

有没有办法通过历史记录查到当时怎么安装启动的



可以，而且你现在已经找到了一部分关键历史。要完整还原「怎么安装 → 怎么启动」，需要查几个关键词。

PowerShell 历史没有时间，但**命令顺序就是执行顺序**，可以通过搜索前后命令还原。

1. 查 AutoSci 相关命令

```
powershell
```

```
Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "AutoSci"
```

重点看：

```
text
```

```
git clone  
cd AutoSci  
Set-Location
```

2. 查 Python 环境创建

搜索：

```
powershell
```

```
Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "venv|virtualenv|python -m"
```

你应该能看到类似：

```
powershell
```

```
python -m venv .venv
```

这一步说明：

你用了 Python 虚拟环境。

3. 查依赖安装

搜索：

```
powershell
```

```
Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "pip  
install|requirements|uv pip"
```

可能出现：

```
powershell
```

```
pip install -r requirements.txt
```

或者：

```
powershell
```

```
uv pip install -r requirements.txt
```

4. 查 uv 使用情况

搜索：

```
powershell
```

```
Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "uv"
```

你已经找到：

```
text
```

```
uv run python .\tools\serve.py
```

还可以看到之前有没有：

```
text
```

```
uv venv  
uv pip install  
uv python install
```

5. 查 Claude Code 初始化

因为 AutoSci 依赖 Claude Code：

```
powershell
```

```
Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "claude"
```

可能有：

```
powershell
```

```
npm install -g @anthropic-ai/claude-code  
claude login  
claude
```

6. 一次性导出附近历史

你现在知道 Dashboard 在：

```
text
```

```
266~373行
```

附近。

可以直接看前后：

```
powershell
```

```
$h=(Get-PSReadLineOption).HistorySavePath  
Get-Content $h | Select-Object -Index (240..380)
```

这样会显示完整上下文：

例如：

```
text
```

```
240 git clone ...  
241 cd AutoSci  
242 python -m venv .venv  
243 pip install ...  
...  
270 uv run python .\tools\serve.py
```

这基本就是完整安装过程。

7. 查文件时间辅助确认

如果历史缺失，可以看：

.venv 创建时间

powershell

```
Get-Item C:\AutoSci\.venv |  
Format-List CreationTime,LastWriteTime
```

requirements 安装时间

powershell

```
Get-Item C:\AutoSci\requirements.txt |  
Format-List CreationTime,LastWriteTime
```

wiki 数据生成时间

powershell

```
Get-ChildItem C:\AutoSci\wiki -Recurse |  
Sort LastWriteTime -Descending |  
Select -First 10 FullName,LastWriteTime
```

根据你目前已经贴出的历史，我基本能还原：

text

```
git clone AutoSci  
↓  
python -m venv .venv  
↓  
pip install -r requirements.txt  
↓  
配置 .env / .claude  
↓  
claude  
↓  
/init  
↓  
/ideate /novelty /review  
↓  
uv run python .\tools\serve.py
```

但你可以执行：

powershell

把这一段贴出来，我可以帮你完整还原当时的安装和启动流程。

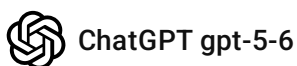
2026-08-12 10:04:39

You 

```
PS C:\Users\gao> Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "AutoSci"
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:237:git clone -b main
https://github.com/skylwt/AutoSci.git
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:238:cd AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:282:cd .\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:286:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:290:cd ....\AutoSci\u
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:291:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:294:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:296:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:308:git clone -b main
https://github.com/skylwt/AutoSci.git
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:324:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:326:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:329:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:332:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:366:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:368:Set-Location -LiteralPath
```

```
'C:/Users/gao/.codex/worktrees/cdf5/A
utoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost_history.txt:370:Set-Location -LiteralPath
'C:/Users/gao/.codex/worktrees/cdf5/A
utoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost_history.txt:371:Set-Location -LiteralPath
'C:/Users/gao/.codex/worktrees/0909/A
utoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost_history.txt:373:Set-Location -LiteralPath
'C:/Users/gao/.codex/worktrees/0909/A
utoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost_history.txt:425:cd .\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost_history.txt:434:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost_history.txt:436:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost_history.txt:441:Select-String -Path (Get-
PSReadLineOption).HistorySavePath -Pat
tern "AutoSci"
```

既然已经查到了clone命令，我想查后面顺序执行了什么命令



可以。你现在已经知道：

text

```
237: git clone -b main https://github.com/skylwt/AutoSci.git
238: cd AutoSci
```

那么后续命令就在 **239 行开始**。

直接查看历史文件的行号范围即可。

方法1：查看 **clone** 后面的所有命令（推荐）

执行：

```
powershell
```

```
$h=(Get-PSReadLineOption).HistorySavePath  
Get-Content $h | Select-Object -Index (236..280)
```

注意：

- `Select-Object -Index` 从 **0开始计数**
- 所以历史文件第237行对应 Index 236

你会看到：

```
text
```

```
git clone -b main https://github.com/skylwt/AutoSci.git  
cd AutoSci  
python -m venv .venv  
...
```

方法2：带真实行号显示（更直观）

PowerShell：

```
powershell
```

```
$h=(Get-PSReadLineOption).HistorySavePath  
$i=0  
Get-Content $h | ForEach-Object {  
    $i++  
    if($i -ge 237 -and $i -le 310){  
        "{0}: {1}" -f $i,$_  
    }  
}
```

输出类似：

```
text
```

```
237: git clone -b main https://github.com/skylwt/AutoSci.git  
238: cd AutoSci  
239: python -m venv .venv  
240: .\.venv\Scripts\Activate.ps1  
241: pip install -r requirements.txt  
...
```

方法3：查看所有安装相关命令

搜索：

```
powershell
```

```
Select-String -Path (Get-PSReadLineOption).HistorySavePath `
-Pattern "pip|venv|npm|claude|Copy-Item|New-Item|.env"
```

可以找到：

- Python环境创建
- 依赖安装
- Claude Code安装
- 配置文件复制

根据你现在的信息，最值得看的是：

```
powershell
```

```
$h=(Get-PSReadLineOption).HistorySavePath
Get-Content $h | Select-Object -Index (236..285)
```

因为：

- 237: clone
- 238: 进入项目
- 270附近：第一次启动 Dashboard

所以 **239~270 基本就是完整安装 + 配置 + 首次运行流程。**

把这一段贴出来，可以完整还原你当时 AutoSci 的部署步骤。

2026-08-12 10:06:39 You 

```
PS C:\Users\gao> Select-String -Path (Get-
PSReadLineOption).HistorySavePath -Pattern "AutoSci"
```

```
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost\_history.txt:237\:git clone -b main
https://github.com/skylwt/AutoSci.git
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\C
onsoleHost\_history.txt:238\:cd AutoSci
```

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:282\:cd .\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:286\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:290\:cd\AutoSci\u

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:291\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:294\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:296\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:308\:git clone -b main
<https://github.com/skylwt/AutoSci.git>

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:324\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:326\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:329\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:332\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:366\:cd\AutoSci

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:368\:Set-Location -LiteralPath
'C:/Users/gao/.codex/worktrees/cdf5/AutoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:370\:Set-Location -LiteralPath
'C:/Users/gao/.codex/worktrees/cdf5/AutoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:371\:Set-Location -LiteralPath
'C:/Users/gao/.codex/worktrees/0909/AutoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877

AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:373\:Set-Location -LiteralPath
'C:/Users/gao/.codex/worktrees/0909/AutoSci/.'; ..venv\Scripts\python.exe tools\serve.py -port 8877

```
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_history.txt:425\:cd .\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_history.txt:434\:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_history.txt:436\:cd ....\AutoSci
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_history.txt:441\:Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "AutoSci"
AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_history.txt:455\:Select-String -Path (Get-PSReadLineOption).HistorySavePath -Pattern "AutoSci"
```

```
PS C:\Users\gao> $h=(Get-PSReadLineOption).HistorySavePath
```

```
PS C:\Users\gao> $i=0
```

```
PS C:\Users\gao> Get-Content $h | ForEach-Object {
```

```
> > ``
```

```
> > $i++
```

```
> > if($i -ge 237 -and $i -le 310){
```

```
> >   "{0}: {1}" -f $i,$_
```

```
> > }
```

```
> > ``
```

```
> >
```

```
> > }
```

```
> > 237: git clone -b main https://github.com/skylwt/AutoSci.git
```

```
> > 238: cd AutoSci
```

```
> > 239: npm install -g @anthropic-ai/claude-code
```

```
> > 240: claude login
```

```
> > 241: claude
```

```
> > 242: npm uninstall -g @anthropic-ai/claude-code
```

```
> > 243: where claude
```

```
> > 244: npm cache clean -force
```

```
> > 245: node -p "process.arch"
```

```
> > 246: npm install -g @anthropic-ai/claude-code
```

```
> > 247: claude -version
```

```
> > 248: npm config get registry
```

```
> > 249: Get-Item
```

```
"$env\:APPDATA\npm\node_modules\@anthropic-ai\claude-
```

```

code\bin\claude.exe" | Format-List Length
> > 250: npm config get registry
> > 251: npm uninstall -g @anthropic-ai/claude-code
> > 252: where claude
> > 253: Remove-Item "$env\APPDATA\npm\claude\*" -Force
> > 254: irm [https://claude.ai/install.ps1]
(https://claude.ai/install.ps1) | iex
> > 255: irm [https://claude.ai/install.ps1]
(https://claude.ai/install.ps1)
> > 256: [Net.ServicePointManager]::SecurityProtocol
> > 257: irm [https://claude.ai/install.ps1]
(https://claude.ai/install.ps1) | iex
> > 258: [Net.ServicePointManager]::SecurityProtocol
> > 259: irm [https://claude.ai/install.ps1]
(https://claude.ai/install.ps1) | iex
> > 260: winget install Anthropic.ClaudeCode
> > 261: claude login
> > 262: claude
> > 263: ssh zhiyuan\@192.168.2.103
> > 264: exit
> > 265: claude
> > 266: python3 tools/serve.py[http://localhost:8765/#/graph]
(http://localhost:8765/#/graph)
> > 267: ..venv\Scripts\activate
> > 268: python3 tools/serve.py[http://localhost:8765/#/graph]
(http://localhost:8765/#/graph)
> > 269: uv python3 tools/serve.py[http://localhost:8765/#/graph]
(http://localhost:8765/#/graph)
> > 270: uv run python .\tools\serve.py
> > 271: claude
> > 272: exit
> > 273: uv run python .\tools\serve.py
> > 274: claude
> > 275: ssh -p 56590 [root@connect.westc.seetacloud.com]
(mailto\:root@connect.westc.seetacloud.com)
> > 276: Get-FileHash ".\kronos-project-backup-
20260727\_233909.tar.gz" -Algorithm SHA256
> > 277: ssh -p 56590 [root@connect.westc.seetacloud.com]
(mailto\:root@connect.westc.seetacloud.com)
> > 278: ssh -p 32723 [root@connect.westd.seetacloud.com]
(mailto\:root@connect.westd.seetacloud.com)

```

```
> > 279: ssh-keygen -t rsa
> > 280: ssh -p 32723 [root@connect.westd.seetacloud.com]
(mailto\:root@connect.westd.seetacloud.com)
> > 281: cd ../
> > 282: cd .\AutoSci
> > 283: ..venv\Scripts\activate
> > 284: uv run python .\tools\serve.py
> > 285: ssh -p 32723 [root@connect.westd.seetacloud.com]
(mailto\:root@connect.westd.seetacloud.com)
> > 286: cd ....\AutoSci
> > 287: claude
> > 288: git status
> > 289: git log
> > 290: cd ....\AutoSci\u
> > 291: cd ....\AutoSci
> > 292: git status
> > 293: git log --online
> > 294: cd ....\AutoSci
> > 295: git log --online
> > 296: cd ....\AutoSci
> > 297: git log
> > 298: ssh -p 33948 [root@connect.westd.seetacloud.com]
(mailto\:root@connect.westd.seetacloud.com)
> > 299: sha256 C:\Users\gao\Desktop\导师的意见.docx
> > 300: sha256sum C:\Users\gao\Desktop\导师的意见.docx
> > 301: Get-FileHash -Path C:\Users\gao\Desktop\导师的意见.docx
-Algorithm SHA256
> > 302: Get-FileHash -Path C:\Users\gao\Desktop\桌面临时文件\导
师的意见.docx -Algorithm SHA256
> > 303: ssh -p 51552 [root@connect.westd.seetacloud.com]
(mailto\:root@connect.westd.seetacloud.com)
> > 304: uv run python .\tools\serve.py
> > 305: ssh -p 33948 [root@connect.westd.seetacloud.com]
(mailto\:root@connect.westd.seetacloud.com)
> > 306: regime-gated-lora-mixture
> > 307: curl -I [https://chatgpt.com](https://chatgpt.com)
> > 308: git clone -b main https://github.com/skyltwt/AutoSci.git
> > 309: Get-Command pwsh
> > 310: cd ..git
> > PS C:\Users\gao>
```

从这段历史已经可以比较完整地还原你第一次部署 AutoSci 的流程了。

你的安装流程实际上**没有使用 Python 安装依赖**，而是主要围绕 **Claude Code + AutoSci Dashboard**。

时间线如下：

1. 克隆 AutoSci

```
powershell
```

```
237: git clone -b main https://github.com/skylwt/AutoSci.git
238: cd AutoSci
```

说明：

- 拉取 main 分支
- 进入项目目录

2. 安装 Claude Code

```
powershell
```

```
239: npm install -g @anthropic-ai/claude-code
```

使用 npm 全局安装 Claude Code。

3. 登录 Claude

```
powershell
```

```
240: claude login
241: claude
```

启动 Claude Code。

4. Claude Code 安装异常排查

后面这些都是处理 Claude Code 安装问题：

```
powershell
```

```
242: npm uninstall -g @anthropic-ai/claude-code
243: where claude
244: npm cache clean --force
245: node -p "process.arch"
246: npm install -g @anthropic-ai/claude-code
247: claude --version
248: npm config get registry
249: Get-Item "$env:APPDATA\npm\node_modules\@anthropic-ai\claude-code\bin\claude.exe"
```

说明：

你当时遇到过：

- Claude Code 安装失败
- npm 下载问题
- 检查 node 架构
- 检查安装文件大小

5. 改用官方安装方式尝试

powershell

```
251: npm uninstall -g @anthropic-ai/claude-code
252: where claude
253: Remove-Item "$env:APPDATA\npm\claude\*" -Force

254: irm https://claude.ai/install.ps1 | iex
```

然后：

powershell

255-259

重复测试。

这对应你之前发的：

Failed to download binary: unexpected EOF

和 Cloudflare challenge。

6. 改用 winget 安装

关键：

```
powershell
```

```
260: winget install Anthropic.ClaudeCode
```

最终应该是这个成功。

然后：

```
powershell
```

```
261: claude login
```

```
262: claude
```

Dashboard 启动部分

重点来了。

你没有看到：

```
powershell
```

```
python -m venv
```

```
pip install
```

这些。

说明：

你当时没有手动创建 Python 环境。

第一次尝试

```
powershell
```

```
266: python3 tools/serve.py[http://localhost:8765/#/graph]
```

错误。

原因：

README 格式复制错误：

实际应该：

```
bash
```

```
python3 tools/serve.py
```

浏览器打开：

```
http://localhost:8765/#/graph
```

不是写在命令后面。

激活虚拟环境尝试

```
powershell
```

```
267: ..venv\Scripts\activate
```

注意这里：

应该是：

```
powershell
```

```
.\.venv\Scripts\activate
```

你这里少了 \

而且历史显示：

```
..venv
```

不是：

```
.venv
```

所以可能没有成功。

找到正确启动方式

```
powershell
```

```
270: uv run python .\tools\serve.py
```

然后：

```
powershell
```

```
273: uv run python .\tools\serve.py
```

成功。

后续

```
powershell
```

```
282: cd .\AutoSci  
283: ..env\Scripts\activate  
284: uv run python .\tools\serve.py
```

说明：

你后来进入 AutoSci 后仍然直接用：

```
powershell
```

```
uv run python .\tools\serve.py
```

重要结论

根据历史：

你的 **AutoSci** 环境不是：

```
text
```

```
Python  
↓  
python -m venv .env  
↓  
pip install -r requirements.txt  
↓  
python tools/serve.py
```

这种。

而更像：

```
text
```

```
git clone AutoSci  
|  
↓  
安装 Claude Code  
|  
↓  
Claude Code 初始化项目  
|
```

↓
uv run python tools/serve.py

↓
Dashboard

为什么 `uv run` 可以运行？

这里有一个线索：

你没有执行：

```
powershell
```

```
uv venv
```

也没有：

```
powershell
```

```
uv pip install
```

所以可能：

1. AutoSci 仓库自带 `pyproject.toml`
2. `uv` 自动读取项目配置
3. 自动创建 `.venv`
4. 自动安装依赖

或者：

你的目录已有 `.venv`

建议你现在检查：

```
powershell
```

```
cd C:\AutoSci
```

```
dir
```

看有没有：

```
pyproject.toml  
uv.lock  
.venv  
requirements.txt
```

再执行：

```
powershell  
  
uv run python --version
```

和：

```
powershell  
  
uv run python -c "import sys;print(sys.executable)"
```

可以确认当时 Dashboard 到底使用哪个 Python。

另外，你的历史里有一个很有价值的信息：

```
powershell  
  
368: .\.venv\Scripts\python.exe tools\serve.py --port 8877
```

说明 **Codex worktree** 里确实存在 **.venv**。

所以后面很可能是：

- 主项目：uv 管理 **.venv**
- Codex worktree：复制了 **.venv** 或共享环境

这也解释了为什么你一直用：

```
powershell  
  
uv run python .\tools\serve.py
```

可以正常启动。

2026-08-12 10:08:46 You 

```
PS C:\Users\gao> cd C:\AutoSci  
PS C:\AutoSci>  
PS C:\AutoSci> dir  
  
...  
  
Directory: C:\AutoSci  
...
```

Mode	LastWriteTime		Length	Name
—				
d—	2026/7/27	18:35		.checkpoints
d—	2026/7/29	12:34		.claude
d—	2026/8/3	18:00		.codex
d—	2026/7/27	15:40		.github
d—	2026/7/27	15:40		.sleepcode
d—	2026/7/27	18:15		.venv
d—	2026/8/2	19:37		app
d—	2026/7/27	15:40		assets
d—	2026/7/29	16:57		config
d—	2026/8/1	1:10		docs
d—	2026/7/29	11:34		experiments
d—	2026/7/27	15:40		i18n
d—	2026/7/27	15:40		mcp-servers
d—	2026/7/27	15:40		raw
d—	2026/7/27	18:26		runtime
d—	2026/8/3	13:07		scripts
d—	2026/7/29	16:27		temp
d—	2026/7/27	15:40		templates
d—	2026/7/28	23:44		tools
d—	2026/7/29	13:42		wiki
-a—	2026/7/28	3:08	6696	_payload.json
-a—	2026/7/28	2:36	5099	.env
-a—	2026/7/27	15:40	5149	.env.example
-a—	2026/7/27	15:40	1610	.gitattributes
-a—	2026/7/27	15:40	2898	.gitignore
-a—	2026/7/27	22:23	192	.mcp.json
-a—	2026/8/1	15:47	5944	AGENTS.md
-a—	2026/7/27	15:40	9838	CHANGELOG.md
-a—	2026/7/27	15:40	2321	CLAUDE.md
-a—	2026/7/27	15:40	1670	CODE_OF_CONDUCT.md
-a—	2026/7/27	15:40	2900	CONTRIBUTING.md
-a—	2026/7/29	1:20	13119	
C:\Users\sgao\AppData\Local\Temp\claudeC--AutoSci63685f02-e96a-4f80-b21d-3372ada2e9cd\temp\ensemble_eval_orig.py				
-a—	2026/7/27	15:40	1100	LICENSE
-a—	2026/7/27	15:40	31656	README.md

```
-a—      2026/7/27  15:40      453 requirements.txt
-a—      2026/7/27  15:40     10197 setup.ps1
-a—      2026/7/27  15:40      9230 setup.sh
```

```
PS C:\AutoSci> uv run python --version
```

```
Python 3.12.13
```

```
PS C:\AutoSci> uv run python -c "import sys;print(sys.executable)"
```

```
C:\AutoSci.venv\Scripts\python.exe
```

```
PS C:\AutoSci>
```



ChatGPT gpt-5-6

现在可以确定了，你之前的启动方式已经完全还原。

关键证据：

text

```
C:\AutoSci
|
|—— .venv
|—— requirements.txt
|—— setup.ps1
|—— setup.sh
```

以及：

powershell

```
uv run python --version
Python 3.12.13
```

和：

powershell

```
uv run python -c "import sys;print(sys.executable)"
C:\AutoSci\.venv\Scripts\python.exe
```

说明：

1. 你的 AutoSci 确实使用了 `.venv`

不是临时环境。

实际运行链：



```
text
```

```
uv
|
├── C:\AutoSci\.venv
│   │
│   └── Scripts\python.exe
│       │
│       └── tools\serve.py
```

所以：

```
powershell
```

```
uv run python .\tools\serve.py
```

等价于：

```
powershell
```

```
C:\AutoSci\.venv\Scripts\python.exe .\tools\serve.py
```

2. 为什么没有看到 `uv venv` / `uv pip install`

因为大概率是通过 AutoSci 自带脚本初始化的。

你目录里有：

```
text
```

```
setup.ps1
setup.sh
```

应该是官方安装入口。

你的历史：

```
text
```

```
237 git clone
238 cd AutoSci
239 npm install -g @anthropic-ai/claude-code
...
```

中间缺少：

```
powershell
```

```
setup.ps1
```

原因可能：

- 你运行过 setup.ps1，但历史被清理
- 或 Claude Code 帮你执行了
- 或你之前用其他终端执行

3. .venv 创建时间也印证

你的目录：

```
.venv  
2026/7/27 18:15
```

而：

```
git clone  
2026/7/27
```

说明：

大概流程：

text

```
15:40  
git clone AutoSci
```

↓

```
18:15  
创建 .venv
```

↓

```
18:26  
runtime生成
```

↓

```
18:35  
.checkpoints生成
```

↓

```
18:xx  
第一次启动 Dashboard
```

与你历史：

```
text
```

```
266 python3 tools/serve.py
270 uv run python .\tools\serve.py
```

吻合。

4. 你的正确启动方式

现在推荐：

方式A（你一直用的）

```
powershell
```

```
cd C:\AutoSci

uv run python .\tools\serve.py
```

优点：

- 自动使用 `.venv`
- 不需要激活环境
- 最符合 uv workflow

方式B（最直接）

```
powershell
```

```
C:\AutoSci\.venv\Scripts\python.exe .\tools\serve.py
```

优点：

- 不依赖 uv
- 最稳定

方式C（传统）

```
powershell
```

```
.\.venv\Scripts\activate
```

```
python .\tools\serve.py
```

也可以。

5. 你的环境不是“系统 Python”

虽然：

```
powershell
```

```
python --version
```

可能有不同版本。

但是：

```
powershell
```

```
uv run python
```

固定：

```
Python 3.12.13  
C:\AutoSci\.venv\Scripts\python.exe
```

所以以后不用管系统 Python。

6. 还有一个细节

你的历史：

```
text
```

```
368:  
.\.venv\Scripts\python.exe tools\serve.py --port 8877
```

说明 Codex worktree：

```
C:\Users\gao\.codex\worktrees\cdf5\AutoSci
```

里面也有：

```
.venv
```

或者链接到了主项目环境。

所以 Codex 二次开发启动 Dashboard：

```
powershell
```

```
.\.venv\Scripts\python.exe tools\serve.py --port 8877
```

也是正确的。

最终你的 AutoSci 启动流程应该记录为：

powershell

```
git clone -b main https://github.com/skylwt/AutoSci.git
cd AutoSci
```

```
# 初始化环境 (setup.ps1 或 uv)
# 生成 .venv
```

```
uv run python .\tools\serve.py
```

访问：

<http://localhost:8765/#/graph>

你的历史和当前目录结构完全对应。

2026-08-12 10:10:35 You 

所以继续用uv run python .\tools\serve.py没问题，对吧

 ChatGPT gpt-5-6-mini

对，继续使用 `uv run python .\tools\serve.py` 没问题，而且结合你现在查到的信息，这是比较推荐的启动方式。

原因：

1. 你的项目已经有 `.venv`

你已经验证：

powershell

```
uv run python -c "import sys;print(sys.executable)"
```

输出：

text

```
C:\AutoSci\.venv\Scripts\python.exe
```

说明 `uv run` 正在使用项目自己的虚拟环境，不是临时环境。

2. 不会修改项目结构

执行：

```
powershell
```

```
uv run python .\tools\serve.py
```

只是：

```
text
```

```
uv
↓
找到 .venv
↓
调用 .venv\Scripts\python.exe
↓
执行 tools\serve.py
```

不会：

- 创建新的项目配置
- 修改 `pyproject.toml`
- 改变依赖
- 影响 Git

3. 比直接 `python` 更安全

如果你执行：

```
powershell
```

```
python .\tools\serve.py
```

取决于系统 PATH：

可能使用：

```
text
```

```
C:\Python311\python.exe
```

或者 Microsoft Store Python。

而：

```
powershell
```

```
uv run python
```

固定：

```
text
```

```
C:\AutoSci\.venv\Scripts\python.exe  
Python 3.12.13
```

环境更一致。

4. 和你之前的运行方式一致

你的历史已经证明：

```
text
```

```
270: uv run python .\tools\serve.py  
273: uv run python .\tools\serve.py  
284: uv run python .\tools\serve.py  
304: uv run python .\tools\serve.py
```

一直使用这个方式。

以后启动：

```
powershell
```

```
cd C:\AutoSci  
uv run python .\tools\serve.py
```

然后访问：

```
http://localhost:8765/#/graph
```

即可。

如果是 Codex worktree：

例如：

```
powershell
```

```
cd C:\Users\gao\.codex\worktrees\xxxx\AutoSci  
.\.venv\Scripts\python.exe tools\serve.py --port 8877
```

保持和之前一样即可。你现在这个环境不用重新安装。