DEP：去中心化大语言模型评估协议

Jianxiang Peng[1*], **Junhao Li**[1*], **Hongxiang Wang**[1*],**HaochengLyu**[1*], **HuiGuo**[1]**Siyi Hao**[1], **Zhen Wang**[1], **ChuangLiu**[2],**ShaoweiZhang**[1],**Bojian Xiong**[1],**YueChen**[1]**ZhuowenHan**[1], **Ling Shi**[1],**Tianyu Dong**[1],**Juesi Xiao**[1], **LeiYang**[1]**Yuqi Ren**[1,3†],**DeyiXiong**[1,3†][1]TJUNLP实验室，天津大学，中国天津[2]天津国家超级计算中心[3]Yuanhui AI{pjasonx, ryq20,dyxiong}@tju.edu.cn

arXiv:2603.01167v1 [cs.CL] 1 Mar 2026

摘要

随着大语言模型（LLM）的快速发展，大量基准测试相继涌现。然而，大多数基准测试缺乏统一的评估标准，且需要手动编写自定义脚本，难以确保结果的一致性和可复现性。此外，主流评估框架采用集中式架构，集中存储数据集和答案，增加了基准泄露的风险。为解决这些问题，我们提出去中心化评估协议（DEP），这是一种通过匹配服务器实现的去中心化、统一且标准化的评估框架，同时不限制基准测试。服务器既可在本地挂载，也可远程部署；适配完成后即可长期复用。通过将用户、大语言模型和基准测试解耦，DEP 实现了模块化、即插即用的评估：基准文件和评估逻辑仅保留在服务器端。在远程场景下，用户无法访问标准答案，从而实现数据隔离和防泄露评估。为便于实际应用，我们开发了**DEP 工具包**，这是一个兼容该协议的工具包，支持断点续传、并发请求和拥塞控制等功能。我们还提供了详细文档，介绍如何将新基准测试适配到 DEP。使用 DEP 工具包，我们对多个基准测试上的多种大语言模型进行了评估。实验结果验证了 DEP 的有效性，并表明它能够降低部署基准评估的成本。截至2026年2月，我们已适配60多个基准测试，并将继续推动社区共建，以支持不同任务和领域的统一评估。

网站： http://dep.openeval.org.cn

*共同贡献
†通讯作者

1 引言

随着大语言模型不断进步，先前的基准测试很快趋于饱和，推动新的评测任务和数据集不断涌现。仅在 ACL 2025（Che 等人，2025 年）和 EMNLP 2025（Christodoulopoulos 等人，2025 年）上，就出现了近 700 项与基准测试相关的研究。然而，大多数基准测试都是独立发布的，其数据格式、推理流程和评估脚本各不相同，且缺乏统一的评估接口规范（Ni 等人，2025a）。因此，想要在特定基准测试上进行评估的模型开发者，通常需要自行编写推理脚本，或手动将基准测试适配到现有评估平台。这不仅增加了时间成本，也使得确保结果的一致性和可复现性变得困难。

此外，现有评估框架仅覆盖少数热门基准测试，对新发布的基准测试或特定领域评估的支持有限（Liu 等人，2024b；He 等人，2024）。这些框架遵循集中式设计，将推理逻辑、数据集、评估指标和执行控制集成在一起。此类框架规模庞大且高度耦合，可插拔性有限，学习成本也较高（Yao 等人，2025；Li 等人，2023 年；He 等人，2024）。基准测试构建者往往需要按照平台特定格式重构数据和逻辑，这限制了评估方法的多样性。此外，集中式框架需要集成用于评估的标准答案，因此存在测试集泄漏和数据污染的风险。

为应对这些挑战，我们提出了**去**中心化**评**估 **协**议（**DEP**），鼓励大语言模型社区共建。图 1 展示了 DEP 与集中式框架之间的差异。DEP 不会强制要求原始基准测试

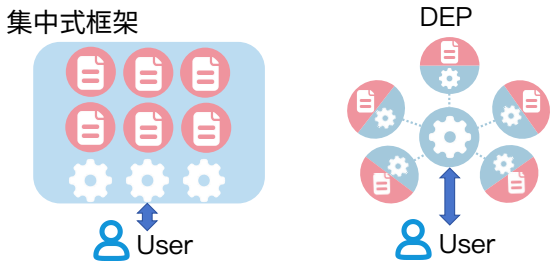


图 1：集中式框架与我们的 DEP 的对比。红色表示基准文件，蓝色表示评估模块。

数据格式。相反，DEP 为每个数据集构建相匹配的基准测试服务器脚本，实现统一、标准化的评估接口，从而支持不同模型、数据集和评估方法之间的兼容。就架构而言，DEP 将大语言模型、客户端和基准测试解耦。在 DEP 中，客户端仅通过协议定义的标准接口与基准测试服务器通信，而评估逻辑和基准测试相关资源则完全包含在服务器端。如果基准作者希望公开真实答案，可以发布相应脚本以供本地挂载，无需进行任何额外工作。相应地，这些服务器也可以远程部署。开发者既可以独立部署，也可以将其托管在基准服务器平台上。我们将提供一个专用平台，开发者可以通过我们的网站申请托管。

此外，我们实现了**DEP 工具包**，这是一款支持大规模大语言模型基准测试的评估工具，具备断点续传、并发请求和拥塞控制功能。它兼容本地推理和基于 API 的推理，并支持多种评估方法和指标，包括自定义指标和大语言模型评审。

我们使用 DEP 工具包对各种开源和闭源大语言模型进行了系统评估。实验结果表明，DEP 在降低基准集成和复用成本的同时，也展现出良好的效果。

总而言之，我们的贡献如下：

- 我们提出了 DEP，这是一种将基准测试与评估客户端解耦的去中心化评估协议。与集中式评估相比，DEP 具备零代码便捷性、可复现评估、数据隐私保护和社区驱动的共享等优势。
- DEP 允许基准作者集成其基准测试，而无需更改原始数据集格式。该协议兼容多种评估集格式和推理方法，支持广泛的评估指标和方法。
- 我们发布了 DEP 工具包，这是一个实现该协议的综合工具包，具备用于大规模评估的实用功能。
- 我们提供协议和开发文档，以便快速适配。截至2026年2月，我们已将超过 60 个基准测试适配至 DEP。[1]

2 相关工作

大语言模型基准测试。 随着大语言模型能力的不断发展，早期的基准测试逐渐过时，难以有效区分不同大语言模型之间的性能。为了更细致地刻画模型在知识（Hendrycks 等人，2020b；Zhang 等人，2023 年）、代码生成（Chen 等人，2021；Zhuo 等人，2024）以及安全对齐（Mazeika 等人，2024；Wang 等人，2023 年）等领域的表现，研究人员不断提出新的评测任务和数据集（Ni 等人，2025a；Shen 等人，2023）。这导致评测生态系统高度碎片化。大多数基准测试以独立代码仓库的形式发布，使用各不相同的数据格式、推理方法和评估脚本，使标准化评估接口的问题一直未得到解决。DEP 提出了一套统一协议，建立标准化接口，同时不限制基准测试底层的表达方式或格式。

评估框架。 大语言模型社区推出了各种集中式评估框架。例如，LM Evaluation Harness[2] 提供统一的任务封装和模型适配接口，大幅降低了使用主流基准测试的门槛。类似地，OpenAI Evals[3] 支持基于模板和评分函数灵活定义评估。其他示例包括Liu 等人 (2024b) 和Yu 等人 (2024)。Hugging Face Evaluate[4] 集成了多种常见指标的实现。然而，这些

[1] http://dep.openeval.org.cn/News [2] https://github.com/EleutherAI/lm-evaluation-harness [3] https://github.com/openai/evals [4] https://github.com/huggingface/evaluate

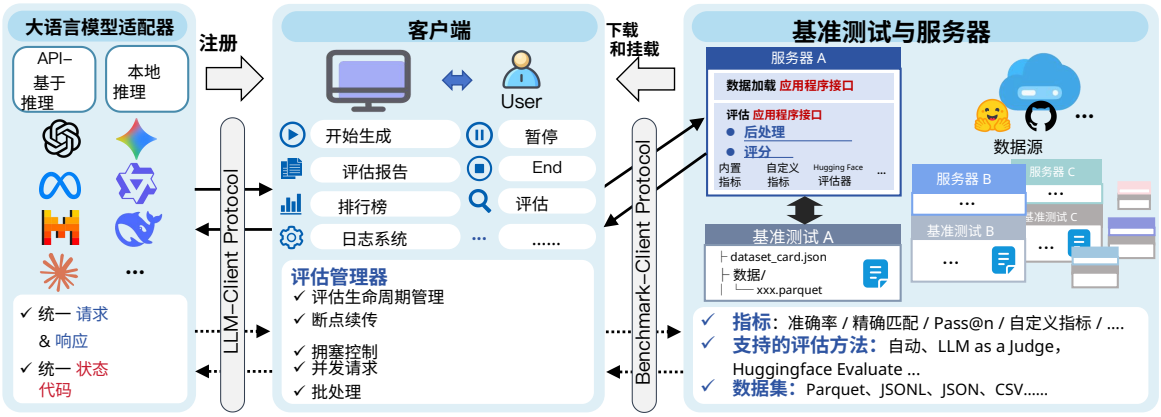


图 2：DEP 概览。客户端通过统一接口连接大语言模型和基准测试。大语言模型适配器将各种大语言模型接口和推理方法标准化，而基准测试服务器则处理来自客户端的请求，例如数据加载和评估提交。

框架通常采用集中式架构，将数据集、大语言模型推理逻辑和评估逻辑整合在单一平台中。这会导致系统臃肿、可扩展性差。基准测试通常必须经过重构才能适配平台特定的数据格式和接口规范，从而限制了灵活性。例如，虽然 LM Evaluation Harness 支持通过 YAML 文件添加新数据集，但用户必须遵循严格受限的字段规范。与此同时，OpenAI Evals 不支持自定义评估脚本。相比之下，DEP 并不打算再构建一个评测平台，而是引入一种协议级机制，使基准测试能够以去中心化的方式进行评估，就像进行一次 API 调用一样简单。

防止数据污染。随着训练语料库不断扩展，测试集污染日益受到关注（Ni 等人，2025b）。如果标准答案被公开发布，大语言模型可能会在训练阶段间接接触到这些数据，从而损害评估结果的可信度（Balloccu 等人，2024；Fang 等人，2025）。为降低这一风险，一些组织采用私有评估或在线评估，要求开发者提交大语言模型的生成结果，再由评估器在封闭环境中计算分数并公布排名（Wang 等人，2018 年；Huang 等人，2023 年；Liu 等人，2024b）。尽管这种方法能够减少数据污染，但它往往依赖人工协调或特定平台的实现方式，且缺乏统一的技术协议。DEP 通过在架构层面直接支持标准答案隔离机制来解决这一问题。

3 DEP 设计

DEP 采用由大语言模型适配器、客户端和基准测试服务器组成的三组件架构。如图2所示，大语言模型适配器提供统一的大语言模型调用接口，以屏蔽不同大语言模型推理方法和 API 之间的差异。客户端负责管理评估生命周期、调度任务，并控制并发和速率。基准测试服务器负责处理来自客户端的评估提交，并实现评估与指标记录逻辑。

3.1 大语言模型适配器

DEP 在大语言模型适配器中定义了统一的调用接口。这种结构使客户端无需感知具体的大语言模型类型，即可完成调度和评估。

（1）输入与输出：输入是指在单个评估实例中提供给大语言模型的基准测试文本内容，而输出则包括标准化的状态码、生成的响应、错误日志和元数据。**（2）通信协议**：大语言模型适配器与客户端通过统一的 JSON 通信格式和标准化的状态码进行交互。整体通信方式遵循 RESTful HTTP 语义，其行为语义根据 RFC 2119 约束定义。例如，状态码 200 绝不能重试，而状态码 429 则应触发并发量降低和退避重试。通过此规范，客户端可以根据状态码自动执行重试、速率限制或终止策略，避免因 API 原生状态码存在差异而导致行为不一致

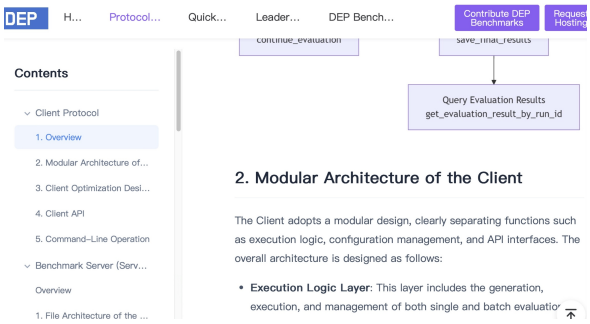


图 3：DEP 网站截图。

状态码。**（3）模型管理**：模型卡作为元数据，用于声明模型能力、参数规模和 API 端点，从而实现自动发现机制。通过继承基础模型类，所有模型都共享协议逻辑，例如日志记录和统一的返回格式，确保大语言模型能够无缝集成。

3.2 基准测试服务器

在 DEP 中，每个基准测试都被打包为一个独立的基准测试服务器，其中包含一个数据加载模块和一个评估器模块，并通过统一的数据集卡元数据文件公开接口规范和配置。

DEP 允许开发者将现有基准测试封装到 DEP 中，而无需更改原始数据集格式。服务器连接原始基准文件（例如 Parquet、JSONL、CSV 和自定义结构），并通过其接口将不同的文件格式和结构标准化为统一的大语言模型输入格式。这些输入遵循数据加载模块的预设逻辑，例如作者设定的思维链（CoT）提示，从而确保评估的可复现性。

评估过程中，评估模块会加载客户端提交的大语言模型生成的输出和标准答案，应用预定义的后处理方法，并在内部执行评估逻辑。评估接口会返回结构化结果，包括总体性能、子任务性能以及可选的详细分数。除了常见指标外，开发者还可以使用自定义评估脚本。该系统也兼容 Hugging Face Evaluate、LLM as a Judge 以及其他外部评估工具。由于所有评估逻辑和标准答案都严格保留在服务器端，客户端和大语言模型均无法访问，从而在架构层面有效防止答案泄漏和数据污染。

3.3 客户端

客户端作为大语言模型适配器与基准测试之间的协调层，负责任务编排和生命周期管理。任务生命周期状态包括：运行中、已暂停、已完成和失败；所有状态转换均由唯一的评估 ID 标识。

客户端还支持自动发现基准测试和模型：通过扫描目录中的数据集卡和模型卡来加载服务器及适配器配置，从而实现即插即用式扩展。

4 DEP 工具包

我们开发了基于 DEP 的综合评估客户端 **DEP 工具包**。以下章节将介绍 DEP 工具包的主要功能实现。使用说明请参阅 DEP 网站上的在线文档。[5]

用户交互。DEP 工具包提供完整的评估生命周期管理。可通过以下方式安装该工具包：

```
$ pip install llmdep
```

用户可以通过命令行界面（CLI）发起新的评测任务。在评估过程中，用户可以实时监控任务状态，包括当前进度、执行阶段和错误日志。如果任务因外部中断而终止，系统支持根据评估 ID 断点续跑，避免重复计算和浪费令牌。客户端还支持统一查询和汇总展示结果，用户可以查看单个模型在特定基准测试上的详细指标，也可以对多个模型进行横向比较，生成按任务维度分类的排行榜。

系统管理。 DEP 工具包采用模块化架构设计，将评估控制逻辑、配置管理、API 层、日志系统和存储解耦。自动发现机制使系统能够扫描目录，并动态加载兼容的基准测试和大语言模型。添加新的大语言模型或基准测试时，只需将其加载到指定目录即可完成集成。这种即插即用方式增强了可扩展性，并降低了运维

[5] http://dep.openeval.org.cn/ProtocolDocs

开销。此外，所有结果和中间状态都以结构化格式存储在本地。

评估控制器。 DEP 工具包实现了面向大规模大语言模型评测的任务调度与流量控制机制。它同时支持并发执行与批量执行，并提供可配置参数，用于控制生成与评估的执行粒度。针对基于 API 的推理，DEP 工具包内置了基于令牌桶算法的拥塞控制机制。该机制用于平滑请求流量并限制瞬时突发，从而避免频繁触发应用程序接口速率限制。此外，系统会详细记录请求历史、速率限制事件以及异常类型。

内置指标。 我们的工具包提供多种用于基准测试的预设指标（例如准确率、精确匹配、F1），开发者可直接使用这些指标，以便构建 DEP 服务器。

5 评估生态策略

我们进一步提出以基准测试生命周期管理与数据治理为核心的 DEP 评估生态策略。该策略旨在应对大语言模型评测中的两项核心挑战：基准测试碎片化、测试集污染以及评估不可持续性。

5.1 协作构建与共享

我们持续将基准测试适配到 DEP，并发布其实现以促进复用。基于这一基础，我们将定期开展大语言模型评测并发布全面报告，系统地追踪模型在知识覆盖范围、推理能力和安全对齐方面的发展趋势。

此外，作为一个去中心化系统，DEP 是面向大语言模型社区的协议生态系统。我们鼓励基准作者贡献其适配实现。对于已有 DEP 适配方案的基准测试，用户可以将其集成到客户端中，无需编写代码即可使用。这种低成本方式使评估组织者能够无缝引入新的基准测试，大幅缩短纳入新任务和数据集所需的时间。

5.2 可持续评估

为防止大语言模型在预训练或对齐阶段接触到公开可用的基准测试答案，研究人员通常采用在线评估设置，即用户提交大语言模型的输出，由评估组织者评分并发布最终结果。DEP 天然支持这一评估范式。由于标准答案和评估逻辑仅存在于基准测试端，客户端无法访问标准答案。这种结构分离有助于防止基准测试污染和答案泄漏。在附录 A.2 中，我们给出一个用例，展示服务器的私有部署如何在实践中实现此类保密评估。

6 实验

2026 年 1 月，我们通过 DEP 工具包使用 12 个开源和闭源大语言模型，在 10 项基准测试上开展了评估，以验证系统的可用性。本节介绍评估的主要发现。此外，我们还招募用户来评估 DEP 的实用性和易用性。附录A中给出了两个用例的详细说明，涵盖标准本地评估以及通过远程部署的服务器进行评估。

6.1 大语言模型评测

设置。 我们评测了 6 个开源大语言模型和 6 个闭源大语言模型，开源模型的规模从 0.6B 到 685B 参数不等。我们使用了总计 1,800 万个词元和 130 GPU 小时的计算资源（NVIDIA A100 80G）。对于闭源大语言模型和 DeepSeek-V3.2，我们使用了 API 调用。表 2和表3全面列出了评估中使用的所有任务和大语言模型。

结果。 实验结果如图4所示。首先，在 ARC 和 BoolQ 等经典基准测试中，除 Qwen3-0.6B 等极小型语言模型（小型语言模型）外，大多数大语言模型之间的性能差距相对较小。这表明，一些传统基准测试已开始显现饱和，导致评估的区分能力下降。相比之下，数学推理等更复杂的任务表现出更大的性能差距，仍然是区分最先进的大语言模型的核心指标。

此外，结果显示出明显的规模效应。在几乎所有任务中，随着大语言模型规模增大，Qwen3 系列的性能都持续提升。从 0.6B 到 320亿参数，其基准测试分数几乎单调上升，

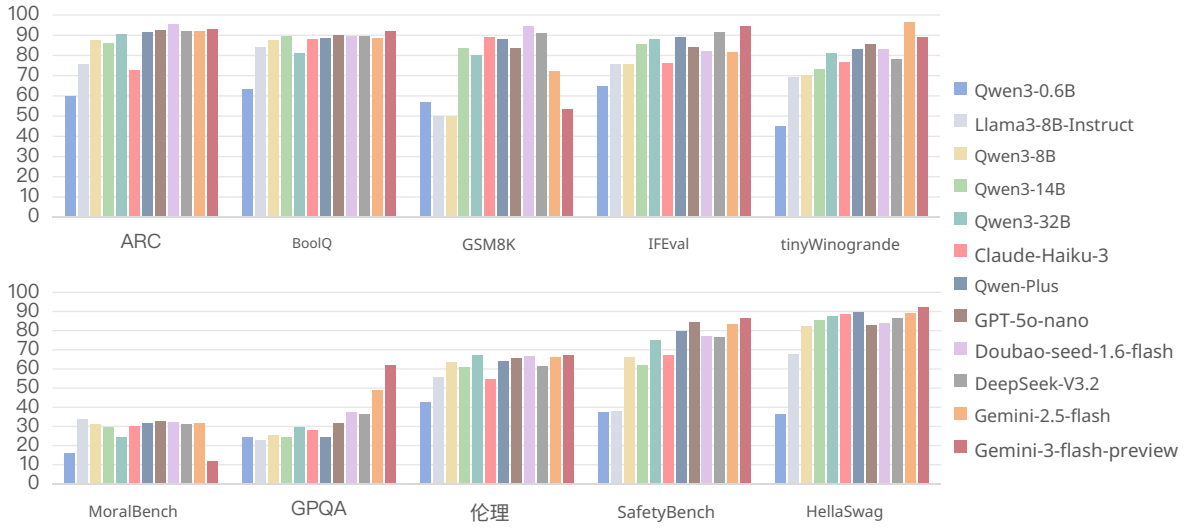


图 4：使用 DEP 进行大语言模型评测的主要结果。

这表明规模仍是解决复杂问题的关键因素。然而，略显意外的是，Gemini-3-flash-preview 在 GSM8K任务上的表现相对较差。尽管它能够遵循 CoT 格式并生成结构正确的输出，但仍会给出错误答案，这表明其数学推理能力可能存在一定不足。

最后，尽管规模更大的大语言模型展现出更强的知识水平和能力，但在伦理对齐和安全方面未必胜过小型语言模型。这表明，在持续提升大语言模型能力的同时，我们也必须密切关注伦理对齐和潜在的安全风险。这些发现证明了基于DEP的评测的有效性。

6.2 用户研究

我们开展了一项用户研究，邀请 8 名参与者对 12 个基准测试进行评估。研究比较了四种不同的方法：（1）手动编写评估脚本；（2）适配评估框架 OpenEval；（3）适配 DEP；（4）复用现有的 DEP 基准测试。我们测量了每个基准测试所需的平均耗时和平均代码行数（LoC）。详细的实验设置见附录 B。

如表 1所示，用户和基准测试作者可以使用可插拔的 DEP 服务器，而无需增加工作量（与手动实现或适配评估框架相比）。一旦 DEP 服务器可用，用户便可以无需编写代码即可轻松挂载基准测试。

指标	MI	A2E	A2D	RD
耗时（分钟）	45	92	69	3
LoC	109	215	162	0

表 1：准备单个基准测试所需的平均时间和代码行数（LoC）。MI：手动实现；A2E：适配评估框架；A2D：适配 DEP；RD：复用现有 DEP 基准测试。

7 结论

我们提出了 DEP，它提供统一的即插即用接口，同时将基准测试与评估客户端解耦，以支持数据隔离。此外，我们发布了 DEP 工具包，这是一种符合协议的评估工具，旨在满足大规模基准测试的实际需求，包括任务管理、资源调度和故障恢复。通过将标准化协议接口与系统化工具能力相结合，我们构建了一个可扩展、可复现且可持续的基准测试评估协议。这使大语言模型评测更加便捷、规范。实验结果证明了使用 DEP 进行评估的有效性。与其他适配方法相比，完成适配过程无需额外工作量，适配完成后即可方便地重复使用。

局限性

DEP 的整体有效性取决于可用的符合协议的服务器数量。尽管我们致力于持续将基准测试适配到 DEP，但基准测试作者和开发者的贡献至关重要，而这不可避免地会引入额外的集成工作量。目前，DEP 专注于大语言模型评测，尚无法评估涉及复杂外部交互的智能体，例如在 Docker 环境中执行软件开发任务的智能体。这是 DEP 未来拓展的方向。

参考文献

Anthropic。2024。Claude 3 Haiku：迄今为止速度最快的模型。

白金泽、白帅、褚云飞、崔泽宇、党凯、邓晓东、范阳、葛文斌、韩宇、黄飞等 1 人。2023 年。通义千问技术报告。*arXiv* 预印本 *arXiv:2309.16609*。

Simone Balloccu、Patrícia Schmidtová、Mateusz Lango 和 Ondrej Dusek。2024。泄漏、作弊、重蹈覆辙：闭源大语言模型中的数据污染与评测不端行为。载于第 *18* 届欧洲计算语言学协会分会会议论文集（第 *1* 卷：长篇论文），第 67–93 页，马耳他圣朱利安斯。计算语言学协会。

字节跳动。2025。Seed1.6 所用技术介绍。

Wanxiang Che、Joyce Nabende、Ekaterina Shutova 和 Mohammad Taher Pilehvar，编者。2025 年。第 *63* 届计算语言学协会年会论文集（第 *1* 卷：长篇论文）。计算语言学协会，奥地利维也纳。

Mark Chen、JerryTworek、Heewoo Jun、QimingYuan、Henrique Ponde De Oliveira Pinto、Jared Kaplan、Harri Edwards、Yuri Burda、Nicholas Joseph、Greg Brockman 及其他 1 位作者。2021 年。评估基于代码训练的大语言模型。*arXiv* 预印本 *arXiv:2107.03374*。

Christos Christodoulopoulos、TanmoyChakraborty、Carolyn Rose 和 Violet Peng，编者。2025 年。*2025* 年自然语言处理实证方法会议论文集。计算语言学协会，中国苏州。

Christopher Clark、Kenton Lee、Ming-Wei Chang、Tom Kwiatkowski、Michael Collins 和 Kristina Toutanova。2019 年。BoolQ：探索自然语言是非问题令人意外的难度。载于《*2019* 年北美计算语言学协会分会年会论文集：人类语言技术》，第 *1* 卷（长篇与短篇论文），第 2924–2936 页。

Peter Clark、Isaac Cowhey、Oren Etzioni、Tushar Khot、Ashish Sabharwal、Carissa Schoenick 和 Oyvind Tafjord。2018 年。你认为自己已经解决了问答问题吗？来试试 ARC，也就是 AI2推理挑战赛。*arXiv* 预印本 *arXiv:1803.05457*。Karl Cobbe、Vineet Kosaraju、Mohammad Bavarian、Mark Chen、Heewoo Jun、Lukasz Kaiser、Matthias Plappert、Jerry Tworek、Jacob Hilton、Reiichiro Nakano、Christopher Hesse 和 John Schulman。2021 年。训练验证器解决数学应用题。预印本，arXiv:2110.14168。Gheorghe Comanici 等人。2025。Gemini 2.5：借助高级推理、多模态、长上下文和新一代智能体能力，推动前沿发展。*arXiv* 预印本 *arXiv:2507.06261*。

方义雄、孙天然、石玉玲、王敏和顾晓东。2025。LastingBench：抵御基准测试中的知识泄漏。载于计算语言学协会：*EMNLP 2025* 论文集，第 18304–18317 页，中国苏州。计算语言学协会。

谷歌。2025。Gemini 3 智能新时代。

Aaron Grattafiori 等 2 人。2024。Llama 3 模型家族。*arXiv* 预印本 *arXiv:2407.21783*。

何超群、罗仁杰、胡盛鼎、赵然驰、周杰、吴航浩、张家杰、韩旭、刘知远和 Maosong Sun。2024。UltraEval：用于大语言模型灵活全面评估的轻量级平台。载于 计算语言学协会第 *62* 届年会论文集（第 *3* 卷：系统演示），第247–257页，泰国曼谷。计算语言学协会。

Dan Hendrycks、Collin Burns、Steven Basart、Andrew Critch、Jerry Li、Dawn Song 和 Jacob Steinhardt。2020a。使人工智能与共同的人类价值观保持一致。*arXiv* 预印本 *arXiv:2008.02275*。

Dan Hendrycks、Collin Burns、StevenBasart、AndyZou、Mantas Mazeika、Dawn Song 和 Jacob Steinhardt。2020b。衡量大规模多任务语言理解能力。*arXiv* 预印本 *arXiv:2009.03300*。

黄宇臻、Yuzhuo Bai、朱志豪、张君磊、张景涵、苏唐军、刘俊腾、吕传成、张一凯、傅尧和另外 1 位作者。2023 年。C-Eval：面向基础模型的多层次多学科中文评测套件。神经信息处理系统进展，36：62991–63010。

纪建超、陈雨桐、金明宇、徐武江、华文月和张永锋。2025。道德基准：大语言模型的道德评测。*ACM SIGKDD*探索通讯，27(1):62–71。

李艳阳、赵建桥、郑铎、胡子元、陈志、苏晓辉、黄永锋、黄世嘉、DahuaLin、吕迈克和王立威。

2023 年。 CLEVA：中文语言模型评测平台。见 *2023* 年自然语言处理实证方法会议论文集：系统演示，第 186–217 页，新加坡。计算语言学协会。

Aixin Liu、Bei Feng、Bing Xue、Bingxuan Wang、Bochao Wu、Chengda Lu、Chenggang Zhao、Chengqi Deng、Chenyu Zhang、Chong Ruan及另外1人。2024a。Deepseek-v3技术报告。*arXiv*预印本 *arXiv:2412.19437*。

Chuang Liu、Linhao Yu、Jiaxuan Li、Renren Jin、Yufei Huang、Ling Shi、Junhui Zhang、Xinmeng Ji、Tingt- ing Cui、Tao Liu、Jinwang Song、Hongying Zan、Sun Li和Deyi Xiong。2024b。OpenEval：跨能力、对齐与安全性评估中文大语言模型的基准测试。载于计算语言学协会第*62*届年会论文集（第*3*卷：系统演示），第190–210页，泰国曼谷。计算语言学协会。

Mantas Mazeika、Long Phan、Xuwang Yin、Andy Zou、Zifan Wang、Norman Mu、Elham Sakhaee、Nathaniel Li、Steven Basart、Bo Li及另外1人。2024。Harm-bench：用于自动化红队测试和稳健拒绝的标准化评估框架。*arXiv*预印本 *arXiv:2402.04249*。

Shiwen Ni、Guhong Chen、Shuaimin Li、Xuanang Chen、Siyi Li、Bingli Wang、Qiyao Wang、Xingjian Wang、Yifan Zhang、Liyang Fan、Chengming Li、Ruifeng Xu、Le Sun 和 Min Yang。2025a。关于大语言模型基准测试的综述。预印本，arXiv:2508.15361。

Shiwen Ni、Xiangtao Kong、Chengming Li、Xiping Hu、Ruifeng Xu、Jia Zhu 和 Min Yang。2025b。仅在基准测试上训练还不够。载于*AAAI*人工智能会议论文集，第39卷，第24948–24956页。

OpenAI。2025。推出 GPT-5。

Felipe Maia Polo、Lucas Weber、Leshem Choshen、Yuekai Sun、Gongjun Xu 和 Mikhail Yurochkin。2024。tinybenchmarks：用更少的示例评估大语言模型。*arXiv* 预印本 *arXiv:2402.14992*。

David Rein、Betty Li Hou、Asa Cooper Stickland、Jackson Petty、Richard Yuanzhe Pang、Julien Dirani、Julian Michael 和 Samuel R Bowman。2024 年。GPQA：研究生水平的谷歌不可解问答基准。载于首届语言建模会议。

Tianhao Shen、Renren Jin、Yufei Huang、Chuang Liu、Weilong Dong、Zishan Guo、Xinwei Wu、Yan Liu 和 Deyi Xiong。2023 年。大语言模型对齐：一项综述。*arXiv* 预印本 *arXiv:2309.15025*。

Alex Wang、Amanpreet Singh、Julian Michael、Felix Hill、Omer Levy 和 Samuel Bowman。2018 年。GLUE：用于自然语言理解的多任务基准及分析平台。载于 *2018* 年 *EMNLP* 研讨会 *BlackboxNLP*：分析和用于 *NLP* 的神经网络解释，第 353–355 页，比利时布鲁塞尔。计算语言学协会。

Yuxia Wang、Haonan Li、Xudong Han、Preslav Nakov 和 Timothy Baldwin。2023 年。Do-not-answer：用于评估大语言模型安全防护机制的数据集。*arXiv* 预印本 *arXiv:2308.13387*。

An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv 等另外 1 人。2025。通义千问 3 技术报告。*arXiv* 预印本 *arXiv:2505.09388*。

Jing Yao、Xiaoyuan Yi、Shitong Duan、Jindong Wang、Yuzhuo Bai、Muhua Huang、Yang Ou、Scarlett Li、Peng Zhang、Tun Lu、Zhicheng Dou、Maosong Sun、James Evans 和 Xing Xie。2025。 价值罗盘基准：一个面向大语言模型价值评估的全面、生成式且自我演进的平台。见《第*63*届计算语言学协会年会论文集（第*3*卷：系统演示）》，第666–678页，奥地利维也纳。计算语言学协会。

Zhuohao Yu、Chang Gao、Wenjin Yao、Yidong Wang、Zhengran Zeng、Wei Ye、Jindong Wang、Yue Zhang 和 Shikun Zhang。2024。 FreeEval：一个用于大语言模型可信且高效评估的模块化框架。见《*2024*年自然语言处理经验方法会议论文集：系统演示》，第1–13页，美国佛罗里达州迈阿密。计算语言学协会。

Rowan Zellers、Ari Holtzman、Yonatan Bisk、Ali Farhadi 和 Yejin Choi。2019年。HellaSwag：机器真的能完成你的句子吗？载于计算语言学协会第*57*届年会论文集，第4791–4800页。

Xiaotian Zhang、Chunyang Li、YiZong、Zhengyu Ying、Liang He 和 Xipeng Qiu。2023 年。评估大语言模型在高考基准上的表现。*arXiv*预印本 *arXiv:2305.12474*。

Zhexin Zhang、Leqi Lei、Lindong Wu、Rui Sun、Yongkang Huang、Chong Long、Xiao Liu、Xuanyu Lei、唐杰和黄民烈。2024。SafetyBench：评估大语言模型的安全性。载于计算语言学协会第*62*届年会论文集（第*1*卷：长篇论文），第15537–15553页，泰国曼谷。计算语言学协会。

Terry Yue Zhuo、Minh Chien Vu、Jenny Chim、Han Hu、Wenhao Yu、Ratnadira Widyasari、Imam Nur Bani Yusuf、Haolan Zhan、Junda He、Indraneil Paul 等共 12 人。2024。Bigcodebench：使用多样化的函数调用和复杂指令对代码生成进行基准测试。*arXiv* 预印本 *arXiv:2406.15877*。

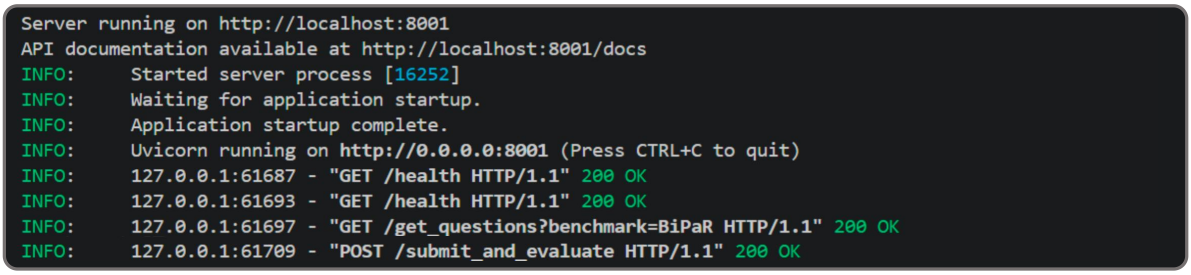


图 5：用于远程评测的服务器端接口。

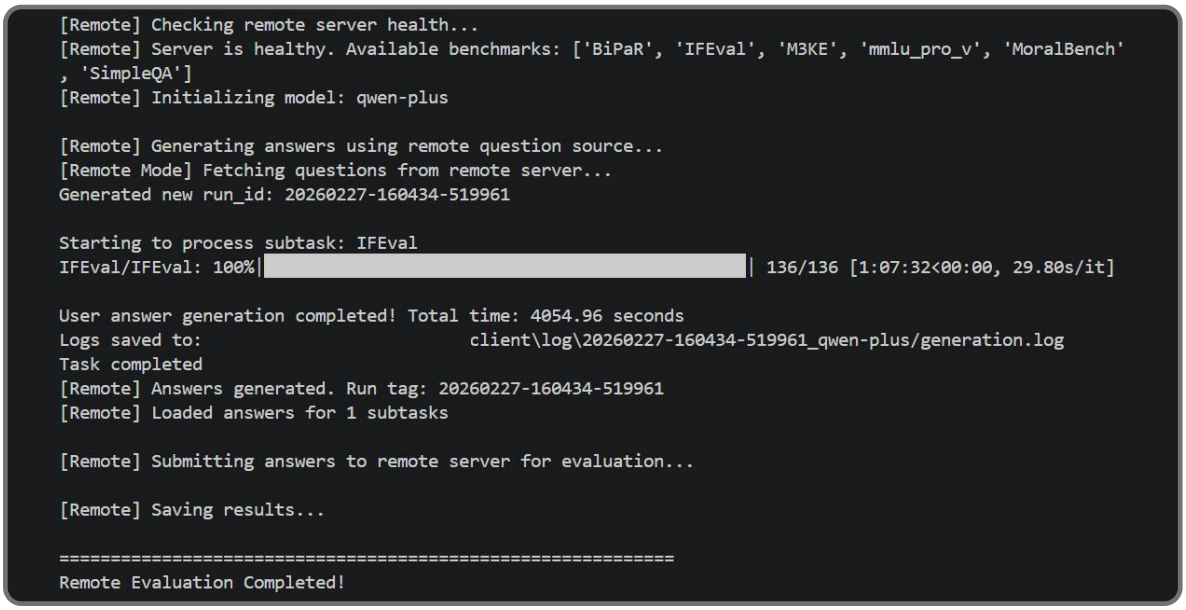


图 6：连接远程基准测试服务器的客户端连接。

一个案例研究

A.1 常规评测

图 7展示了使用 DEP 工具包进行评估的典型示例。我们使用 Doubao-seed-1.6-flash 模型对 IFEval 数据集进行了评估。截图展示了执行日志，涵盖从数据集加载、评测执行到结果提交和最终评分的完整工作流程。

A.2 远程部署

如果基准测试开发者希望以私密方式开展评估（即不披露标准答案），则需要部署服务器并开放相应端口。我们模拟部署了六个基准测试服务器进行测试，包括 BiPaR、IFEval、M3KE、MMLU_Pro、MoralBench 和 SimpleQA。

如图所示 5，这是 BiPaR 数据集的服务器端，用于接收评估任务客户端提交的任务。如图 6 所示，客户端展示了六个已部署且可用的基准测试服务器，以及评估的执行与提交过程。

B 用户研究详情

我们招募了 8 名拥有本科学位且具有大语言模型研究背景的计算机科学专业人士，并按每小时 9 美元的标准向他们支付报酬。我们比较了四种不同的方法：手动实现、适配 OpenEval 平台、适配 DEP 服务器，以及使用现有 DEP 服务器。

手动实现。 1. 研究基准测试论文和数据文件。2. 编写串行评测脚本，其中包括基本的测试集遍历、结果存储和评估逻辑。3. 验证并测试脚本。4. 执行评估。

适配 OpenEval。 1. 研究基准测试论文和数据文件。2. 重新整理并调整基准测试，使其符合平台的数据格式

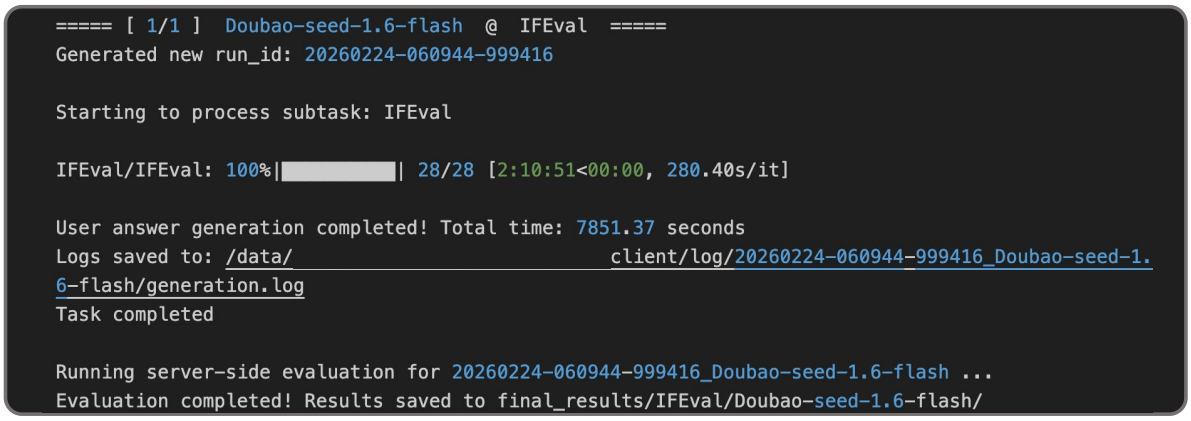


图7：使用 DEP 工具包进行常规评测的用例。

ID	Type	模型名称	#参数	访问方式
1	闭源	Claude-Haiku-3 (Anthropic，2024)	-	API
2	闭源	GPT-5o-nano (OpenAI, 2025)	-	API
3	闭源	Gemini-2.5-flash (Comanici等人,2025)	-	API
4	闭源	Gemini-3-flash-preview (谷歌, 2025)	-	API
5	闭源	Doubao-seed-1.6-flash（字节跳动，2025）	-	API
6	闭源	Qwen-Plus（Bai 等人，2023 年）	-	API
7	开源	DeepSeek-V3.2（Liu 等人，2024a）	37/785B	API
8	开源	Qwen3-0.6B（Yang 等人，2025）	0.6B	权重
9	开源	Qwen3-8B（Yang 等人，2025）	8B	权重
10	开源	Qwen3-14B（Yang 等人，2025）	14B	权重
11	开源	Qwen3-32B（Yang 等人，2025）	32B	权重
12	开源	Llama3-8B-Instruct(Grattafiori 等人，2024）	8B	权重

表 2：第 6.1 节中使用的大语言模型。

和评估方法要求。3. 验证并测试脚本。4. 通过平台执行评估。

适配 DEP。 1. 研究基准测试论文和数据文件。2. 根据协议文档，基于 DEP 模板脚本调整基准测试。3. 验证并测试脚本。4. 使用 DEP 工具包执行评估。

使用现有的基准测试服务器。 1. 将目标基准测试服务器加载到 DEP 工具包中。2. 使用 DEP 工具包执行评估。

对于前三种方法，我们记录了步骤 2 和 3 所需的平均时间和代码行数。对于第四种方法，我们记录了步骤 1 和 2 的这些指标。所有代码均经过交叉验证，以确保正确性。

ID	基准测试
1	GSM8K（Cobbe 等人，2021）
2	GPQA（Rein 等人，2024）
3	IFEval（Cobbe 等人，2021）
4	MoralBench (Ji 等人,2025)
5	ARC (Clark 等人, 2018 年)
6	Ethics (Hendrycks 等人, 2020a)
7	BoolQ (Clark 等人, 2019年)
8	HellaSwag (Zellers 等人,2019年)
9	tinyWinogrande(Polo 等人,2024)
10	SafetyBench (Zhang 等人，2024)

表3：在第 6.1 节中评估的基准测试。