeium 旗下一款辅助编程插件，提供单行和多行代码生成、内置聊天和搜索等核心功能。Codeium 支持超过 70 种编程语言，并与 17+主流 IDEs 兼容，包括 VSCode、JetBrains IDEs、Visual Studio 和 Eclipse 等。这款工具基于大模型技术自动提供补全建议，减少编写样板代码到单元测试的时间和劳动量，同时通过自然语言查询迅速找到所需代码，告别传统复杂的正则表达式搜索方式。同时 Codeium 还具备 AI 代码助手功能，通过 Codeium Chat，用户可以获得代码生成、重构、文档编辑甚至错误修复等智能建议，提升编程灵活性和准确性。

部署

相较于 Github Copilot、CodeGPT 以及 MarsCode，Codeium 支持安装在更多 IDE 上，例如 Vim、Jupyter Notebook、Deepnote，具体部署方式可以参考官方文档，以 VSCode 为例，部署方式同样是在 Extension 中找到 Codeium 插件进行安装，安装完成后，VSCode 会提示授权 Codeium，并在右下角弹窗提示授权 Codeium。在模型方面，Codeium 提供六种选择，包括 Base Model（基于 Llama3.1 70B），Codeium Premium（基于 Llama3.1 405B），Claude-3.5-Sonnet，o1-mini 以及 o1-preview。

图表27：Codeium 插件页面展示



注：左图为 VSCode 平台，右图为 PyCharm 平台

资料来源：Codeium 平台，华泰研究

基础功能

代码自动补全（Autocompletion）

Codeium 的核心功能之一是 Autocompletion 自动完成，包含三类应用场景，编程加速、探索以及情景感知。Codeium 由一流的专有模型提供支持，在每次输入后都会建议用户下一步可能想要输入什么。接受建议：TAB；逐步接受建议：Ctrl+►；触发器建议：Alt+\；下一个建议：Alt+]；上一个建议：Alt+[。编程加速场景下，Codeium 可以节省大量击键和时间，尤其是在编写简单、重复或样板代码。探索场景下，在使用不熟悉的库或语言时，Codeium 的自动完成功能，可以极大地帮助生成新的代码想法并跨越编程语言障碍。

编程过程中会遇到最困难的挑战之一，是需要了解代码库的不同部分如何工作以及如何联动组合。情景感知场景下，Codeium 上下文感知引擎可以查找和使用最相关的代码片段，以提高其自动完成建议的质量。上下文感知引擎可以提高输出质量的核心功能之一是固定上下文。用户可以告诉 Codeium 上下文引擎在生成自动补全时应该优先考虑代码库的哪些部分。具体而言，在单击 Chat input 文本框的正上方的 Advanced 按钮，然后单击“Add Context”按钮以添加上下文，类似设定大模型的系统级提示词，这将在整个 Codeium 所有模型中持续发挥效果。

关于如何较好使用自动补全功能，Codeium 官方提供的建议是，自动补全不适用于大规模多层次的代码更改或回答问题，而是更专注于帮助用户提高速度和效率。对于自动补全功能，用户可以选择性接受建议，以提高代码质量，同时避免被低质量的建议分散精力；此外，Codeium 的功能本质上也是与大模型对话，因此可以使用一定的提示词技巧、清晰的代码、文档、注释以及变量命名来帮助 Codeium 去理解帮助用户。

命令（Command）

Codeium 的命令包含两个部分，Generation 自动生成以及 Edits 编辑。通常当编写一个简单的代码块时，通过自然语言描述来生成代码，相较于自动补全会更容易且迅速。因此当用户希望指示 Codeium 在文件中编写一些代码块的情况，建议使用 Codeium Command。对于自动生成，用户可以使用快捷键 Ctrl+I 打开行内编辑器，输入对函数或代码的描述，Alt+A 来接受。编辑功能则是选中想要修改的代码，同样打开 Ctrl+I，输入修改要求，新生成的内容以及原有代码会分别标红和标绿方便用户对比。需要注意的是命令功能仅支持在 VSCode 和 Jetbrains 上使用。

聊天（Chat）

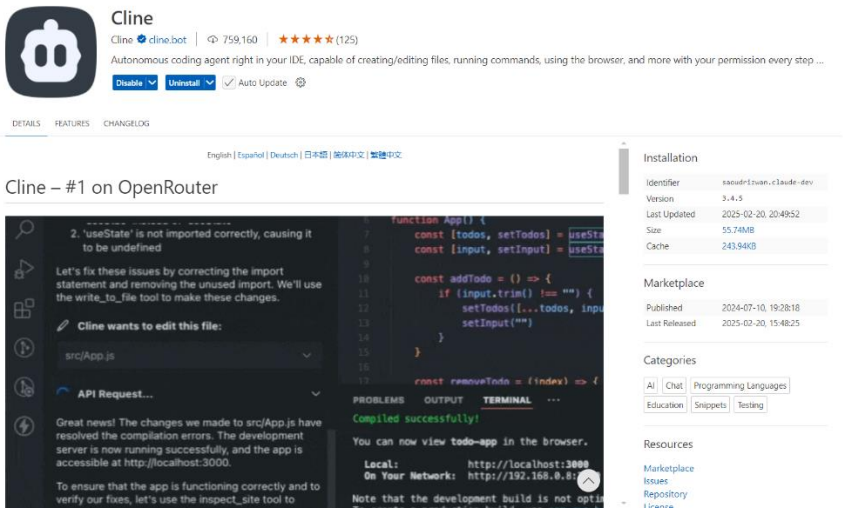
用户可以使用快捷键 Ctrl+Alt+A 快速打开对话框，基础功能与其他插件类似，包括根据代码生成文档、生成注释以及重构代码。其次，用户也可以通过@特定代码、库或者是文档，引用这些内容帮助 Codeium 提高输出质量以及准确度。以上功能仅在 VS Code，JetBrains，Eclipse，Visual Studio，Xcode，Vim/Neovim 上可以使用。

Cline：新兴开源自动化编程插件

Cline（**CLI** a**N**d **E**ditor）是 2024 年 7 月 10 推出的一款开源的 VSCode 插件，是一个可以操作 CLI（Command-line interface，命令行界面）和编辑器的 AI 编程助手。其核心功能在于将多模型能力深度集成至开发流程，支持通过 OpenAI、DeepSeek、Google Gemini 等主流模型的 API 实现智能代码生成与优化。

不同于传统编程助手仅提供代码补全功能，Cline 通过源代码 AST（抽象语法树）分析和正则表达式搜索实现项目级代码重构，能主动创建和编辑文件、探索大型项目以及执行终端命令等。同时，Cline 也支持在浏览器中启动网站，以进行可视化调试，例如当任务完成时，Cline 将通过终端命令如 open -a "Google Chrome" index.html 展示结果，用户可通过点击按钮运行该命令。Cline 较为擅长处理复杂任务，例如根据自然语言指令创建符合项目结构的 Vue 组件或 Python 脚本，具备精准的上下文理解能力。

图表28：Cline 插件页面展示



注：插件来源于 VSCode 平台
资料来源：Cline，VSCode 平台，华泰研究

部署

当前 Cline 仅支持在 VSCode（或依赖于 VSCode 的 Cursor）环境中部署。VSCode—扩展中搜索 Cline 安装即可运行 Cline。与 Github Copilot 不同，Cline 需提前设置大模型 API 才能使用，其支持的模型 API 格式类型如下表：

图表29：Cline 支持的 API 格式类型

OpenRouter	Anthropic	AWS Bedrock
OpenAI Compatible	GCP Vertex AI	Google Gemini
DeepSeek	Mistral	OpenAI
VS Code LM API	Requesty	Together
Alibaba Qwen	LM Studio	Ollama
LiteLLM		

资料来源：Cline，VSCode 平台，华泰研究

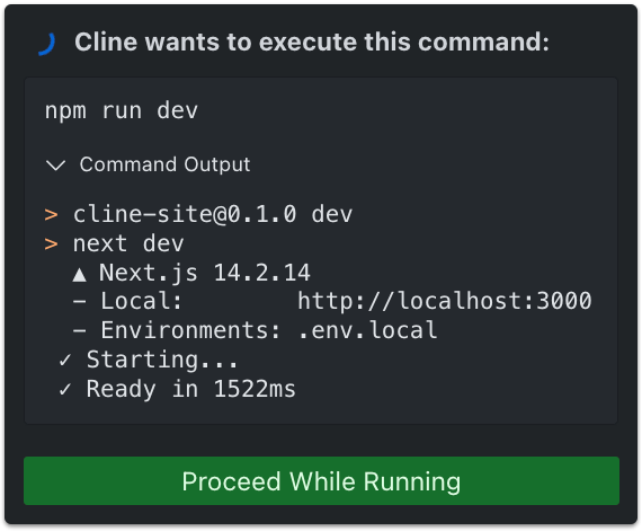
基础功能：运行终端命令、创建修改文件、使用浏览器等

运行终端命令

Cline 可以直接在用户终端中执行命令并接收输出。这一能力使得 Cline 能够执行广泛的任务，从安装包和运行构建脚本到部署应用程序、管理数据库和执行测试，同时适应用户的开发环境和工具链以正确完成工作。

对于长时间运行的进程如开发服务器，使用“在运行时继续”按钮让 Cline 在命令后台运行时继续任务。当 Cline 工作时，它会在过程中收到任何新的终端输出通知，从而对可能出现的问题做出反应，例如编辑文件时的编译时错误。

图表30：Cline 执行终端命令图示



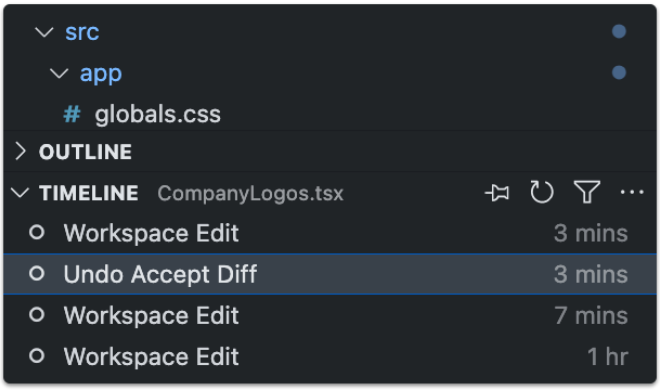
资料来源：Cline，VSCode 平台，华泰研究

创建和编辑文件

Cline 能在用户编辑器中直接创建和编辑文件，向用户展示更改的差异视图。用户可以直接在差异视图编辑器中编辑或恢复 Cline 的更改，或在聊天中提供反馈，直到对结果满意。同时，Cline 还会监控 linter/编译器错误（缺少导入、语法错误等），以便于在过程中自行修复出现的问题。

此外，Cline 所做的所有更改都会记录在用户的文件时间轴中，提供了一种简单的方法来跟踪和恢复对文件的修改。

图表31：Cline 文件修改记录图示

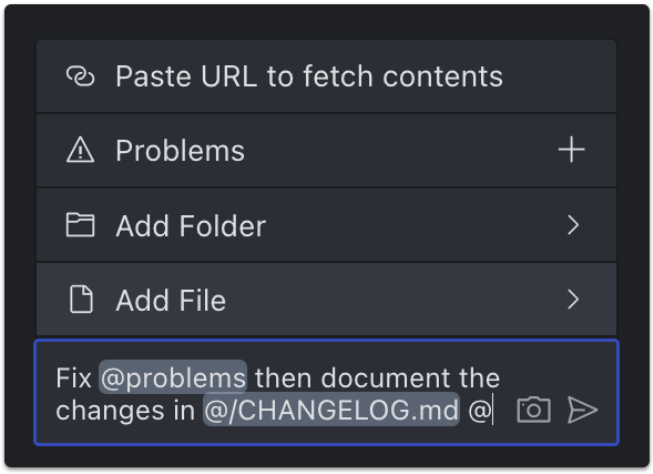


资料来源：Cline，VSCode 平台，华泰研究

添加上下文

Cline 可通过底部@图标为当前任务增加上下文，包括 url、problems、file、folder 等。
@url：粘贴一个 URL 以供扩展获取并转换为 markdown，当用户想给 Cline 提供最新文档时非常有用。
@problems：添加工作区错误和警告（“问题”面板）以供 Cline 修复。
@file：添加文件内容，这样用户便不必浪费 API 请求批准读取文件。
@folder：一次添加文件夹的文件，以进一步加快工作流程。

图表32：Cline 增加上下文图示

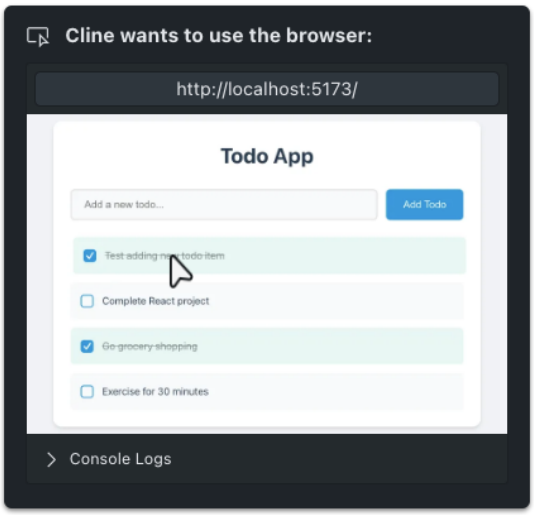


资料来源：Cline，VSCode 平台，华泰研究

使用浏览器

Cline 可以启动浏览器，点击元素，输入文本和滚动，在每一步捕获截图和控制台日志。这一功能允许用户进行交互式调试、端到端测试，使得 Cline 能够自主修复视觉错误和运行时存在的问题，而无需用户亲自操作和复制粘贴错误日志。本功能仅限 Claude 3.5 Sonnet 模型下使用。

图表33：Cline 请求使用浏览器图示

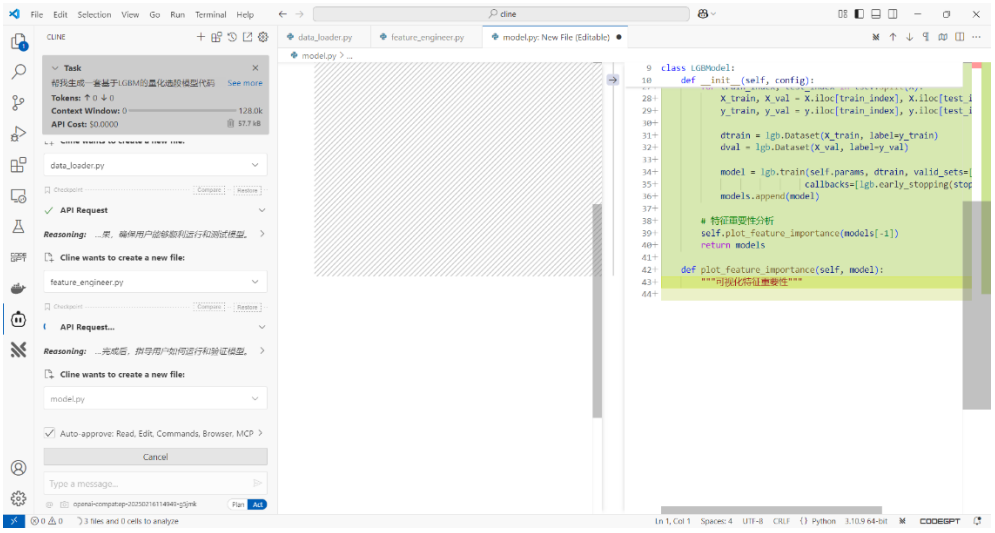


资料来源：Cline，VSCode 平台，华泰研究

核心功能：自动化编程（Task）与 MCP

Cline 的核心功能是自动化编程，通过新增任务（New Task）实现。在任务中有两种模式：一种是“Plan”模式，在该模式下，Cline 将收集信息来构建编程任务；另一种模式是“Act”模式，Act 模型默认被选择，Cline 将立即采取行动已完成用户指示的任务。例如下图中，在 Act 模式下，我们输入指示：“帮我生成一套基于 LGBM 的量化选股模型代码”，Cline 立即开始生成相关代码，最终逐一生成数据加载模块（data_loader.py）、特征工程模块（feature_engineer.py）、模型训练与验证模块（model.py）、回测系统（backtest.py）和主执行文件（main.py）共 5 个模板。

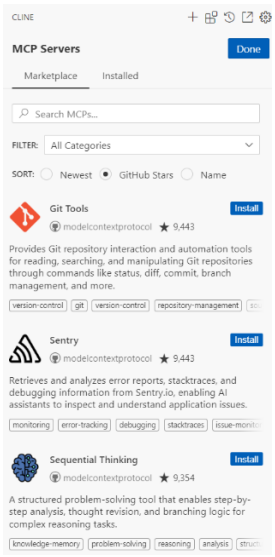
图表34：Cline 自动生成代码文件图示



资料来源：Cline，VSCode 平台，华泰研究

此外，Cline 在 3.4 版本中引入了 MCP 功能，极大拓展了 Cline 的能力边界。MCP（Model Context Protocol，模型上下文协议）是一种开放协议，通过标准化的服务器实现，使 AI 模型能够安全地与本地和远程资源进行交互。简单来说，MCP 是大模型的“万能插件”，能够有效扩展大模型能力，例如文件访问、数据库连接、浏览器自动化等等。除 Cline 外，下文中的 Cursor 同样拥有 MCP 功能。

图表35：Cline MCP Server 界面



资料来源：Cline，VSCode 平台，华泰研究

插件类工具对比

图表36：插件类大模型辅助编程工具对比

特性	Github Copilot	CodeGPT	MarsCode	Codeium	Cline
价格	10$/Month or 100$/Year VSCode 免费使用	专业版: 9.16$/Month	免费	专业版：15$/Month	免费
支持的 IDE	VSCode，JetBrains，Vim/Neovim	VSCode， JetBrains	VSCode， JetBrains	支持 17 多种编辑器	VSCode
功能	代码补全、聊天、命令行支持、拉取请求描述生成（仅限企业版）。	代码生成、完成、调试辅助等功能，支持创建 AI 代理，支持调用本地模型	代码补全、代码生成、代码解释、单元测试生成、问题修复、智能问答等功能。	代码补全、代码生成、聊天、搜索等功能。	运行终端命令、创建修改文件、使用浏览器等，核心功能是全自动化编程
特色	依托于 Guthub，可直接接入 GitHub 官网，响应迅速	CodeBuilder、CodeInterpreter 以及 Stack Overflow 功能，拓展插件类辅助编程工具边界；支持接入本地大模型，且提供模型选择最多；支持 API 接口	使用国内豆包代码大模型，且完全免费	情景感知功能	可外接 MCP server，为模型提供更丰富的外部工具

资料来源：各官方文档，华泰研究

IDE 类

IDE 类大模型辅助编程工具是一种直接集成到集成开发环境（IDE）中的独立智能编程助手，它们通过实时分析开发者的编程上下文、项目结构和编码意图，提供智能化的编程辅助服务，在插件类的基础上拥有更高的权限，与开发环境的结合更为紧密。这些工具可以理解当前代码文件的结构、上下文关系以及项目依赖，在开发者编写代码时提供实时的智能建议。

关于大模型辅助编程更高级功能的探索，正如前文 Codeium 官方建议中提及，自动补全功能并不适用于大规模、多层次的代码编写与修改，实际上命令和聊天功能也难以胜任这一任务。CodeGPT 后续推出的 CodeBuilder 功能虽然一定程度上具有大规模代码项目编写的雏形，然而实际使用下来，限制仍然较多且智能程度低。

IDE 类大模型辅助编程工具的出现则填补了这方面的空缺，尤其是多文件多层次协同编辑功能，使得这类辅助编程工具真正脱离出“辅助”的定义，成为独立的代码项目开发智能体。这类工具在兼容插件类工具功能的基础上，更适用于日常编程过程中的项目编写、调试和重构场景，尤其适合处理独立创新的编程任务。代表性的工具如 Cursor、Windsurf 和 MarsCode，它们各自以其独特的功能和优势，帮助开发者提高编程效率和代码质量。

Cursor：辅助编程工具的集大成者

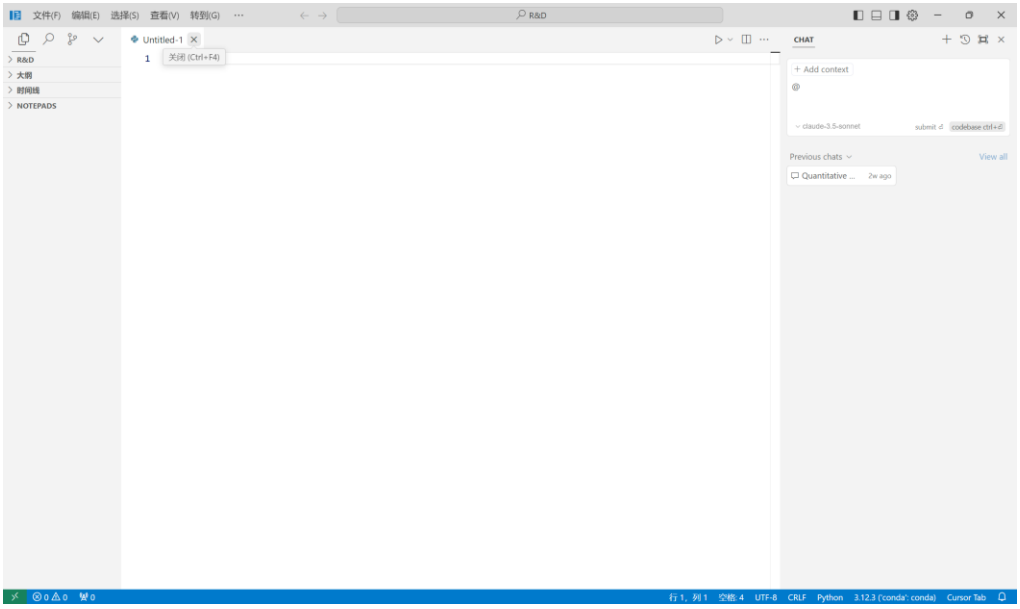
Cursor 是一个基于 VSCode 开发的集成开发环境（IDE），它通过集成先进的大型语言模型（LLMs），如 GPT-4 和 Claude 3.5，为开发者提供了一个强大的 AI 辅助编程助手。这个工具支持多种编程语言，包括但不限于 Python、Java、C#和 JavaScript，并且可以在 Mac、Windows 和 Linux 等不同的操作系统上运行。

Cursor 的特点在于它能够理解代码的上下文，提供智能的代码生成、优化建议和项目结构调整。它还具备自动补全功能和代码索引，能够根据项目的具体情况提供相关的建议和查询。此外，Cursor 还提供了一些快捷键功能，比如 Ctrl/Cmd + L 可以打开对话框，Ctrl/Cmd + K 可以打开生成窗口，而 Ctrl/Cmd + I 则可以打开 Composer，而 Composer 是 IDE 类辅助编程工具特有的功能，允许用户在对话窗口中同时对多个文件进行修改。为了保护用户的隐私和减少不必要的干扰，Cursor 还允许用户将敏感或无关的文件排除在 AI 索引之外。同时由于 Cursor 基于 VSCode 开发，Cursor 用户 UI 以及常见功能与 VSCode 保持一致，经常使用 VSCode 或者 Jetbrains 的用户可以快速上手适应，同时 Cursor 上也支持安装 VSCode 插件。

传统的大模型编程插件（如 GitHub Copilot、Codeium 等）通常只能基于当前文件或局部代码提供建议，这种局限性使得这些工具在处理复杂项目时显得力不从心。而 Cursor 通过索引整个代码库，能够提供更加精准和全面的建议。例如，当用户需要修改一个组件时，Cursor 可以自动识别所有相关文件，并提示需要更新的地方，这种能力在大型项目中尤为重要。

Cursor 安装较为简单，登录 Cursor 官网（https://www.cursor.com/downloads），根据操作系统选择对应版本下载即可，安装完成后，需要注册/登录 Cursor 账户才可正常使用 Cursor 功能。在模型选择方面，Cursor 提供各种模型应对不同场景需求，包括 Claude-3.5-Sonnet、GPT4 与 GPT4o 系列、o1 系列以及 Cursor 自行训练的 Cursor-small 模型。而最新 0.43 版本，支持用户上传照片，进行多模态对话。不同模型要求不同的定价，其中 Cursor-small 模型价格最低，而需要注意的是，即使是订阅用户，使用高级模型的次数以及迅速响应的次数均有限制，当请求次数达到一定水平后，每个请求将成为慢请求。

图表37：Cursor 界面展示



资料来源：Cursor，华泰研究

基础功能：代码补全、行内编辑、对话

代码自动补全（Autocompletion）

Cursor 的自动补全功能与 Github Copilot 等插件类辅助编程工具一致，在体验上也没有显著差异，会根据前文代码或者注释进行联想编程，需要点击 Apply 逐个接受建议或 Apply all 一次性接受所有建议。最新 0.43 版本中，可以自动补全 GitHub 提交信息。

行内编辑（Inline Edit）

行内编辑是插件类辅助编程工具的常见功能之一，嵌入到 Cursor 中的行内编辑功能表现与其余插件类似，可以选中代码，使用快捷键 Ctrl+K 打开行内编辑器，一般而言两种使用方向，包括根据要求直接生成代码块，或者是根据要求修改代码，响应迅速。

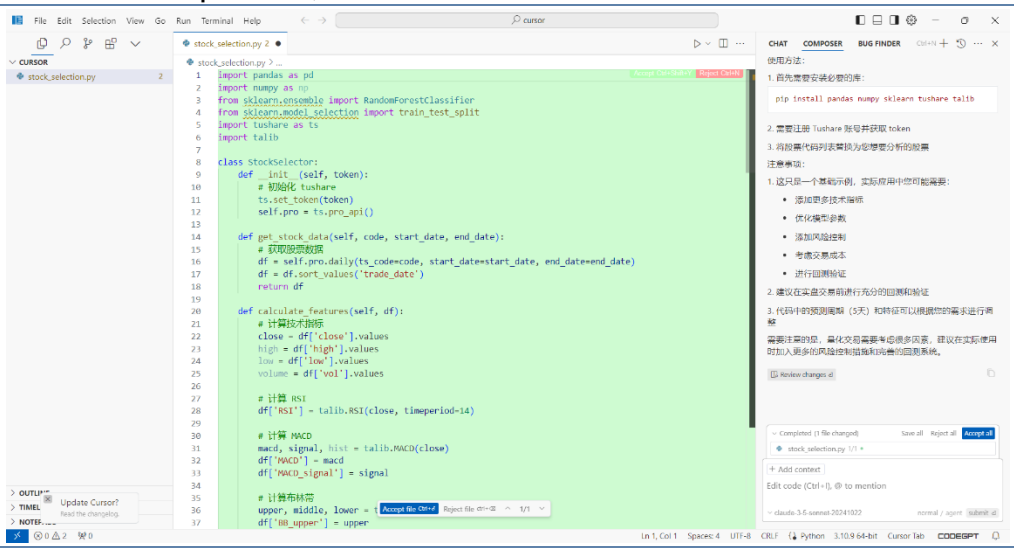
对话（Chat）

在 IDE 类辅助编程工具中，对话功能主要是针对完整文件进行修改的，也是与插件类辅助编程工具一致的功能，可以使用快捷键 Ctrl+L 打开对话框。在指定参考内容方面，Cursor 提供更多选择，不同于插件类工具常常只能@Workplace 等，Cursor 支持指定文件（@Files）、文件夹（@Folders）、代码（@Code）、文档（@Docs）、Git 存放区（@Git）、Notepad（@Notepad）、账号代码库（@Codebase）、潜在错误（@Lint errors）以及在线链接（@Web）。

核心功能：Composer 自动化编程

Cursor Composer 是集成在 Cursor 编辑器中的 AI 驱动工具，是一个改变“游戏规则”的功能，显著加快开放过程。Composer 突破了单行和单文件编辑的局限，让用户能够同时编辑多个文件，根据高级指令生成整个应用程序，或者是利用对项目结构的上下文理解，以及交互式地优化生成的代码。Composer 功能的主要特性是能一次性操作，包括创建或者修改多个文件；根据高级描述开发完整的项目；同时 Composer 的对话将不再局限于单个文件，而是会考虑整个项目结构以及项目代码。用户可以使用快捷键 Ctrl+I 打开 Composer，Ctrl+Shift+I 打开全屏 Composer。使用场景方面，Composer 内容特别适用于快速根据高级指令进行量化策略编写，或是创建小项目；跨多文件实现复杂功能；根据项目重构现有代码。

图表38：Cursor Composer 功能展示

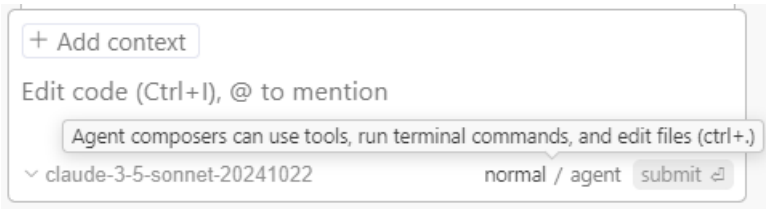


资料来源：Cursor，华泰研究

在 0.43 版本中 Cursor 为 Composer 功能引入了 Agent 功能，填补 Cursor 在自主性和让大模型自主完成更多工作的空白。理论上，Agents 能根据用户的请求找到相关文件，同时对多个文件进行更改，甚至能自动化在终端中运行各种命令并不断迭代代码，而无需用户直接介入。与原始 Composer 功能相比，Agent 自动上下文处理有所提升，需要手动添加的上下文更少，但是并不能完全省去上下文管理的工作。类似于 Cline，Agent 模式下的 Composer 具备调用 MCP server 的能力，大幅扩展了可使用的工具范畴。

尽管 Agent 功能丰富、自动化程度高，但用户仍需注意检查 Agent 的工作。需要注意的是 Cursor Agent 自动创建文件和安装依赖的时候提供的是 Linux 命令，使用 Windows 开发时，可以通过选择默认配置文件，切换到 Git bash 进行 linux 命令支持，而 Windows Power Shell 则不要使用。操作方面，可以在 Composer 对话框下方 normal/agent 切换普通与 Agent 模式。

图表39：Cursor Composer Agent 模式



资料来源：Cursor，华泰研究

在具体使用 Composer 功能时有以下建议：

1. **清晰指令：**在使用 Composer 功能时，建议使用清晰详细的提示词，可以优先使用 Chat 功能中的 GPT-4，尤其是 Claude 模型先对项目进行充分的头脑风暴，将项目细节进行整理并保存到文档中（例如 instruction.md）。
2. **Chat 与 Composer 各有所长，应合理运用：**相较于 Chat 功能更适用于处理小型任务、解释代码/命令，提问以及项目导航，Composer 功能更适用于编写代码，上一步编写 instruction.md 文件后，确保与 Composer 对话过程中使用@引用该文件，并指示根据项目进度更新文档内容。
3. **逐步提出要求：**为了避免模型错误，建议一次只让 Composer 执行一项任务，逐步修改，不要提出复杂的修改要求，每次修改后务必检查代码正确性。此外，由于 Composer 每次对话会调用多个文件，因此如果不希望 Composer 修改其他文件，需要在指令中明确指出修改范围。

4. **结合 Git 进行版本管理：**上述提及 Composer 每次修改可能涉及多个文件，而目前 Cursor 版本管理功能暂不齐全，因此建议每次接受重大改变时，推送至 Git 确保代码可以回溯，或是点击上次对话上方 checkout 按钮进行回溯。
5. **合理使用 CursorRules：**用户可以通过设置中设定 CursorRules 确定贯穿全局的指令，类比设定系统提示词，例如可以规范代码位置与开发流程，确保 Composer 不会创建平行目录，或者是按照需求创建项目结构。此举能大幅减少项目开发中干预的工作量，同时规范代码以及项目。用户同样可以自行创建.cursorrules 文件，将规则写入，并在对话过程中引用。关于 CursorRules 的编写，可以参考 https://cursor.directory/中用户自制的基础规则，根据自身需求调整修改。

Windsurf：AI Flow 范式下的新兴 IDE

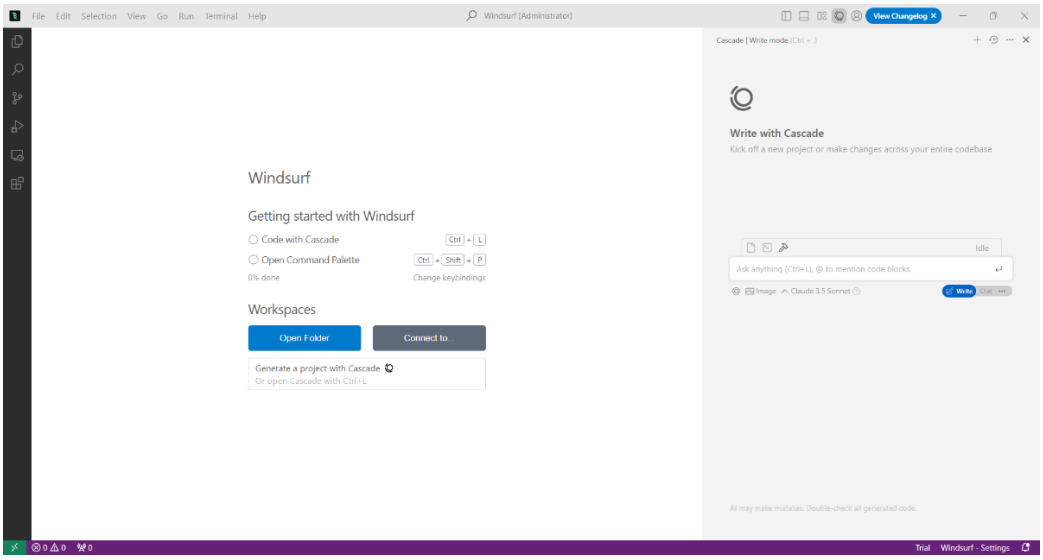
Windsurf 是出 Codeium 推出的 AI 驱动的集成开发环境（IDE），旨在提升开发者的编程效率和协作体验。通过结合 AI 助手与独立代理(Agent)功能，Windsurf 支持多文件编辑、自然语言命令执行、上下文感知等智能化功能帮助开发者在编写代码时保持“Flows”状态。Windsurf 适用于个人开发者、团队协作及各种编程任务，提供跨平台支持，优化开发工作流程。Windsurf 官方声称是首个具备 Agent 功能的 IDE，其核心范式是“AI Flow”，由 Agent 和 Copilot 共同构成，支持多步骤、多工具协同，自动维护上下文状态，智能任务规划和执行。在 AI Flow 范式下，Windsurf 中的 AI 能够像 Copilot 一样与开发者协作，或者像代理一样独立完成复杂任务。

图表40：Windsurf 的 AI Flow 范式

资料来源：Windsurf 官网，华泰研究

Windsurf 安装只需要从 https://codeium.com/windsurf 官网上下载安装，并运行安装。开始运行后，可以选择从 Cursor 或从 VSCode 上导入配置，可以选择迁移设置、扩展或者同时迁移两者，方便用户上手。设置保存后，用户需要注册/登录 Codeium 账户。模型选择方面，Windsurf 提供 GPT-4o 以及 Claude-Sonnet-3.5 两种模型。

图表41：Windsurf 页面展示



资料来源：Windsurf，华泰研究

Windsurf 中拥有与 Cursor Composer 的类似功能，名为 Cascade。Cascade 有两种模式：编辑和聊天，相当于将 Cursor 中 Chat 和 Composer 功能进行合并，用户可以选择聊天模式和写入模型进行自由切换。写入模式允许 Cascade 创建和修改用户代码库，而聊天模式针对有关代码库或一般编码原则的问题进行了优化回答。其中自动补全（Tab 功能）、行内编辑器（Ctrl+I）以及 Cascade 聊天（Chat 功能）与 Cursor 和其他插件类辅助编程工具一致。

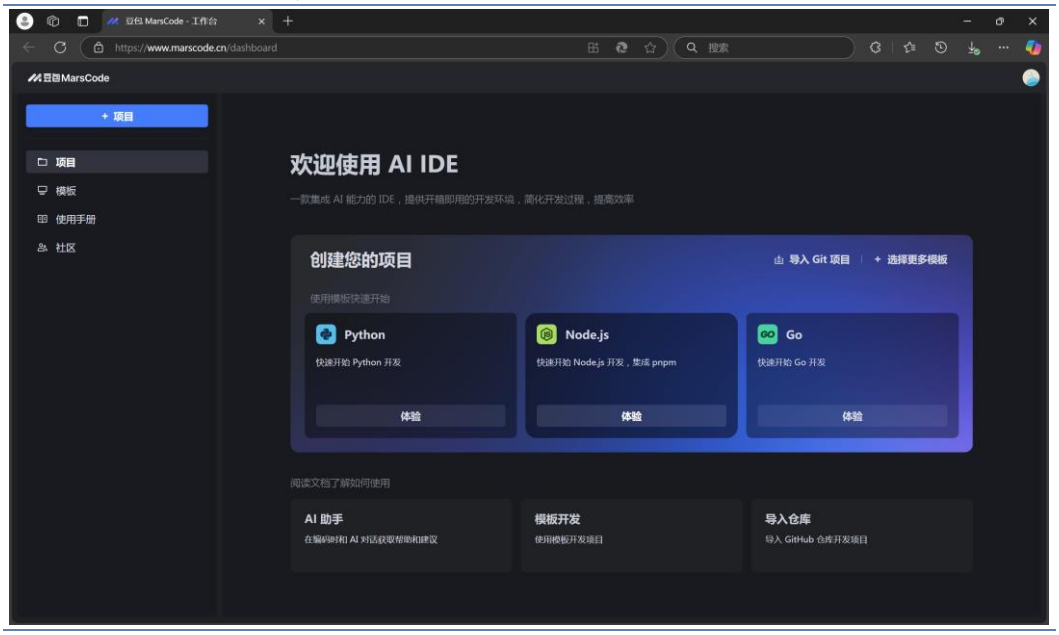
在 Windsurf 的 Agent 功能中，用户无需通过上下文提示 AI 了解先前操作，Agent 本身可以感知上下文以及代码库，例如当重命名变量后，仅需要提示 Cascade continue，即可自动检测变量名称改变，从而重命名其他实例。此外，Cascade 可以检测用户正在使用的包和工具，识别需要安装的项，甚至可以自动进行安装，用户只需向 Cascade 询问如何运行项目并点击“接受”。

在高级功能方面，像 VSCode 一样，Windsurf 实现 SSH 支持，仅需要安装 OpenSSH，目前仅支持连接到基于 Linux x64 的远程主机；此外 Windsurf 同时也支持通过以下命令使用 DevContainer：①Open Folder in Container，使用指定的 devcontainer.json 文件在新的工作区中打开文件夹，②Reopen in Container，在新容器中重新打开当前工作区，指定 devcontainer.json 文件以配置该容器，③Attach to Running Container，如果已经有一个运行中的开发容器，用户可以将远程服务器附加到该容器，并将当前工作区连接到该容器。目前 Windsurf 暂时不支持 SSH+DevContainer 与 WSL 功能。

MarsCode：轻量化云端 IDE

MarsCode IDE 是一款集成了大模型技术的云端集成开发环境（IDE），它通过内置的 AI 编程助手，提供了代码补全、生成、解释和调试等功能，旨在简化软件开发流程。MarsCode 重点在轻量化，除去自动代码补全生成等基础功能，MarsCode 支持从模板创建项目以及从 Git 导入项目，最重要的是，MarsCode 仅支持在网页上使用，无需配置开发环境，也不会受到本地计算机能力限制。MarsCode 提供 C、C++、C#、Go、JavaScript、Java、Node.js、Rust、TypeScript 等语言的开发环境和模板，避免繁琐的环境配置。目前单个账号在豆包 MarsCode IDE 上创建的项目数量不得超过 10 个。MarsCode 云端服务器资源为两核 CPU、4GiB 内存以及 10GiB 硬盘的环境，满足用户轻度使用。

图表42：MarsCode IDE 页面展示



资料来源：MarsCode，华泰研究

图表43：MarsCode 快捷键汇总（以 windows 系统为例）

行动	快捷键
换行编辑	Ctrl+Enter
触发代码自动补全	Ctrl+Space
打开侧边对话框	Ctrl+U
打开内嵌对话框	Ctrl+I

资料来源：MarsCode 文档，华泰研究

整体而言，相较于独立的 IDE，MarsCode IDE 更像将其云端服务器与插件类辅助编程工具结合，实现的轻量化开发平台，具体使用内容与操作逻辑与 MarsCode 插件工具一致。

大模型辅助编程实践

以 Cursor 为例

本文使用 Cursor Composer 功能展示如何使用 IDE 类辅助编程工具实现 SMA 简单移动平均线量化策略的编写，包含数据获取、因子生成、因子测试以及策略回测部分。遵从关于 Composer 功能的规则，本文设计提示词，从模糊的项目内容，逐步生成具体策略。

首先，使用 Ctrl+L 打开对话框功能，输入以下提示词，使用 Claude-Sonnet-3.5 模型头脑风暴，生成关于项目每个文件的具体功能、依赖项等细节内容。

图表44：Chat 生成项目结构对话提示词

提示词

role
Professional quantitative analyst
skills
1. A deep knowledge of mathematics and statistics, including basic methods such as probability theory, statistics, time series analysis and regression analysis, as well as the application of advanced machine learning models such as XGBoost, GRU models
2. Proficient programming skills, especially in languages such as Python, R, and C++, for data processing, model building, and backtesting
3. Powerful data analysis capabilities, able to extract valuable information from massive data, and use machine learning and artificial intelligence technology to predict and optimize
4. Have an in-depth understanding of financial markets, including various financial instruments, market microstructures and trading strategies, and can design and implement new quantitative models and strategies
task
You need to design a set of prompt words, the prompt for the assistant program cursor to write a double average quantization strategy, need to implement the whole content framework from 1.data acquisition, 2.factor generation, 3.factor testing, 4.strategy backtesting and 5. run the entire system, you need to answer in English, and for each part, you need to point out what python package should be used for each function. This also needs to be included in the prompt word
sample output
Step1 Data Obtain
{
File name: Data obtain.py
Function: A Python script to fetch historical stock price data for a specified stock symbol, including open, high, low, close prices, and volume.
Package: akshare, pandas
Correlation with previous result: None
Result save path: ./Data/data.parquet
}
Step2 Factor Generation
{
File name: Factor Generation.py
Function: A Python function to calculate two moving averages for a given stock price dataset, one short-term and one long-term, and identify crossover points as potential trading signals.
Package: numpy, pandas
Correlation with previous result: Use ./Data/data.parquet as input to generate
Result save path: ./Data/data.parquet
}
...
attention
1. Only give the output directly, don't explain or give other information
2. For each part, consider carefully, select the most appropriate package
3. Write config.py, requirement.txt and readme.md for this project
4. Visualize the result in factor test and backtesting, use package backtesting in backtest module, akshare in data obtain

资料来源：Cursor，华泰研究

下表展示 Chat 功能对话返回的结果，可见整体项目框架清晰可行，同时不同文件之间的联系也具体标注，方便后续 Composer 进行读取，生成结构严谨的项目。

图表45： Cursor Chat 生成项目结构对话结果

Response

Step1 Data Obtain
{
File name: data_obtain.py
Function: Fetch historical stock price data from akshare API, including OHLCV data for specified stock symbols
Package: akshare, pandas
Correlation with previous result: None
Result save path: ./data/raw_data.parquet
}

Step2 Factor Generation
{
File name: factor_generation.py
Function: Calculate short-term (5-day) and long-term (20-day) moving averages, generate crossover signals
Package: numpy, pandas, ta-lib
Correlation with previous result: Use ./data/raw_data.parquet
Result save path: ./data/factor_data.parquet
}

Step3 Factor Testing
{
File name: factor_testing.py
Function: Analyze factor effectiveness through IC test, factor returns analysis, and visualization
Package: scipy, statsmodels, matplotlib, seaborn
Correlation with previous result: Use ./data/factor_data.parquet
Result save path: ./results/factor_analysis.html
}

Step4 Strategy Backtesting
{
File name: strategy_backtesting.py
Function: Implement double moving average strategy, calculate returns, drawdowns, and performance metrics
Package: backtesting, pandas, numpy
Correlation with previous result: Use ./data/factor_data.parquet
Result save path: ./results/backtest_results.html
}

Step5 Main System
{
File name: main.py
Function: Orchestrate the entire workflow, from data fetching to strategy execution
Package: logging, argparse
Correlation with previous result: Coordinates all previous modules
Result save path: ./logs/system.log
}

config.py:
```python
STOCK_CODE = "000001"
START_DATE = "2020-01-01"
END_DATE = "2023-12-31"
SHORT_WINDOW = 5
LONG_WINDOW = 20
INITIAL_CAPITAL = 1000000
```

requirements.txt:
```
akshare>=1.0.0
pandas>=1.5.0
```

Response

numpy>=1.21.0
ta-lib>=0.4.0
scipy>=1.7.0
statsmodels>=0.13.0
matplotlib>=3.5.0
seaborn>=0.11.0
backtesting>=0.3.0
```

README.md:
```markdown
# Double Moving Average Trading Strategy

## Overview
Quantitative trading strategy based on short and long-term moving average crossovers.

## Installation
1. Install requirements: `pip install -r requirements.txt`
2. Install TA-Lib system dependencies
3. Configure parameters in config.py

## Usage
Run: `python main.py`

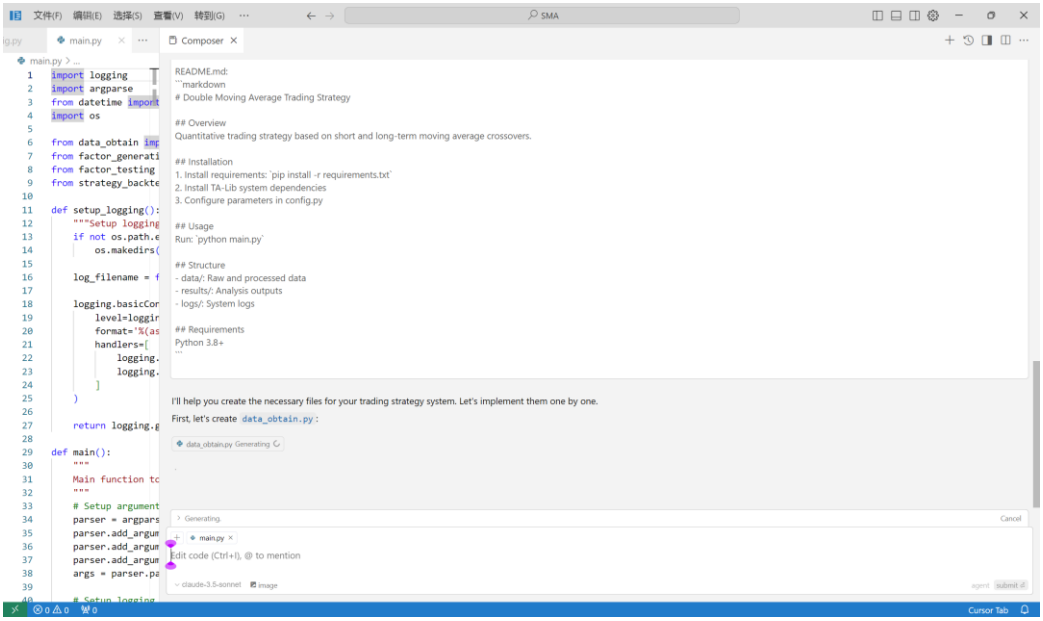
## Structure
- data/: Raw and processed data
- results/: Analysis outputs
- logs/: System logs

## Requirements
Python 3.8+
```

资料来源：Cursor，华泰研究

将上述对话结果输入 Composer，并点击 normal/agent 按键启动 agent 功能，点击 submit，对于每个生成的文件，需要点击 accept 以保存进度。

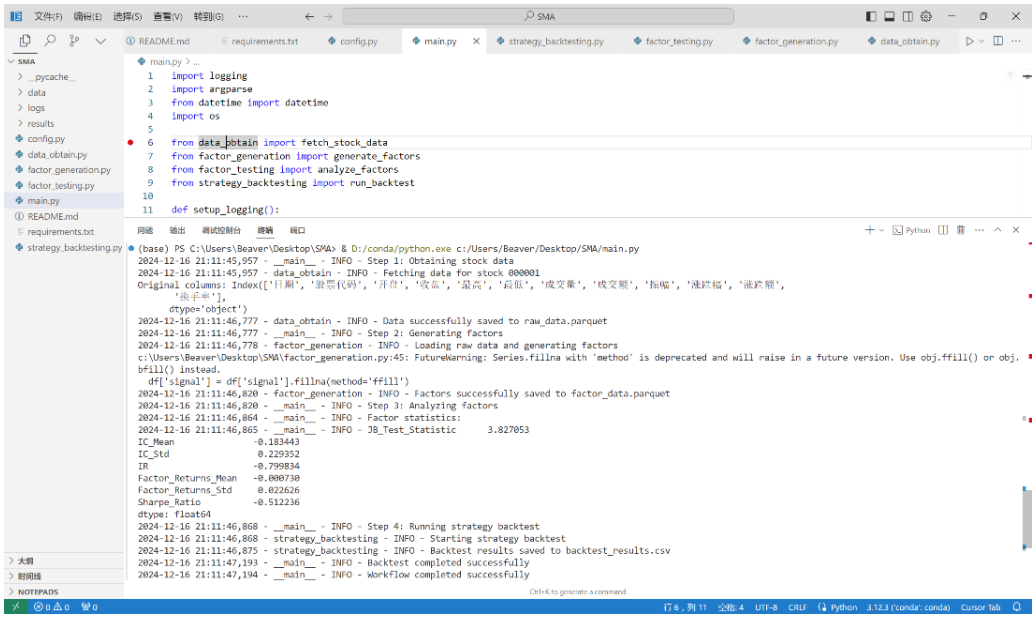
图表46：Cursor Composer Agent 编写 SMA 策略过程页面展示



资料来源：Cursor，华泰研究
```

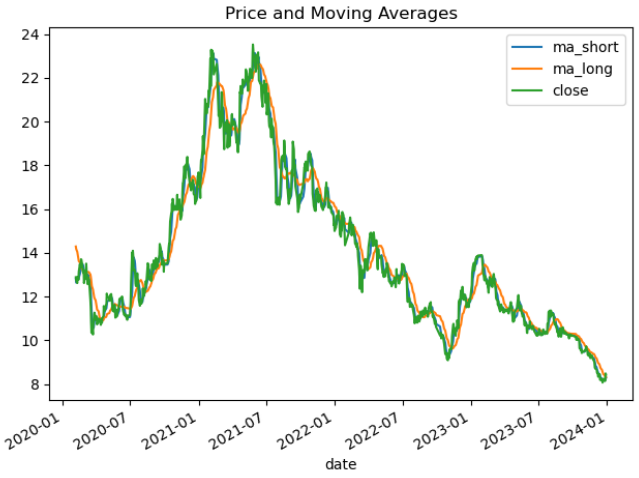
初步生成代码框架后，需要根据运行情况进行调整，常见报错包括 columns 名称不匹配，以及对于 Jarque-Bera 测试数据量不足等，具体修改过程可以将报错信息输入 Composer，并指定 Composer 仅修改相关文件，逐步解决报错信息，最终运行成功。

图表47： Cursor Composer Agent 编写 SMA 策略运行过程日志



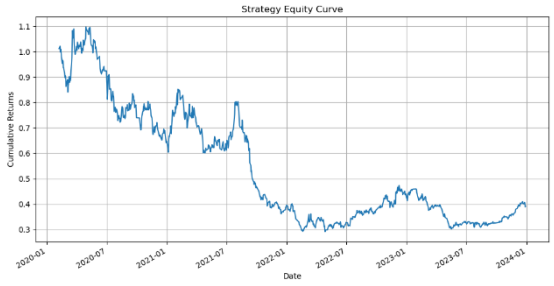
资料来源：Cursor，华泰研究

图表48： Cursor Composer Agent 编写 SMA 权益曲线可视化



资料来源：Cursor，华泰研究

图表49： Cursor Composer Agent 编写 SMA 策略回测结果



资料来源：Cursor，华泰研究

总结

本文提供了一份详尽的大模型辅助编程实用手册。在探讨大模型辅助编程工具的分类时，本文根据大模型嵌入开发环境的程度，将工具分为两类：插件类大模型辅助编程工具和 IDE 类大模型辅助编程工具，并以 Github Copilot、CodeGPT、MarsCode、Codeium 和 Cline，与 Cursor、Windsurf 和 MarsCode IDE 为例，详细介绍了这些工具的功能特点和部署流程，同时以不同维度对比分析了不同工具的特色及优势。

针对大模型辅助编程应用，本文以 Cursor Composer 功能为例，展示了其在量化策略编写等方面的实例构建过程。从实践效果来看，这些工具显著提升了编程效率和代码质量，进一步表明大模型辅助编程工具在提升投资研究和策略开发效率方面的重要价值。

总结而言：

1. 对于大模型辅助编程工具的选择，从用户体验和功能完备性来看，Cline 和 Cursor 尤为突出。Cline 支持用户自行嵌入模型 API，便于用户使用开源、低成本或有特定维度优势的模型，与此同时，Cline 拥有高度自动化的辅助编程功能，能够自行创建、编辑文件，以及调用外部工具，从而给予用户极佳的使用体验；而 Cursor Composer 则通过其 IDE 类的集成环境，提供了更为直观和全面的编程辅助，适合大型项目的开发，最新支持的 Agent 模式也极大提高其智能化程度。
2. 在考虑模型选择、数据敏感性问题时，Cline 和 CodeGPT 作为支持本地大模型部署、同时兼顾多种国产大模型接口的辅助工具方案，适合对代码数据安全性要求较高，或者是对模型选择有特殊要求的用户。
3. 大模型辅助编程工具的效果不仅取决于工具本身的功能，还受限于大模型的能力和开发者如何有效利用这些工具。因此，选择合适的模型并结合具体的编程实践，对于提升编程效率和代码质量至关重要。

本文尝试帮助开发者全面地了解大模型辅助编程工具的部署策略、功能特点和实际应用，以便更好地利用这些工具来提升编程工作流程的效率和智能化水平。本文亦有诸多不足之处，例如，本文尚未能够完全覆盖所有辅助编程应用产品，包括 Trae、Continue、通义灵码等，以及各产品频繁的功能更新，亦可能无法全面覆盖。总之，大模型辅助编程应用正处于蓬勃发展阶段，或正是开发者深入体验和探索的优质时机。

风险提示

大模型是海量数据训练获得的产物，输出准确性可能存在风险；不同大模型辅助编程工具效果存在差距，对于大模型生成的代码，需要谨慎参考；大模型辅助编程工具功能及稳定性可能受到版本切换影响。

免责声明

分析师声明

本人，林晓明、何康，兹证明本报告所表达的观点准确地反映了分析师对标的证券或发行人的个人意见；彼以往、现在或未来并无就其研究报告所提供的具体建议或所表达的意见直接或间接收取任何报酬。

一般声明及披露

本报告由华泰证券股份有限公司（已具备中国证监会批准的证券投资咨询业务资格，以下简称“本公司”）制作。本报告所载资料是仅供接收人的严格保密资料。本报告仅供本公司及其客户和其关联机构使用。本公司不因接收人收到本报告而视其为客户。

本报告基于本公司认为可靠的、已公开的信息编制，但本公司及其关联机构(以下统称为“华泰”)对该等信息的准确性及完整性不作任何保证。

本报告所载的意见、评估及预测仅反映报告发布当日的观点和判断。在不同时期，华泰可能会发出与本报告所载意见、评估及预测不一致的研究报告。同时，本报告所指的证券或投资标的的价格、价值及投资收入可能会波动。以往表现并不能指引未来，未来回报并不能得到保证，并存在损失本金的可能。华泰不保证本报告所含信息保持在最新状态。华泰对本报告所含信息可在不发出通知的情形下做出修改，投资者应当自行关注相应的更新或修改。

本公司不是 FINRA 的注册会员，其研究分析师亦没有注册为 FINRA 的研究分析师/不具有 FINRA 分析师的注册资格。

华泰力求报告内容客观、公正，但本报告所载的观点、结论和建议仅供参考，不构成购买或出售所述证券的要约或招揽。该等观点、建议并未考虑到个别投资者的具体投资目的、财务状况以及特定需求，在任何时候均不构成对客户私人投资建议。投资者应当充分考虑自身特定状况，并完整理解和使用本报告内容，不应视本报告为做出投资决策的唯一因素。对依据或者使用本报告所造成的一切后果，华泰及作者均不承担任何法律责任。任何形式的分享证券投资收益或者分担证券投资损失的书面或口头承诺均为无效。

除非另行说明，本报告中所引用的关于业绩的数据代表过往表现，过往的业绩表现不应作为日后回报的预示。华泰不承诺也不保证任何预示的回报会得以实现，分析中所做的预测可能是基于相应的假设，任何假设的变化可能会显著影响所预测的回报。

华泰及作者在自身所知情的范围内，与本报告所指的证券或投资标的不存在法律禁止的利害关系。在法律许可的情况下，华泰可能会持有报告中提到的公司所发行的证券头寸并进行交易，为该公司提供投资银行、财务顾问或者金融产品等相关服务或向该公司招揽业务。

华泰的销售人员、交易人员或其他专业人士可能会依据不同假设和标准、采用不同的分析方法而口头或书面发表与本报告意见及建议不一致的市场评论和/或交易观点。华泰没有将此意见及建议向报告所有接收者进行更新的义务。华泰的资产管理部门、自营部门以及其他投资业务部门可能独立做出与本报告中的意见或建议不一致的投资决策。投资者应当考虑到华泰及/或其相关人员可能存在影响本报告观点客观性的潜在利益冲突。投资者请勿将本报告视为投资或其他决定的唯一信赖依据。有关该方面的具体披露请参照本报告尾部。

本报告并非意图发送、发布给在当地法律或监管规则下不允许向其发送、发布的机构或人员，也并非意图发送、发布给因可得到、使用本报告的行为而使华泰违反或受制于当地法律或监管规则的机构或人员。

中国香港

本报告由华泰证券股份有限公司制作,在香港由华泰金融控股（香港）有限公司向符合《证券及期货条例》及其附属法律规定的机构投资者和专业投资者的客户进行分发。华泰金融控股（香港）有限公司受香港证券及期货事务监察委员会监管，是华泰国际金融控股有限公司的全资子公司，后者为华泰证券股份有限公司的全资子公司。在香港获得本报告的人员若有任何有关本报告的问题,请与华泰金融控股（香港）有限公司联系。

香港-重要监管披露

- 华泰金融控股（香港）有限公司的雇员或其关联人士没有担任本报告中提及的公司或发行人的高级人员。
- 有关重要的披露信息，请参华泰金融控股（香港）有限公司的网页 https://www.htsc.com.hk/stock_disclosure

其他信息请参见下方 **“美国-重要监管披露”**。

美国

在美国本报告由华泰证券（美国）有限公司向符合美国监管规定的机构投资者进行发表与分发。华泰证券（美国）有限公司是美国注册经纪商和美国金融业监管局（FINRA）的注册会员。对于其在美国分发的研究报告，华泰证券（美国）有限公司根据《1934 年证券交易法》（修订版）第 15a-6 条规定以及美国证券交易委员会人员解释，对本研究报告内容负责。华泰证券（美国）有限公司联营公司的分析师不具有美国金融监管（FINRA）分析师的注册资格，可能不属于华泰证券（美国）有限公司的关联人员，因此可能不受 FINRA 关于分析师与标的公司沟通、公开露面和所持交易证券的限制。华泰证券（美国）有限公司是华泰国际金融控股有限公司的全资子公司，后者为华泰证券股份有限公司的全资子公司。任何直接从华泰证券（美国）有限公司收到此报告并希望就本报告所述任何证券进行交易的人士，应通过华泰证券（美国）有限公司进行交易。

美国-重要监管披露

- 分析师林晓明、何康本人及相关人士并不担任本报告所提及的标的证券或发行人的高级人员、董事或顾问。分析师及相关人士与本报告所提及的标的证券或发行人并无任何相关财务利益。本披露中所提及的“相关人士”包括 FINRA 定义下分析师的家庭成员。分析师根据华泰证券的整体收入和盈利能力获得薪酬，包括源自公司投资银行业务的收入。
- 华泰证券股份有限公司、其子公司和/或其联营公司，及/或不时会以自身或代理形式向客户出售及购买华泰证券研究所覆盖公司的证券/衍生工具，包括股票及债券（包括衍生品）华泰证券研究所覆盖公司的证券/衍生工具，包括股票及债券（包括衍生品）。
- 华泰证券股份有限公司、其子公司和/或其联营公司，及/或其高级管理层、董事和雇员可能会持有本报告中所提到的任何证券（或任何相关投资）头寸，并可能不时进行增持或减持该证券（或投资）。因此，投资者应该意识到可能存在利益冲突。

新加坡

华泰证券（新加坡）有限公司持有新加坡金融管理局颁发的资本市场服务许可证，可从事资本市场产品交易，包括证券、集体投资计划中的单位、交易所交易的衍生品合约和场外衍生品合约，并且是《财务顾问法》规定的豁免财务顾问，就投资产品向他人提供建议，包括发布或公布研究分析或研究报告。华泰证券（新加坡）有限公司可能会根据《财务顾问条例》第 32C 条的规定分发其在华泰内的外国附属公司各自制作的信息/研究。本报告仅供认可投资者、专家投资者或机构投资者使用，华泰证券（新加坡）有限公司不对本报告内容承担法律责任。如果您是非预期接收者，请您立即通知并直接将本报告返回给华泰证券（新加坡）有限公司。本报告的新加坡接收者应联系您的华泰证券（新加坡）有限公司关系经理或客户主管，了解来自或与所分发的信息相关的事宜。

评级说明

投资评级基于分析师对报告发布日后 6 至 12 个月内行业或公司回报潜力（含此期间的股息回报）相对基准表现的预期（A 股市场基准为沪深 300 指数，香港市场基准为恒生指数，美国市场基准为标普 500 指数，台湾市场基准为台湾加权指数，日本市场基准为日经 225 指数，新加坡市场基准为海峡时报指数，韩国市场基准为韩国有价证券指数，英国市场基准为富时 100 指数），具体如下：

行业评级

增持：预计行业股票指数超越基准
中性：预计行业股票指数基本与基准持平
减持：预计行业股票指数明显弱于基准

公司评级

买入：预计股价超越基准 15%以上
增持：预计股价超越基准 5%~15%
持有：预计股价相对基准波动在-15%~5%之间
卖出：预计股价弱于基准 15%以上
暂停评级：已暂停评级、目标价及预测，以遵守适用法规及/或公司政策
无评级：股票不在常规研究覆盖范围内。投资者不应期待华泰提供该等证券及/或公司相关的持续或补充信息

法律实体披露

中国：华泰证券股份有限公司具有中国证监会核准的“证券投资咨询”业务资格，经营许可证编号为：91320000704041011J

香港：华泰金融控股（香港）有限公司具有香港证监会核准的“就证券提供意见”业务资格，经营许可证编号为：AOK809

美国：华泰证券（美国）有限公司为美国金融业监管局（FINRA）成员，具有在美国开展经纪交易商业务的资格，经营业务许可编号为：CRD#:298809/SEC#:8-70231

新加坡：华泰证券（新加坡）有限公司具有新加坡金融管理局颁发的资本市场服务许可证，并且是豁免财务顾问。公司注册号：202233398E

华泰证券股份有限公司

南京
南京市建邺区江东中路 228 号华泰证券广场 1 号楼/邮政编码：210019

电话：86 25 83389999/传真：86 25 83387521
电子邮件：ht-rd@htsc.com

北京
北京市西城区太平桥大街丰盛胡同 28 号太平洋保险大厦 A 座 18 层/邮政编码：100032
电话：86 10 63211166/传真：86 10 63211275
电子邮件：ht-rd@htsc.com

深圳
深圳市福田区益田路 5999 号基金大厦 10 楼/邮政编码：518017
电话：86 755 82493932/传真：86 755 82492062
电子邮件：ht-rd@htsc.com

上海
上海市浦东新区东方路 18 号保利广场 E 栋 23 楼/邮政编码：200120
电话：86 21 28972098/传真：86 21 28972068
电子邮件：ht-rd@htsc.com

华泰金融控股（香港）有限公司
香港中环皇后大道中 99 号中环中心 53 楼
电话：+852-3658-6000/传真：+852-2567-6123
电子邮件：research@htsc.com
http://www.htsc.com.hk

华泰证券（美国）有限公司
美国纽约公园大道 280 号 21 楼东（纽约 10017）
电话：+212-763-8160/传真：+917-725-9702
电子邮件: Huatai@htsc-us.com
http://www.htsc-us.com

华泰证券（新加坡）有限公司
滨海湾金融中心 1 号大厦, #08-02, 新加坡 018981
电话：+65 68603600
传真：+65 65091183