LLM选型策略报告：面向Obsidian自动化Agent的零成本高效框架构建

I. 执行摘要与战略背景

I.A. 项目概述：零成本第二大脑Agentic框架

用户的核心目标是建立一个结合Obsidian和Agent的“第二大脑”，并最终实现“全流程自动化”与“稳定地高效率”，同时将成本控制在“0或低成本”范围。这不仅仅是对大型语言模型（LLM）的文本生成能力要求，更是对Agentic AI系统的复杂功能需求。Agentic AI系统必须能够执行动态的、目标驱动的动作，具备持久的记忆能力，并能够熟练地使用外部工具，真正成为用户的“手和脚” [1]。

为了实现这一目标，LLM的选择必须聚焦于几个关键技术维度：功能调用/工具使用准确性，这是实现“手和脚”的基础；快速响应（低延迟），以保证“稳定地高效率”；以及长上下文处理能力，用于对Obsidian知识库进行深入的检索增强生成（RAG）。Agent系统通过将复杂的任务分解为多个子任务，并分配给具有专业技能的代理，能够显著消除流程切换时间，从而提高整体效率和输出质量 [3]。因此，所选模型必须擅长规划、推理和动态适应 [2]。

I.B. 架构蓝图：Obsidian、Agent与VSCode集成

实现“全流程自动化”的关键在于LLM与Obsidian及VSCode等外部系统的桥梁。这项战略的核心在于LLM必须能够生成结构化的、机器可读的输出，而不仅仅是自由文本。

功能调用（Function Calling）正是实现这一目标的核心技术。它允许LLM根据用户的自然语言输入，解释其意图，并生成一个结构化的JSON对象，描述需要调用的外部函数名称和所需参数 [5]。例如，LLM不会直接操作Obsidian，而是生成一个指令，告诉外部执行器（如LangChain、

Qwen-Agent或用户定制的Python/VSCode脚本）去执行特定的操作，例如检索特定的笔记、创建新文件或运行VSCode中的特定脚本 [7]。

此外，为了构建一个真正的“第二大脑”，系统需要增强其记忆和外部知识基础。检索增强生成（RAG）是利用Obsidian知识库的关键手段 [9]。通过将LLM与知识图谱（Knowledge Graphs, KG）结合，可以进一步提升RAG的性能，特别是将原始笔记转换为结构化的KG，这不仅能够以结构化方式存储不断更新的事实信息，还能在个性化响应生成中增强模型的推理能力、提高准确性并减少幻觉 [10]。通过集成先进的LLM优化技术和微调工作流程，可以实现高效且可扩展的知识图谱生成，为企业级AI应用提供强大的基础 [11]。

在一个完全自动化的流程中，延迟的累积效应会大幅增加实际运行成本。Agentic工作流依赖于迭代的步骤，如ReAct（规划、行动、观察、反思）。如果基础模型的平均延迟较高，例如Claude 3.5 Sonnet比GPT-4o慢24% [12]，这种延迟将在多步工作流中不断累积。因此，对追求“快速响应”和高效率的低成本方案而言，选择具有卓越首个令牌生成时间（TTFT）的模型至关重要，因为低延迟直接转化为更少的实际等待时间，从而实现更高的效率 [13]。

此外，从长期成本控制的角度来看，工具调用准确性是决定系统稳定性的首要因素。如果LLM在生成工具调用参数时出现错误或“幻觉”，将迫使Agent框架进入昂贵的重新规划循环，消耗不必要的令牌和时间 [14]。因此，为了确保项目长期稳定的高效率运行，对LLM而言，

工具调用准确性是比基础令牌价格差异更具战略性的成本控制因素。

II. 核心选择标准与维度分析

LLM的战略选择必须平衡三个核心约束：成本（Cost, C）、速度/稳定性（Speed/Stability, S）和能力（Capability, K）。

II.A. Agent系统性能指标：速度与准确性

1. 延迟（TTFT）与响应速度：
   在Agentic工作流中，模型的响应速度是衡量效率的关键。例如，GPT-4o在延迟方面表现卓越，平均延迟比Claude 3.5 Sonnet快24%，并且首个令牌生成时间（TTFT）快2倍 12。这种高速度使得GPT-4o成为需要快速迭代的实时Agent执行的当前行业标杆。对于要求“快速响应”的Agent循环，模型必须经过优化以实现快速TTFT，因为Agent的成功更多依赖于快速、结构化的工具执行，而不是单纯的文本生成 13。
2. 工具调用准确性与结构化输出：
   功能调用能力是实现Agent“手和脚”的基石。OpenAI、Gemini和Anthropic等国际领先模型

都原生支持这一功能 6。值得注意的是，在国内和开源模型中，
Qwen家族的模型在工具调用方面领先于其他开源选项，即使是较小的8B版本也表现优异 [16]。此外，DeepSeek-R1专注于推理和问题解决能力 [17]，这使其在需要复杂结构化输出生成（如从文本中提取知识图谱三元组）的任务中成为有力竞争者。

II.B. 成本与上下文维度分析

1. 长上下文需求（长文本）：
   “第二大脑”的核心需求之一是处理长文档和执行RAG。目前行业领先的长上下文能力显著，例如Google Gemini 1.5 Pro/Flash提供100万甚至更高的Token上下文窗口 9。xAI的Grok 4 Fast也提供了200万Token的上下文窗口 19。国内的Qwen 2.5 Pro也支持100万Token的上下文 21。然而，经验证据表明，许多LLM在处理超过64K Token的上下文时，准确性的一致性难以维持 9。因此，在利用长上下文时，必须结合RAG和知识图谱技术 10，确保关键信息被高效检索，而不是仅仅依赖模型自身的长记忆能力 15。
2. 零/低成本范围分析：
   为了实现“0或低成本”的目标，可以采用多种策略：
   - 免费**API**层级（**Gemini/OpenAI**）：Gemini提供免费层级（Free of charge）[22]。然而，这些免费层级受到严格的请求限制，例如Flash/Lite模型每天的请求次数（RPD）上限为500次 [22]，并且可能有未公开的Token使用上限（例如，针对某些Pro模型的每日6M输入Token限制）[24]。OpenAI的免费层级有每月100美元的使用上限 [25]。
   - 低成本付费**API**（**Qwen/DeepSeek**）：国内模型提供了极具竞争力的价格。例如，混元（Hunyuan）定价极具性价比（每百万输入Token仅0.8元）[21]。DeepSeek-Chat的价格也极低（约$0.56/百万输入Token）[26]。
   - 自托管经济学（**0**成本）：通过在本地部署高效模型（如Qwen3-8B），使用Ollama或vLLM等工具 [27]，可以消除所有外部API成本和速率限制，实现真正的“0成本”稳定性（在不考虑硬件折旧的情况下）。例如，租用消费级GPU（如RTX 4090）的成本约为每小时$0.34，与高频API使用费相比微不足道 [29]。

对于稳定自动化而言，硬性的每日请求限制（RPD）是致命的。例如，Gemini免费层的500 RPD上限 [22]意味着一旦达到限制，整个自动化工作流将停滞至第二天。因此，战略上必须避免依赖这些硬性限制，转而采用自托管方案（无限RPD）或选择官方声明不限制请求速率的低成本付费API（如DeepSeek，但需注意可能出现的服务器拥堵导致的限流）[30]。

通义千问（Qwen）家族，特别是DeepSeek-R1，在量化（Quantization）方面表现出强大的韧性。这意味着即使将模型压缩到更小的格式（如Q4_K_M）以适应消费级硬件，它们也能保持高准确性 [32]。这种量化弹性对于实现零成本自托管策略至关重要。用户可以在较低配置的硬件上运行一个功能强大、工具调用准确性高（Qwen3-8B）的Agent模型 [16]，极大地降低了实现“0或低成本”目标的门槛，并确保了Agentic精度不因压缩而大幅下降。

III. 国际平台综合模型比较（高速度/长上下文）

III.A. OpenAI (ChatGPT)：GPT-4o重点分析

GPT-4o是当前国际市场上速度最快的LLM之一，其延迟低、TTFT快，使其在需要快速迭代的Agent任务中表现出色 [12]。它拥有成熟的128K上下文窗口和业界最完善的功能调用生态系统 [6]。然而，GPT-4o本质上是一个付费模型。虽然OpenAI提供了免费层级（每月$100使用限额） [25]，但要获得GPT-4o所需的稳定性和高速度，必须升级至付费层级，以获得更高的速率限制（例如，付费Tier 1账户通常获得比免费层高得多的RPM和TPM） [25]。因此，GPT-4o不适合作为零成本的主要执行引擎。它的战略定位应是作为

高品质的后备**Agent**大脑，或用于对速度和数学推理（76.6%的MATH基准分数）等要求极高的非日常任务 [12]。

III.B. Google (Gemini)：长上下文与免费层挑战

Gemini 1.5 Flash以其100万Token的超长上下文窗口，直接满足了用户对PKM RAG的“长文本”要求 [18]。Gemini 2.5 Flash-Lite则提供了极具成本效益的付费和免费API访问 [22]。

然而，免费层的使用具有严格的限制和潜在的风险。除了免费使用，用户的数据可能会被用于改进Google的产品（付费层则不会） [22]。更关键的是，免费层存在硬性请求限制，例如Flash/Lite模型的每日请求上限（RPD）通常为500次 [22]，且某些模型可能存在约每日6M输入Token的隐性上限 [24]。这种硬性限制组合使得Gemini免费层难以用于持续的、高吞吐量的自动化工作流，因为它随时可能触发停机。

因此，Gemini 1.5 Flash的最佳战略角色是作为主要的长期上下文处理引擎，专门用于批量的RAG任务和复杂的笔记摄取，利用其1M Token的优势，但必须严格监控API调用，确保每日使用量远低于RPD上限，以维持服务连续性。

III.C. Anthropic (Claude) 与 xAI (Grok)

Claude 3.5 Sonnet在研究生级别的推理能力（GPQA 59.4%）和复杂的文本/代码处理方面表现出色 [12]。但它在延迟方面落后于GPT-4o [12]，这在需要快速迭代的Agent循环中构成了效率障碍。其成本结构也不符合零成本战略。

xAI的Grok 4 Fast的突出特点是其巨大的200万Token上下文窗口 [19]。Grok 4 Fast专注于提高速度和效率，据称比早期版本使用的Token减少了40% [19]。其API定价（Fast模型每百万Token输入

0.20，输出0.50）具有竞争力 [20]。然而，Grok的功能调用性能虽受支持，但相较于GPT、Gemini和Qwen等已建立的模型，其Agentic生态相对较新。

IV. 国内与开源平台综合模型比较（零成本重点）

实现“0或低成本”稳定性的最可行路径在于国内和自托管模型，它们提供了优越的成本效益，并能直接控制速率限制。

IV.A. 阿里巴巴 (通义千问 Qwen 家族)：Agentic执行器领导者

1. 工具调用优势：Qwen模型在开源模型中对工具调用的准确性表现最佳 [16]。其专用的**Qwen-Agent**库封装了工具调用模板和解析器，大大降低了Agentic工作流的开发复杂性 [7]。
2. **API**成本与上下文：Qwen提供了有竞争力的低价API服务 [35]，Qwen3-Next支持256K上下文 [7]。另外，混元的积极定价策略（0.8元/百万输入Token）也是国内市场的超低成本代表 [21]。
3. 自托管可行性（零成本方案）：较小的Qwen3-8B模型体积紧凑（约5.2GB），性能足以支持本地Agentic工作流 [27]。通过Ollama或vLLM进行本地部署，用户可以建立一个OpenAI兼容的API端点 [7]，彻底绕开外部API的速率限制和费用。通过自托管Qwen3-8B，用户可以在本地硬件条件允许下，同时满足“0成本”和“稳定高效率”的关键目标。

由于Qwen模型在工具调用方面具有已验证的高准确性 [16]，并且提供了专用的Qwen-Agent框架 [7]，这极大地简化了将“手和脚”（即Obsidian/VSCode操作）集成的开发工作。这种开发时间的缩短（用户的隐性成本）是项目整体效率的一个巨大优势。

IV.B. DeepSeek：推理能力与价值主张

DeepSeek专注于推理和问题解决[17]，使其成为处理需要逻辑思维任务的绝佳选择，例如将自然语言笔记转换为结构化的元数据或知识图谱元素。DeepSeek-R1-Distill家族在重度量化（压缩）后，仍能保持高指令遵循能力和准确性[32]，这使其非常适合在资源有限的环境中进行高效推理。在API稳定性方面，DeepSeek官方文档声明其

不限制用户速率，尽管在流量高峰期可能会出现性能下降（限流）[30]。这种不设硬性上限的策略，在稳定性上优于具有明确RPD上限的免费国际平台。

IV.C. 其他竞争者（豆包、混元）

腾讯混元以其激进的定价策略（每百万输入Token 0.8元）[21]，在预算友好的选项中占有重要地位。这些国内模型代表了高价值的低成本替代方案，可作为次要的RAG或翻译引擎。

对于核心执行器，必须认识到依赖免费API层级（如Gemini Flash）存在因RPD限制而导致的不稳定风险[22]。同时，仅依赖本地模型（如Qwen-8B）会牺牲商业API提供的1M+ Token长上下文能力。因此，最优的解决方案是采用

混合**Agent**系统：以自托管**Qwen**作为主要Agent的大脑和执行器（解决0成本、速度和稳定性问题），同时将**Gemini 1.5 Flash**的免费层仅用于需要长上下文的批量摄取任务，并严格控制其使用量，以保持在每日RPD上限之下。这种策略结合了本地部署的稳定性与商业API的卓越能力。

V. LLM评分、最终推荐方案与实施路线图

V.A. 定量评分矩阵（Agentic PKM工作流战略LLM比较）

下表根据用户的核心约束（成本、速度、上下文、Agentic能力）对主要竞争模型进行了量化比较，评分范围为1-10分。

Agentic PKM工作流定量LLM记分卡

模型 (层级)	输入成本 (每1M Tokens)	最大上下文 (Tokens)	Agentic 能力 (功能调用)	延迟/速度 (TTFT)	稳定性/速率限制控制	战略得分 (1-10)
GPT-4o (付费)	约$5.00 (高)	128k	优秀 (公认领导者)	优秀 (TTFT最快) [13]	高 (付费后，分级RPD) [33]	7 (成本过高)
Gemini 1.5 Flash (免费/付费)	$0.30 (付费) [22]	1M–2M	很好	很好	中等 (免费：硬性RPD上限) [23]	8.5 (零成本下最佳能力/上下文)
Qwen3-8B (本地/0成本)	$0 (仅硬件成本) [27]	128k [36]	优秀 (最佳开源工具) [16]	取决于硬件	优秀 (完全控制，无限RPD) [28]	**9.5 (最佳零成本稳定性)**
Deepseek-Chat (付费API)	约$0.56 (极低) [26]	131k+ [35]	很好 (推理能力强) [17]	良好	高 (无官方硬性限制，但可能限流) [30]	9.0 (最佳低成本API稳定性)
Grok 4 Fast (付费/免费)	$0.20 (低) [20]	2M	良好	很好 (注重效率) [19]	高 (付费API)	8.0 (优秀上下文，新生态)

V.B. 最终推荐方案：混合Agent系统战略

基于对成本、稳定性和效率的分析，最优的方案是采用混合Agent系统，以实现最高程度的自动化和成本控制。

1. 首选**Agent**大脑与执行器（**"手和脚"**）：**Qwen3-8B**（自托管）
   - 作用：零成本、高精度执行。
   - 理由：自托管Qwen3-8B模型通过Ollama或vLLM部署 [27]，彻底解决了外部API的费用和速率限制问题，确保了
     零成本和稳定性。Qwen在工具调用方面的领先地位 [16]保证了Agentic准确性，极大减少了失败的Agent循环，从而提高了效率。这将是所有连续、高频任务（笔记解析、例行清

理、功能调用、即时响应）的主要引擎。

2. 次要**RAG**与批量处理引擎（**"深层记忆"**）：**Gemini 1.5 Flash**（免费**API**）
   - 作用：长上下文知识检索。
   - 理由：利用免费层提供的1M Token上下文能力[18]，专门用于需要处理大篇幅研究论文或对Obsidian知识库进行深度RAG的批量任务。使用时必须严格监控API调用，确保每日请求量远低于500 RPD的上限，以避免服务中断[22]。
3. 第三方成本效益后备方案：**DeepSeek-Chat**（低成本**API**）
   - 作用：稳定的付费补充。
   - 理由：若本地硬件不足以支撑Qwen3-8B或Gemini免费层使用达到上限，DeepSeek-Chat提供了卓越的推理能力、极低的Token成本以及无硬性RPD限制的政策[26]，是付费API中的高价值选择。

V.C. 实施路线图：实现自动化与稳定性

1. **Agent**编排框架选择：采用LangChain或LlamaIndex作为整体Agent管理和工具封装的框架[8]。对于基于Qwen的本地Agent，推荐使用专用的**Qwen-Agent**库，以利用其内置的工具调用模板和解析器，加速开发[7]。
2. **Obsidian**工具开发与功能定义：定义精确的函数，例如create_note(title, content)或search_dataview(query)，并将这些函数映射到通过VSCode集成的Python脚本执行。这些功能必须在系统提示中明确描述给LLM（Model Context Protocol - MCP）[5]。
3. 稳定性与成本优化策略：
   - 任务分流：将简单的逻辑任务（如基本算术或精确事实检索）卸载给外部工具（如Python函数或Obsidian DataView查询），避免LLM处理这些低效任务，从而减少Token消耗[15]。
   - 记忆管理：实施健壮的记忆系统（例如，对历史对话进行总结、使用向量压缩记忆）[4]，以避免在每次Agent调用时重复发送巨大的上下文，从而最大化核心Agent大脑的效率和最小化成本。
   - 结构化输出强制：严格要求功能调用采用结构化的JSON输出（可通过Pydantic程序实现）[37]，以确保高工具调用准确性，减少因解析错误导致的重新提示和计算成本[5]。

结论

为了满足用户对"零或低成本"、"快速响应"、"稳定地高效率"和"长文本"的综合需求，单一模型或单一付费模式均无法完美满足所有约束。

最终的战略结论是采用**Qwen/Gemini**混合栈架构。Qwen3-8B（自托管）提供稳定、无上限、高准确度的Agent执行能力，解决了"0成本"和"手和脚"的核心需求。Gemini 1.5 Flash（免费层）则提

供1M Token的“长文本”能力，作为处理知识库深度RAG的补充。这种分层架构最大限度地提高了系统的稳定性和能力，同时将运营成本最小化。通过将Agent的核心执行器本地化，用户将运营支出转换为可控的硬件成本，保证了自动化流程的长期稳定和高效率。

引用的著作

1. AI Agent - overview - follow the idea - Obsidian Publish, 访问时间为 九月 27, 2025，https://publish.obsidian.md/followtheidea/Content/AI/AI+Agent+-+overview
2. Agentic AI vs LLMs: Key Differences in 2025 - Young Urban Project, 访问时间为 九月 27, 2025，https://www.youngurbanproject.com/agentic-ai-vs-llms/
3. LLM based Agents - concise - follow the idea - Obsidian Publish, 访问时间为 九月 27, 2025, https://publish.obsidian.md/followtheidea/Content/AI/LLM+based+Agents+-+concise
4. LLM based Agents - detailed - follow the idea - Obsidian Publish, 访问时间为 九月 27, 2025, https://publish.obsidian.md/followtheidea/Content/AI/LLM+based+Agents+-+detailed
5. Function calling using LLMs - Martin Fowler, 访问时间为 九月 27, 2025, https://martinfowler.com/articles/function-call-LLM.html
6. LLM Function Calling Explained: A Deep Dive into the Request and Response Payloads | by James Tang | Medium, 访问时间为 九月 27, 2025, https://medium.com/@jamestang/llm-function-calling-explained-a-deep-dive-into-the-request-and-response-payloads-894800fcad75
7. Qwen3-Next: Towards Ultimate Training & Inference Efficiency - Qwen AI, 访问时间为 九月 27, 2025, https://qwen.ai/blog?id=4074cca80393150c248e508aa62983f9cb7d27cd&from=research.latest-advancements-list
8. LlamaIndex vs LangChain: 7 Ultimate Side-by-Side Showdown for AI Builders - HyScaler, 访问时间为 九月 27, 2025, https://hyscaler.com/insights/llamaindex-vs-langchain-a-comparison/
9. Long Context RAG Performance of Large Language Models - arXiv, 访问时间为 九月 27, 2025，https://arxiv.org/html/2411.03538v1
10. Personalizing Large Language Models using Retrieval Augmented Generation and Knowledge Graph - arXiv, 访问时间为 九月 27, 2025, https://arxiv.org/html/2505.09945v1
11. Insights, Techniques, and Evaluation for LLM-Driven Knowledge Graphs | NVIDIA Technical Blog, 访问时间为 九月 27, 2025, https://developer.nvidia.com/blog/insights-techniques-and-evaluation-for-llm-driven-knowledge-graphs/
12. Claude 3.5 Sonnet vs GPT-4o: Complete AI Model Comparison - SentiSight.ai, 访问时间为 九月 27, 2025, https://www.sentisight.ai/claude-3-5-sonnet-vs-gpt-4o-ultimate-comparison/
13. Claude 3.5 Sonnet vs. GPT 4o: which is better? | by Hendrix | Medium, 访问时间为 九月 27, 2025,

https://medium.com/@hendrix_56915/claude-3-5-sonnet-vs-gpt-4o-which-is-better-f4c4fe3a8f16
14. Cut Costs, Not Accuracy: LLM-Powered Data Processing with Guarantees - arXiv, 访问时间为 九月 27, 2025, https://arxiv.org/html/2509.02896v1
15. 10 Ways to Make LLMs Cheaper and Faster Using Data Products, 访问时间为 九月 27, 2025, https://www.moderndata101.com/blogs/10-ways-to-make-llms-cheaper-and-faster-using-data-products
16. Local LLM Tool Calling: Which LLM Should You Use? | DockerTool Calling with Local LLMs: A Practical Evaluation | Docker, 访问时间为 九月 27, 2025, https://www.docker.com/blog/local-llm-tool-calling-a-practical-evaluation/
17. DeepSeek-R1: An Open-Source LLM Powerhouse in a Quantized World | by Frank Morales Aguilera | The Deep Hub | Medium, 访问时间为 九月 27, 2025, https://medium.com/thedeephub/deepseek-r1-an-open-source-llm-powerhouse-in-a-quantized-world-e77f4dc4df40
18. Billing | Gemini API | Google AI for Developers, 访问时间为 九月 27, 2025, https://ai.google.dev/gemini-api/docs/billing
19. xAI Releases Grok 4 Fast With Free Access and API Pricing Details - Techloy, 访问时间为 九月 27, 2025, https://www.techloy.com/xai-releases-grok-4-fast-with-free-access-and-api-pricing-details/
20. API | xAI, 访问时间为 九月 27, 2025, https://x.ai/api
21. The AI Language Model Landscape in 2025: Qwen, DeepSeek, and Hunyuan Lead the Pack | by Cogni Down Under | Medium, 访问时间为 九月 27, 2025, https://medium.com/@cognidownunder/the-ai-language-model-landscape-in-2025-qwen-deepseek-and-hunyuan-lead-the-pack-662d65db066a
22. Gemini Developer API Pricing | Gemini API | Google AI for Developers, 访问时间为 九月 27, 2025, https://ai.google.dev/gemini-api/docs/pricing
23. How to Fix Google Gemini 2.5 Pro API Rate Limits - CometAPI - All AI Models in One API, 访问时间为 九月 27, 2025, https://www.cometapi.com/how-to-fix-google-gemini-2-5-pro-api-rate-limits/
24. Gemini 2.5 pro API free tier has a 6m token limit : r/Bard - Reddit, 访问时间为 九月 27, 2025, https://www.reddit.com/r/Bard/comments/1lpb9fl/gemini_25_pro_api_free_tier_has_a_6m_token_limit/
25. Rate limits - OpenAI API, 访问时间为 九月 27, 2025, https://platform.openai.com/docs/guides/rate-limits/usage-tiers
26. Models & Pricing | DeepSeek API Docs, 访问时间为 九月 27, 2025, https://api-docs.deepseek.com/quick_start/pricing
27. Deploying AI Agents Locally with Qwen3, Qwen-Agent, and Ollama - DEV Community, 访问时间为 九月 27, 2025, https://dev.to/bconsolvo/deploying-ai-agents-locally-with-qwen3-qwen-agent-and-ollama-1ddm
28. Deploying local LLM hosting for free with vLLM - Rabiloo, 访问时间为 九月 27, 2025, https://rabiloo.com/blog/deploying-local-llm-hosting-for-free-with-vllm

29. Pricing | Runpod GPU cloud computing rates, 访问时间为 九月 27, 2025, https://www.runpod.io/pricing
30. Rate Limit | DeepSeek API Docs, 访问时间为 九月 27, 2025, https://api-docs.deepseek.com/quick_start/rate_limit
31. Deepseek Free Tier Limits: Features & Usage 2025 - BytePlus, 访问时间为 九月 27, 2025, https://www.byteplus.com/en/topic/382772
32. Benchmarking Quantized LLMs: What Works Best for Real Tasks? - Ionio, 访问时间为 九月 27, 2025, https://www.ionio.ai/blog/llm-quantize-analysis
33. Rate Limits for LLM Providers: working with rate limits from OpenAI, Anthropic, and DeepSeek - Requesty.ai, 访问时间为 九月 27, 2025, https://www.requesty.ai/blog/rate-limits-for-llm-providers-openai-anthropic-and-deepseek
34. Comparison Analysis: Claude 3.5 Sonnet vs GPT-4o - Vellum AI, 访问时间为 九月 27, 2025, https://www.vellum.ai/blog/claude-3-5-sonnet-vs-gpt4o
35. Transparent Pricing for Model APIs & GPU Solutions - Novita AI, 访问时间为 九月 27, 2025, https://novita.ai/pricing
36. vLLM - Qwen docs, 访问时间为 九月 27, 2025, https://qwen.readthedocs.io/en/latest/deployment/vllm.html
37. LlamaIndex vs LangChain: Which Framework Is Best for Agentic AI Workflows? - ZenML, 访问时间为 九月 27, 2025, https://www.zenml.io/blog/llamaindex-vs-langchain