开放权重大模型的技术扩散与分层创新

摘要：开放权重大模型已成为经济体系中的关键基础设施，本文将开放权重大模型界定为一种“选择性开放技术系统”，借助大语言模型辅助识别技术路线与技术层级，系统考察开放权重的技术扩散、层级创新与跨层反馈。研究发现：第一，开放权重大模型通过模型谱系形成强大的向下扩散通道，但核心层缺乏可观察的有向传播结构。第二，下游模型并非被动复制上游，而是在后训练、适配、量化、推理部署等外围层形成活跃的再加工活动。第三，核心层技术路线的更新主要表现为非谱系式共同采用，呈现"弱连接广泛、强连接内聚"的分层结构；剔除通用 L1 架构标签后，跨家族强连接下降 80.5%，表明 T1 层共同采用的主体是基础架构范式趋同，仅存 DeepSeek—Nemotron 等少数模型共同采用具体技术路线。因此，开放权重大模型生态形成了“强向下扩散—活跃外围再加工—核心层横向共同采用—弱公开回流证据”的非对称开放技术系统，系统性扩大了下游行动者的创新机会，但现有公开数据尚不能证明下游创新已系统性反馈到 T1 核心层。

关键词：开放权重大模型；技术扩散；模型谱系

本文仅是阶段性成果，可视为研究预览版 v0.1

第一章 作为选择性开放技术系统的开放权重大模型

1.1 从完整开源到开放权重

大模型已经成为当前经济体系中的关键基础设施之一，得益于其泛化能力，其被广泛用于代码生成、多模态理解、教育、科研和企业办公自动化等场景，日益成为软件开发、数据分析、内容生产和智能体系统的重要通用能力底座。正因大模型具有明显的基础设施属性，其开放方式会极大影响外部开发者、企业和研究机构能否进入模型生态，能在哪些层级进行运用和知识再生产。

在传统软件语境中，完整开源通常意味着外部主体不仅可以获得可运行程序，还可以获得源代码，并在许可证允许范围内检查、修改和再分发该软件。更一般地说，开源还应允许派生作品在相应条件下继续发布。因此，完整开源的核心是源代码、修改权、再分发权和许可证自由度共同构成的制度安排。然而，在大模型语境中，“开源”一词经常被更宽泛地使用。许多“开源大模型”并未开放完整训练代码、训练数据或训练日志，也未必提供足以独立复现模型训练过程的材料。即使模型权重可以下载，外部主体通常也难以完整进入模型的预训练和核心研发过程。因此，若直接沿用传统软件的完整开源概念，就会高估大模型开放的深度。

本文因此将研究对象界定为“开放权重大模型”。所谓开放权重，指模型发布者至少向外部主体提供可下载或可访问的模型权重，使其能够在本地或第三方环境中加载模型，并在许可证和技术条件允许范围内进行推理、微调、适配、量化或再分发。开放权重通常还伴随不同程度的辅助工件开放，例如 config 文件、分词器文件、模型卡、评测结果、推理代码或微调代码。但这些辅助工件并不必然完整，也不必然包含训练数据和完整训练流程。

由此可见，开放权重大模型既不同于完全封闭的专有模型，也不同于传统意义上的完整开源软件。它开放了外部主体进行使用和再加工所必需的一部分模型制品，却通常保留了更接近核心知识生产过程的训练数据、完整训练流程和训练代码。正是在这个意义上，本章将开放权重大模型理解为一种“选择性开放技术系统”其有选择地开放某些技术组件并在不同技术层级之间形成开放深度差异。

这一界定也为本章后续分析奠定基础。本章致力于探究当权重和部分模型制品被开放后，哪些技术层级变得可进入？外部主体主要在哪些层级展开创新？模型制品如何沿谱系向下扩散？这些下游再加工能否重新进入核心层？这些问题共同指向开放技术系统中的一

个更一般命题：开放的深度和边界，会影响创新发生的位置、知识扩散的方向以及跨层反馈的可能性。

1.2 开放权重大模型的开放层级结构

开放权重大模型的开放性是由多个技术工件构成的层级结构，其中权重文件决定外部主体能否直接加载和复用模型；config 和分词器等文件决定模型能否被稳定适配和部署；模型卡和 README 文件影响使用者能否理解模型来源、能力边界和使用条件；训练代码、日志和数据集元数据则更接近模型知识生产过程本身。

如表 1 所示，本章的重点样本中，权重文件、模型卡、许可证元数据、config 和 tokenizer 的覆盖率较高，说明本章样本主要围绕可下载、可加载、可配置的模型制品展开。与此同时，训练/微调代码、推理代码以及训练过程深层记录的覆盖率明显较低。这一结构说明，开放权重大模型的主流开放形态并不是完整训练过程开放，而是模型制品和使用工件开放。

表 1　重点样本模型主要开放对象可得性

开放对象	T1—T3 整体	T1 核心模型	T2 重要衍生	T3 社区高影响
权重文件	93.9%	92.0%	92.3%	97.6%
模型卡	97.7%	96.0%	98.2%	99.0%
许可证元数据	84.8%	82.5%	86.0%	86.3%
config	78.3%	90.0%	76.8%	66.3%
分词器文件	75.0%	85.6%	79.6%	58.3%
推理代码	1.2%	1.8%	1.3%	0.4%
训练/微调代码	0.5%	0.8%	0.6%	0.1%
数据集元数据	0.1%	0.0%	0.0%	0.1%

为了避免把开放权重误写成完整开源，本研究进一步使用规则化开放程度指标刻画模型制品开放深度。该指标根据权重文件、config 文件等对象计算开放分数，并划分开放等级。需要强调的是，该指标衡量的是模型工件可得性和文档可得性，而不是判断模型是否满足 OSI 意义上的开源定义。

如表 2 所示，本章样本中多数模型处于 level 2 或 level 3。level 2 表示模型具备权重文件、config 和 tokenizer 等基本可加载条件；level 3 表示在此基础上具有较充分模型技术文档和较高开放分数。达到 level 4 的模型较少，level 5 在当前样本中未观察到。这进一

步说明，开放权重大模型生态的主流开放形式是开放权重与模型使用工件，而不是开放完整训练生产过程。

表 2　重点样本模型开放程度等级分布

样本层级	level 0	level 1	level 2	level 3	level 4	level 5
T1 核心模型	304（8.0%）	375（9.9%）	1654（43.6%）	1434（37.8%）	29（0.8%）	0（0.0%）
T2 重要衍生	252（7.7%）	660（20.2%）	193（5.9%）	2150（65.7%）	19（0.6%）	0（0.0%）
T3 社区高影响	79（2.4%）	1,380（41.5%）	192（5.8%）	1668（50.2%）	4（0.1%）	0（0.0%）

1.3 与区块链开放系统的机制差异

开放权重大模型与区块链都属于开放技术系统，但二者开放的对象和价值反馈机制并不相同。区块链系统通常围绕协议代码、链上状态、智能合约、客户端、应用部署和资产结算机制展开。其开放性不仅体现在代码可见和接口可组合，也体现在链上状态、交易、手续费和治理权等经济关系的公开可观察性。由此，区块链试图把技术扩散、应用增长和价值反馈纳入同一个协议化系统之中。

开放权重大模型则不同。它主要开放的是模型权重和围绕模型使用的辅助性技术制品，而不是完整训练过程和统一的经济系统。下游主体可以基于权重进行微调、适配和再分发，但这些活动带来的收益并不必然通过一个透明且自动的共同价值载体反馈给上游模型发布者或核心研发层。开放模型生态中的价值反馈更可能通过品牌声誉、云服务、硬件销售和许可证约束作等间接渠道实现。

因此，本文并非在简单比较区块链和开放权重大模型“谁更开放”，而是聚焦于二者所代表的两种不同的开放组织形态：区块链更接近协议化开放与共同价值载体结合的系统；开放权重大模型更接近模型制成品开放与平台化、间接化价值反馈结合的系统。正是这种机制差异，使大模型开放生态可能呈现出不同于区块链的创新业态。

1.4 本章的理论问题与分析路径

在上述界定基础上，本章关心的问题是当模型权重和部分制成品被开放后，技术如何沿模型谱系扩散？下游行动者在哪些层级进行再加工？核心层技术路线如何更新？下游外围创新是否能够重新反馈到核心层？这些过程又由哪些组织主体承担，并如何受到开放边

界和价值反馈结构的影响？

本章的基本判断是开放权重大模型形成了一种非对称的开放技术系统。其开放性首先表现为强大的向下扩散能力。技术通过公开声明的模型谱系进入下游，父模型的权重、配置和架构信息成为技术沿谱系流动的重要载体。与此同时，下游模型并非简单复制父模型，而是在后训练、适配、推理部署和再分发等环节形成大量外围层再加工。这种开放系统的非对称性在于下游创新机会被显著扩大，但这些机会主要集中在可接触和部署的外围层；核心层技术路线的变化，则更多表现为头部主体之间的领域级共同采用，而非明确的有向谱系传播。进一步说，现有公开数据能够较强地观察到向下扩散和外围再加工，却难以观察到下游创新稳定、清晰地被核心层采纳，这一体系尚未呈现清晰的跨层创新回流闭环。

因此，本章将依次展开三个层面的分析。第一，考察开放权重的技术扩散与创新层级，分析模型谱系承载的向下扩散、衍生模型中的外围再加工、核心层横向共同采用以及下游知识回流。第二，考察组织分工与开放模型再生产，说明正式组织、社区账号和服务型账号如何分别承担底座发布、衍生开发、部署转换和再分发功能。第三，考察开放边界、价值反馈与创新业态，解释为什么开放权重大模型会形成强扩散、强外围再加工、弱公开回流证据和平台化反馈并存的结构。

这一章由此为全文理论提供一个关键对照，开放本身并不自动产生跨层协同。开放对象决定外部主体能够进入哪些技术层级，价值反馈结构影响外部创新能否重新连接核心层，而技术层级和组织分工共同塑造开放系统中的实际创新业态。

第二章　样本构建与技术标签识别

本章的实证分析基于 Hugging Face 平台的模型生态结构化数据库。本研究先对平台模型数据进行全量抓取，再在其中构建适合研究开放权重大模型技术扩散的数据样本框，并通过大语言模型辅助，进一步识别模型之间父子谱系关系、模型技术标签和层级以及发布主体类型等核心变量。

2.1 样本选择与代表性

本研究所用的基础模型信息表由 Hugging Face 平台元数据、模型文件信息、模型卡和相关处理脚本合并而成，共包含 2638300 行模型数据。每条记录以 Hugging Face 平台

的模型名作为模型 ID，并保留发布时间、发布账号、标签、许可证、下载量、社区点赞量、文件结构、上游模型声明、模型家族、模型角色、是否官方账号、是否 LLM 以及样本层级等字段。

在全量模型样本基础上，研究首先识别大语言模型的样本框，该样本框包含 1294671 个模型。鉴于本章关注的是可观察的技术路线和模型谱系扩散，且由于长尾模型数量巨大、信息缺失较多，后续详细分析将聚焦于 T1—T3 这一三级样本框，共 10399 个模型。具体而言，T1—T3 的分层含义如下：T1 指白名单机构发布的核心大模型，共 3798 个，白名单包含了主要的大模型公司、实验室以及研究机构，名单详见附录；T2 指白名单机构发布的、基于 T1 样本框中核心大模型的衍生大模型，共 3275 个；T3 指社区高影响模型，共 3326 个。这种分层的样本框设计使本章能够同时观察核心模型、重要衍生模型和高影响社区模型之间的关系。其他模型虽然在数量上构成开放模型生态的大部分，但模型卡质量、谱系信息和技术标签覆盖不足，且长尾噪声较高，因此暂不进入本章主分析。

为说明代表性问题，下表展示了 T1—T3 样本在开放权重大模型生态中的影响力情况。如表所示，T1—T3 样本合计占模型数量的 0.80%，但贡献了 90.89%的近 30 日下载量、和 88.02%的累计下载量。因此，T1—T3 样本虽然只占模型数量的一小部分，却集中承载了开放权重大模型生态中的主要使用量和关注度。本章将 T1—T3 作为主分析对象，并不是因为它们在数量上占优，而是因为它们更集中地体现了开放大模型生态中可观察的技术路线、模型谱系和扩散关系。

表 3　T1 - T3 样本在开放权重大模型生态中的数量与影响力占比

样本层级	模型数	模型数占比	30 日下载（百万）	30 日下载占比	累计下载（十亿）	累计下载占比
T1 核心模型	3798	0.29%	715.8	37.72%	12.76	47.63%
T2 重要衍生	3275	0.25%	785.3	41.38%	7.80	29.12%
T3 社区高影响	3326	0.26%	223.8	11.79%	3.02	11.28%
T1–T3 合计	**10399**	**0.80%**	**1724.8**	**90.89%**	**23.59**	**88.02%**
T4–T5 合计	1284272	99.20%	172.9	9.11%	3.21	11.98%
LLM 样本框合计	**1294671**	**100.00%**	**1897.7**	**100.00%**	**26.80**	**100.00%**

2.2 模型谱系识别

2.2.1 识别方法

模型谱系指模型之间的衍生关系，其基本单元为有向边，记作父模型→子模型，表征子模型在训练过程中以父模型的权重或配置作为起点。谱系边的构建质量直接决定后续技术扩散分析的精确性，因此本研究采用多来源识别与证据分级策略，对谱系边按来源置信度进行分层处理。

考虑到 Hugging Face 平台上模型继承关系的声明方式存在结构性差异，本研究从三个来源（配置文件声明、模型卡元数据声明以及模型卡正文文本启发式识别）识别父子模型关系，并赋予差异化的置信度。具体方式如下：

①配置文件声明，置信度为 1。模型发布者在发布权重时，通常于 config.json 文件 中通过 base_models 等字段显式声明所继承的父模型。此类字段由元数据直接直接返回，属于机器可读的结构化声明，证据强度最高。

②模型卡元数据声明，置信度为 1。部分模型在模型卡顶部的 YAML 信息中声明 base_models 等字段。此类声明虽不在配置文件内，但同样属于发布者通过结构化字段主动填写的元数据，证据强度与来源一等同。

③模型卡正文文本启发式识别，置信度为 0.80。对于未在上述结构化字段中声明继承关系的模型，本研究对模型卡正文中出现的显式关系短语进行启发式匹配，识别诸如 "fine-tuned version of"、"based on" 等表述所指向的父模型。此类识别依赖于大语言模型对于自然语言文本的识别与匹配，有可能存在语义歧义与误识别风险，证据强度低于前两类来源。

当同一对父子关系被多个来源同时识别时，本研究保留置信度最高的记录以消除冗余。三个来源合并后，全量候选谱系边共计 880758 条。在候选边的基础上，本研究依循以下步骤构造分析样本：

①子模型范围界定。本研究聚焦于 T1 – T3 样本框内的模型扩散过程，因此仅保留子模型属于 T1 – T3 层级的谱系边，得到候选边 6230 条。此举排除了子模型为 T0（非大语言模型）、T4（社区长尾）、T5（社区原创）或层级信息缺失的边，以确保分析对象聚焦于具有实质性扩散意义的核心模型群体。

②证据来源分流。 候选边按其识别来源分为两类，分别赋予不同的分析地位。其中，可靠谱系边共 5840 条，来源于配置文件声明或模型卡元数据声明，系模型发布者通过结构化字段显式声明的继承关系。此类边作为本章分析模型工件向下扩散的强结构性证据，进入主分析。文本启发式边共 390 条，来源于模型卡正文的文本模式匹配，可能包含误识

别或语义歧义，此类边不进入任何主指标，仅用于稳健性分析的参考。

2.2.2 关系类型的推断

谱系边的关系类型可用于表征子模型对父模型的加工方式。经过整理后，本研究将其分为六类：微调（finetune）、适配（adapter）、量化（quantized）、合并（merge）、蒸馏（distill）与 未明确（unknown）。需要指出的是，上述关系类型并非从 API 字段中直接读取，而是基于子模型的功能角色标签、模型标识符及关系提示词进行关键词匹配推断。例如，当上述字段中出现量化格式标识时，判定为量化（quantized）；出现适配器技术标识时，判定为 适配（adapter）。未能匹配任何预设关键词的关系则标记为未明确（unknown）。这一推断方式的优势在于覆盖面广，但存在一定的分类模糊性，尤其在未明确类别中可能混杂未识别的衍生关系。

不同的谱系关系类型代表了不同形式的下游再加工。微调（finetune）通常指在父模型权重基础上，使用新的任务数据、领域数据或指令数据继续训练，使模型更适合特定任务或场景。适配（adapter）则是在尽量保留原模型主体参数的情况下，加入 LoRA、adapter 等轻量模块，以较低成本改变模型行为。量化（quantized）主要是将模型权重转换为更低精度或特定推理格式，以降低显存、存储和部署成本，因此更接近部署侧扩散机制。合并（merge）是将多个模型或多个微调结果进行权重层面的组合，可能形成新的技术组合，但这种重组是否具有实质创新仍需具体案例验证。蒸馏（distill）则是让较小模型或新模型学习上游模型的输出、偏好或推理行为，是一种能力和行为层面的再表达，而不一定继承完整架构或训练过程。

2.2.3 谱系边的字段构成

经过上述构建与流程，每条可靠谱系边最终记录以下字段：边标识符、父模型标识、子模型标识、关系类型、父模型发布账号、子模型发布账号、实体级跨边界状态、父模型创建时间、子模型创建时间、父模型家族、子模型家族以及父模型数量。

2.3 技术路线识别与标签抽取

2.3.1 数据来源

本研究对于技术路线识别主要依赖模型 README 文件以及模型卡。在开放权重大模型生态中，许多关键技术并不总是出现在结构化元数据中，而是写在模型卡的训练说明、模型结构、使用方式以及致谢和引用部分。针对 T1—T3 的 10399 个模型，研究构建了大

语言模型技术路线识别输入清单，成功匹配 9655 个 README 文件，覆盖率为 92.8%。未匹配 README 文件的模型共有 744 个，缺失主要来自两类情形，一类是受限模型，获取 README 文件需要发布方人工授权，高度集中于 Google 和 Meta 这两家海外公司；另一类是 Hugging Face 上本身没有模型卡的模型。对于这些模型，API 元数据通常仍能提供模型 ID、发布时间、标签、部分配置代码和文件列表，但不能提供 README 正文，因此无法替代 README 中关于技术路线的文字说明。

2.3.2 技术标签抽取与合并

鉴于 Hugging Face 平台上模型技术信息的呈现方式高度异质，既有结构化的配置字段，也有大段非结构化的自然语言描述，因此本研究采用大语言模型辅助语义识别与技术分类体系文档相结合的策略，并以证据可追溯性为核心对抽取结果进行分级控制。

本研究自行构建了面向大语言模型的技术分类体系文档（taxonomy v2），作为技术标签抽取的参照标准。该体系包含 15 个大类、48 个子类、276 项技术条目，覆盖模型权重内生架构、分词与训练目标、数据与训练方法、后训练与对齐、推理时方法、压缩与合并、多模态、服务系统、安全评测与报告等技术领域。每项技术均附有检测信号词、判定规则与排除条件，并规定了四档证据声明等级：明确声明、多信号一致、仅弱信号关联与证据不足。同时，体系明确了证据来源的优先级：同版本官方文档优先于配置文件，配置文件优先于论文复现，论文复现优先于第三方描述。这一设计确保了技术标签的判定具有一致性与可审计性。

技术标签的抽取由两个互补流程组合完成。主流程以本地下载的模型 README 文件以及模型卡为输入，调用 DeepSeek-V4-Flash 模型进行结构化语义理解和抽取。首先，将 README 文件以及模型卡按 Markdown 章节切分，并剔除目录、许可证、联系方式等不携带技术证据的章节。其次，对模型的配置文件进行确定性解析，提取可由配置字段直接判定的事实（如位置编码类型、注意力机制实现等）。此类事实不依赖语义判断，作为大模型抽取的补充证据。在此基础上，根据章节文本的信号匹配与章节主题的类别关联，从技术分类体系中检索相关技术条目，构建精简的候选集合。随后，将章节文本、配置事实与候选技术条目一同输入大模型，要求其判断模型是否采用了某项技术。大模型的输出须为结构化断言，每条断言包含技术标识、证据关系、判定结果、证据等级、作用对象、原文引用与判断理由等字段。

为防止大模型生成缺乏证据支撑的标签，本研究设置了多重约束。第一，大模型只能使用技术分类体系中已定义的技术标识。第二，每条断言必须给出原文引用，且该引用必须是输入文本的连续字面子串。第三，程序自动校验原文引用是否能在输入文本中定位。

第四，并非所有识别结果都进入正向标签，仅判定为“已接受”的断言才进入后续分析。对于判定结果存在争议或风险较高的标签，进一步路由至 DeepSeek-V4-Pro 模型进行复核。复核采用分级策略：低风险标签直接采纳，中风险标签由 Pro 模型进行快速裁定，高风险标签则启用思维链模式进行深入判断。

主流程完成后，仍有部分模型未获得任何核心技术标签。为降低覆盖缺口，本研究对这类模型启动补救流程。首先，基于模型名称、模型角色、元数据与文件名中的高精度线索进行大模型补充识别。例如，模型名包含量化格式标识（如 AWQ、GPTQ、FP8）则判定为对应量化技术；模型角色标注为“量化产物”则判定为量化标签；模型名包含 LoRA、QLoRA 等标识则判定为适配器技术。同时，对配置文件进行确定性解析，提取可直接判定的事实。

技术标签抽取完成后，本文并不直接使用全部识别结果，而是根据证据强度和分析用途逐步收紧口径。第一步是审计记录口径，只要模型在技术抽取或补救流程中留下过记录，就计入该口径，共有 8026 个模型。但这一口径包含“已接受”、“未解决”和“已拒绝”等不同状态，不能直接表示模型真实采用了某项技术。第二步是已接受标签口径，只保留至少有一条被判定为“已接受”技术标签的模型，共有 7186 个模型。第三步是本文主分析使用的核心已接受技术标签口径，在已接受标签中进一步排除“基础模型关系”、“模型卡”、“文件格式”“简单分词”等辅助性或格式性标签，只保留能够代表实质技术路线的标签。按这一口径，共有 7074 个模型拥有核心技术标签，对应 11373 行“模型—技术标签”记录。

表 4　技术标签覆盖率口径

口径	模型数	覆盖率	是否用于主分析	说明
审计记录口径	8026	83.1%	否	包含 unresolved / rejected 等审计记录
已接受标签口径	7186	74.4%	可作宽松口径	至少有一条 accepted 标签
核心已接受技术标签口径	7074	73.3%	是	排除辅助和关系标签，适合技术路线分析

2.3.3 技术标签的层级划分

为了比较不同技术在开放模型生态中的扩散方式，本文将技术标签进一步映射到五个经验技术层级。该分类体系基于 taxonomy v2，共包含 276 个技术项，并通过技术层级映射表完成归类。五个层级从核心架构到应用系统，分别对应模型技术系统中的不同位置。

层级越靠前，越接近模型结构和训练生产；层级越靠后，越接近后训练、部署和应用组合。

表 5　技术标签的经验层级划分

技术层级	中文名称	含义	代表性技术
L1	核心架构技术	直接塑造模型结构或基础能力的技术，通常与模型架构、注意力机制、位置编码、归一化、激活函数等有关	注意力机制、位置编码、MoE 架构、归一化、激活函数、tokenizer、训练目标
L2	训练生产技术	与模型预训练和生产过程有关的技术，主要涉及数据、训练目标、优化方法、分布式训练和训练流程	预训练数据、训练目标、分布式训练、优化器、训练 pipeline、数据治理
L3	后训练与适配技术	在基础模型之后进行的对齐、微调、偏好优化、轻量适配和模型组合技术	SFT、LoRA、QLoRA、RLHF、DPO、GRPO、蒸馏、模型合并、指令微调、偏好对齐
L4	推理部署技术	与模型压缩、推理加速、服务部署和运行时优化有关的技术，主要影响模型如何被部署和使用	量化、KV cache、投机解码、vLLM、llama.cpp、服务运行时
L5	应用系统技术	将模型嵌入具体应用系统时使用的技术，通常涉及检索、工具调用、智能体和工作流编排	RAG、工具使用、函数调用、Agent、工作流编排、检索系统

2.4 技术层级与谱系技术事件构造

在获得模型谱系关系以及技术标签网络以及层级后，为了分析技术如何沿模型谱系扩散，研究将清洗后的谱系边与模型技术标签连接，构造边 × 技术的谱系技术事件。每条事件包含父模型是否具有某技术、子模型是否具有该技术、子模型是否声明父模型、关系类型、技术层级、时序状态和跨账号信息。

由于公开模型卡和 README 的披露具有选择性，因此谱系技术事件的分类不是简单的"有/无"二分，而是基于已知端、未知端和不确定性方向的差异化判断。下表列出五类传播分类的判断口径：（1）技术延续，表示父子模型均声明采用有该技术，是较强的技术延续证据；（2）候选携带，表示父模型有该技术、子模型未显式重复声明，但由于子模型声明继承父模型，该技术可能随权重或配置被携带到子模型；（3）新增采用，表示

子模型有该技术、父模型未观察到该技术，代表子模型中的新增显式采用；（4）多源重组，表示多父谱系或出现多重候选技术组合；（5）证据不足。特别值得注意的是，区分“候选携带”和“新增采用”这两类容易混淆的状态。候选携带和新增采用的状态是不对称的：前者的已知端在父模型，不确定的是子模型是否保留；后者的已知端在子模型，不确定的是父模型是否真的没有。从理论意义上说，候选携带更多呈现“可能随模型工件向下流动的技术规模”，而新增采用则更多呈现“子模型主动声明新增了什么技术”

表 6　谱系技术事件传播分类的判断口径

传播分类	事实含义	需推断部分	不确定性方向
技术延续	父模型显式声明该技术，子模型也显式声明该技术	两者实现是否真的"同一技术"	较小
候选携带	父模型显式声明该技术，子模型未重复声明	子模型是否实际保留了该技术	依赖子模型继承了父的权重或配置这一假设
新增采用	子模型显式声明该技术，父模型未观察到	父模型是否真的没有该技术	依赖父模型未显示声明则父模型未使用该技术这一假设
多源重组	多父或 merge 关系下，子模型显式出现某技术	该技术是否真的来自多个父模型的组合	多父端推子端
证据不足	证据不足以判定上述任一类	无法判定	—

此处有一个分层分流，根据技术层级处理，对于父模型有技术而子模型未显式重复声明的情况，若是技术标签层级为 L1、L2 等更可能嵌入权重或配置的技术，可进入候选携带；对于 L3 后训练适配技术，需要结合关系层级判断；对于 L4/L5 推理部署和应用系统技术，由于它们不一定随权重传播，通常归为证据不足。这一处理规则的核心含义是：同一分类标签在不同技术层级上的可信度不同。对于 L1，候选携带是合理的推断方向；对于 L4/L5，候选携带在机制上不成立，必须降级为证据不足。因此，本研究在后续报告结果时同时给出两个口径：主样本口径（含候选携带，呈现通道规模）和敏感性口径（去除候选携带，呈现子模型主动加工的真实信号）。两个口径都不能单独代表完整图景——前者会高估 L1 的技术延续，后者会高估 L3/L4 的技术新增。只有同时报告两者，才能既不低估谱系携带的可能规模，也不把未经验证的携带写成确认传播。

表 7　候选携带与新增采用的不对称对比

维度	候选携带	新增采用
已知端	父模型显式声明该技术	子模型显式声明该技术
推断端	子模型是否保留了该技术	父模型是否真的没有该技术

对 L1 的影响	严重高估技术延续	严重低估父模型实际技术存量
对 L3 的影响	候选携带相对少	信号相对真实、L3 披露更主动
对 L4 的影响	按规则不进入候选携带、L4 不随权重传播	信号相对真实、L4 披露主要由量化分发账号主动声明
正确使用方式	呈现可能随工件向下流动的技术规模	呈现子模型主动声明新增了何种技术

2.5 主样本与统计单位

经过数据清洗过滤、可靠边筛选和实质性技术过滤后，本研究所用的数据样本具体如下表所示，技术扩散主样本为 6218 条“谱系边×技术事件”，对应 276 项实质性技术和 4447 次的技术跨组织扩散。

表 8　论文主样本与统计单位

指标	数值	统计单位	说明
T1 模型数	3798	模型	头部/核心模型层
T2 模型数	3275	模型	重要衍生模型层
T3 模型数	3326	模型	高影响社区模型层
可靠谱系边	5840	谱系边	强结构性证据
文本启发式边	390	谱系边	不进入主分析
主样本	6218	边×技术事件	严格快照主样本
跨账号技术事件	4447	边×技术事件	父子发布账号不同的技术事件

第三章　开放权重的技术扩散与创新层级

3.1 模型谱系作为向下扩散通道

开放权重大模型首先通过模型谱系形成可直接观察的技术扩散通道。与论文引用、专利引用或一般技术术语共现不同，开放模型扩散的对象不是抽象知识声明，而是可以被直接加载、继承和改造的模型制品。父模型的权重、配置文件、分词文件和架构信息为子模型提供了可复用基础。因而，模型谱系边不仅记录模型之间的发布关系，也记录技术本身如何跨模型层级流动。

在本研究中，谱系扩散主要依据可靠谱系边识别。所谓可靠谱系边，是指来源于配置文件声明或模型卡元数据声明的父子模型关系。相比模型卡正文中的文本启发式识别，这类边由发布者通过结构化字段显式声明，因而更适合作为分析开放模型制品向下扩散的主证据。需要强调的是，谱系边是通过所记录的是父模型与子模型之间的继承或衍生关系，进而对技术扩散路径进行刻画，而不是对技术因果影响的完整还原。

从全量可靠谱系边看，T1 和 T2 是开放模型向下扩散的主要来源。T1→T2 有 1224 条可靠谱系边，T1→T3 有 1139 条，T2→T2 有 1245 条，T2→T3 有 833 条。相比之下，T1→T1 显式谱系边仅有 3 条。这一结构说明，在当前公开可观察的模型谱系中，最清晰的扩散形态不是头部核心模型之间的直接父子继承，而是核心模型和重要衍生模型向下游模型层级的扩散。

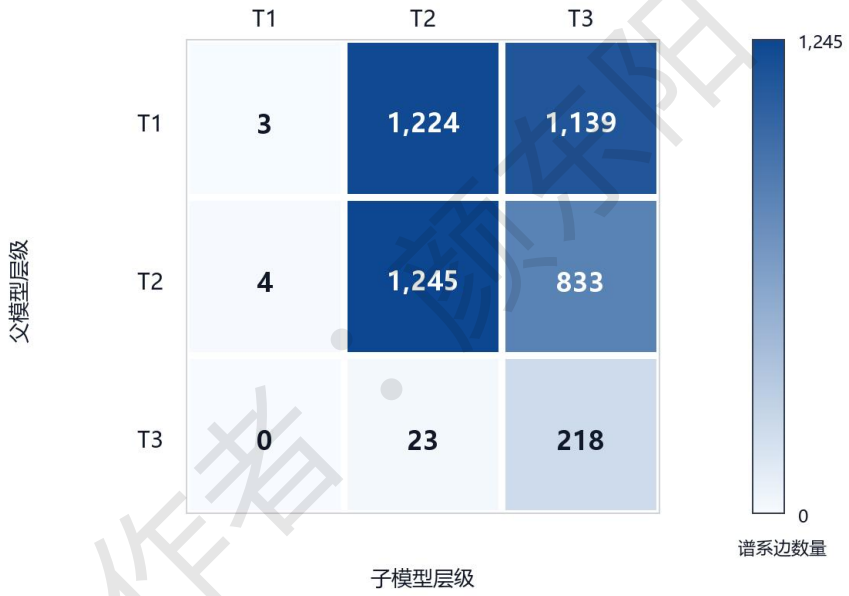

图 1　T1/T2/T3 可靠谱系边流向矩阵

谱系技术事件进一步说明，T1—T3 之间的技术扩散并不是均匀分布的，而是明显集中在由核心层和重要衍生层流向下游的路径上。这里需要区分两类统计单位：谱系边表示模型之间是否存在父子关系，而“边 × 技术项”的谱系技术事件表示在一条父子模型边上观察到某项技术的延续、候选携带、新增采用或其他分类。因此，技术事件反映的是模型谱系关系中可观察技术内容的流动强度。

从层级流向看，T1→T2 和 T1→T3 分别包含 1775 条和 1513 条谱系技术事件。这说明头部核心模型在开放生态中不仅作为单个模型被下载和使用，也作为后续模型开发的技术来源被持续调用。T1→T2 的 1775 条事件分布在 854 条不同谱系边和 104 项技术上，表明核心模型向重要衍生模型的扩散并非只围绕少数单一技术，而是覆盖较宽的技术组合。

T1→T3 的 1513 条事件则全部发生在跨发布账号关系中，说明当核心模型进入高影响社区模型层时，模型制品已经跨越原发布账号，成为社区再加工和再分发的基础。

T2 也不是简单的中转层。T2→T2 包含 1370 条谱系技术事件，涉及 787 条不同谱系边和 90 项技术，说明重要衍生模型之间也存在大量连续开发、版本迭代和再衍生关系。T2→T3 的 969 条事件则显示，重要衍生模型进一步进入社区高影响模型，形成从“头部核心模型—重要衍生模型—社区高影响模型”的多级扩散链条。换言之，开放权重扩散不是一次性的 T1 向下释放，而是在 T2 层继续被加工、延续和重新分发。

表 9　主流 tier flow 技术事件摘要

层级流向	事件数	不同谱系边	不同技术项	跨账号比例	量化关系占比
T1→T2	1775	854	104	52.73%	41.24%
T1→T3	1513	705	79	100.00%	66.95%
T2→T2	1370	787	90	44.38%	49.71%
T2→T3	969	519	68	100.00%	71.41%

跨账号比例进一步揭示了不同扩散路径的边界差异。T1→T2 和 T2→T2 的跨账号比例分别为 52.73% 和 44.38%，说明这两类流动中既包含跨账号扩散，也包含同一发布账号内部的版本更新、系列扩展或衍生开发。相比之下，T1→T3 和 T2→T3 的跨账号比例均为 100.00%，表明一旦技术事件进入 T3 社区高影响模型层，几乎完全表现为跨发布账号的模型再加工。这一结果支持开放模型生态中模型制品跨账号流动的判断。

进一步看，不同模型层级流向中的技术扩散并不只体现为谱系边数量差异，也体现为关系类型差异。为刻画开放模型沿谱系进入下游时的具体加工方式，本文特别关注各层级流向中的量化关系占比。本文所说的量化关系，指谱系边的关系类型被识别为“quantized”的父子模型关系。其典型情形是子模型在继承父模型权重基础上，将权重转换为更低精度或特定推理格式，例如 GPTQ、AWQ、GGUF、FP8 等，以降低存储或推理成本，主要表示模型制品在部署侧的压缩、格式转换以及部署适配。

结果显示，T1→T3 和 T2→T3 中量化关系占比分别达到 66.95% 和 71.41%，显著高于 T1→T2 的 41.24% 和 T2→T2 的 49.71%。这说明，核心模型和重要衍生模型进入社区高影响层时，常常伴随以部署适配为核心的量化技术继承和创新。换言之，开放权重的向下扩散并不简单地表现为核心架构创新的复制，而很大程度上聚焦于在部署适配过程中对于模型制品的压缩和格式转换等外围层的技术创新。这一类技术通常并不改变模型的基础架构，却显著降低模型使用门槛，使开放权重模型能够在更多硬件环境、推理框架和应用场景中被使用。如上表所示，此类技术创新应被视为开放模型向下扩散中的关键创新路径。

总体而言，模型谱系结果支持的是一个“开放制品向下扩散”的结构性判断：T1 和 T2 模型构成开放权重大模型生态中最重要的上游来源，T2 层继续承担再衍生和重新分发功能，T3 层则集中呈现跨发布账号的下游扩散。这个结果说明，开放权重确实建立了强大的下游传播通道；但它的证据边界也必须明确。首先，谱系边证明的是模型制品沿公开继承关系发生可观察流动，并不能直接证明下游创新已经反馈到 T1 核心层。其次，T1→T1 显式谱系边极少，因此本章不能基于谱系数据构造 T1 核心模型之间的有向传播网络。换言之，本节所能确认的是公开模型谱系中的强向下扩散通道，而不是核心层回流，也不是头部主体之间的有向技术传播。至于这些扩散通道中具体携带了哪些层级的技术、下游主体又在哪些层级进行主动再加工，则需要在下一节进一步分析。

3.2 外围层创新：适配、量化与再分发

上一节说明，开放权重通过公开模型谱系形成了强向下扩散通道。但向下扩散并不意味着下游模型只是被动复制上游模型。开放权重的更重要影响在于，外部主体能够在可接触、可加载和可部署的模型制品基础上进行再加工。因而，在确认模型谱系构成向下扩散通道之后，还需要进一步分析：这些通道中流动的技术分别处于哪些层级？下游模型主动声明的技术活动又主要发生在哪里？

首先，主流层级流向携带的技术层级并不相同。T1、T2 向下游扩散时，L1 核心架构技术占据重要位置，但 L3 后训练适配和 L4 推理部署技术也随衍生关系进入下游。这说明，开放模型谱系传递的不只是基础架构，还包括围绕模型使用、训练后加工和部署形成的技术组合。

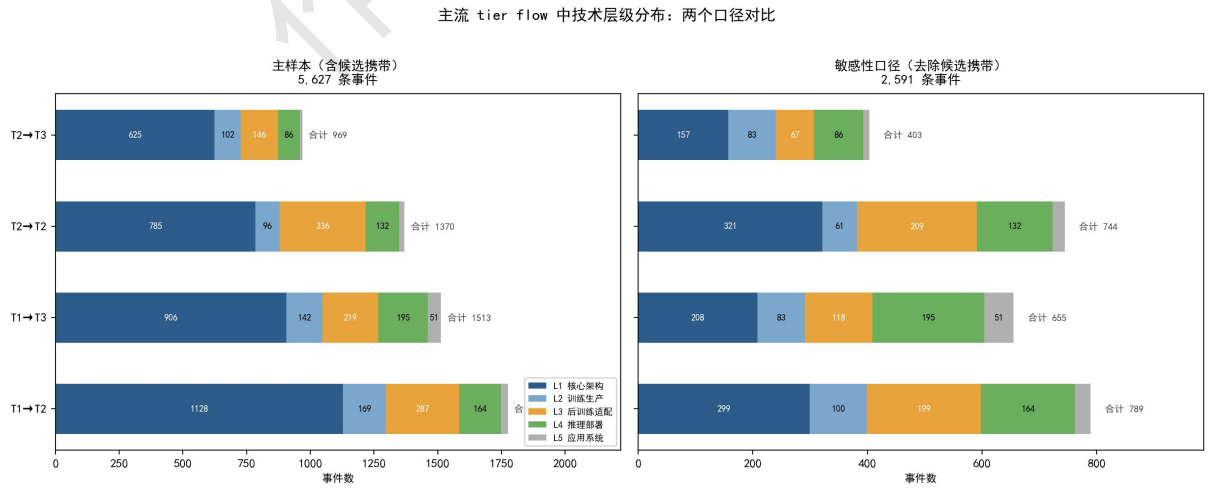

图 2 主流层级流向中的技术层级分布

图 2 进一步采用两个口径展示层级流向中的技术层级分布。主样本口径包含候选携带，用于呈现谱系通道中“可能随模型工件向下流动”的技术规模。在这一口径下，L1 核心架构技术占据较高比例，说明父模型中声明的基础架构和模型结构相关技术，可能通过权重和配置进入下游模型。但这一口径也会高估 L1 技术延续，因为父模型声明某项技术、子模型未重复披露时，研究只能将其处理为候选携带，而不能确认子模型实际保留了该技术。敏感性口径则去除候选携带，主要保留子模型显式声明的技术采用信号，用于观察下游模型主动披露了新增哪些技术创新。在这一口径下，L1 占比明显下降，L3 后训练适配和 L4 推理部署的占比上升，说明下游模型主动声明的技术活动更多集中在后训练、适配、量化、推理部署和格式转换等层级。因此，两个口径需要同时报告。

进一步看，技术层级与传播分类之间的交叉结果也显示，下游再加工主要集中在外围层。由于该图采用含候选携带的主样本口径，因此它同时呈现了可能随模型工件流动的技术规模，以及子模型端显式披露的技术采用信号。从结果看，L1 核心架构技术中候选携带占比最高，达到 69.1%；与 L1 不同，L3 后训练与适配技术、L4 推理部署技术中，“新增采用”的占比更高。L3 中新增采用占 50.1%，L4 中新增采用占 72.9%。这表明，在子模型显式声明的技术活动中，后训练、适配、量化、推理部署和格式转换等外围层活动更为突出。换言之，开放权重并不只是把核心架构向下传递，也为下游主体围绕模型制品进行后训练和部署侧再加工提供了空间。

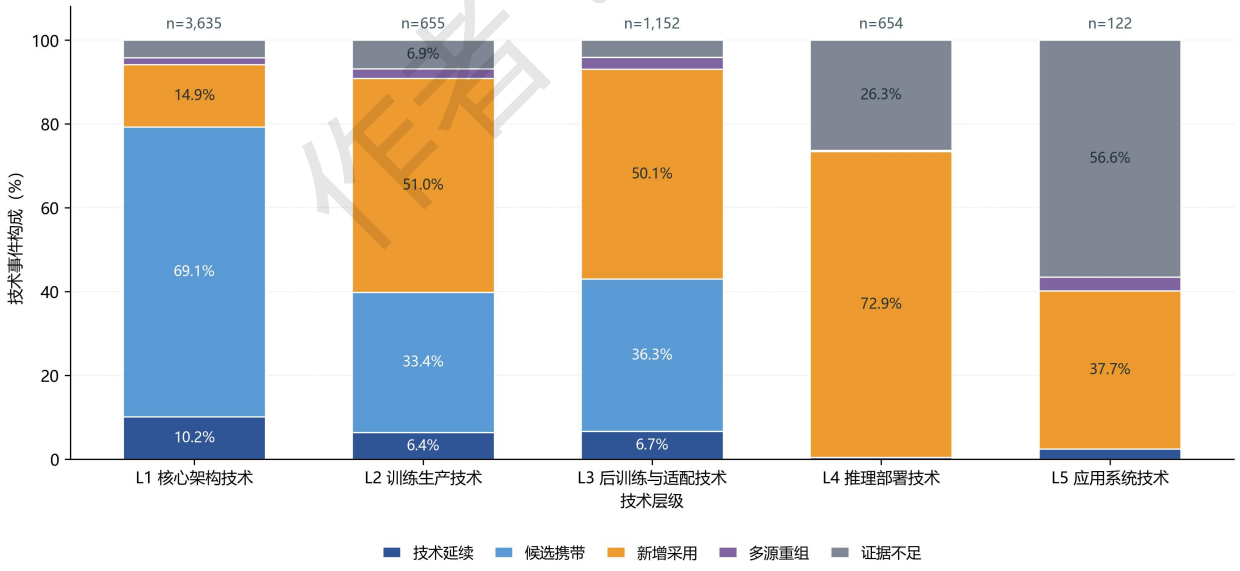

图 3 技术层级与传播分类分布

图 4 从谱系关系类型角度进一步说明下游再加工的具体方式。不同关系类型对应的是子模型主动声明对父模型的加工方式，而不是单纯的技术标签共现。结果显示，量化关系

在多个技术层级中都占据较高事件数，尤其在 L1 核心架构技术中有大量候选携带事件，在 L4 推理部署技术中也较为突出。这说明，量化类子模型往往以既有模型权重为基础，在继承父模型结构和配置的同时，进行低精度转换、格式适配和部署侧再分发。

同时，微调、适配、合并和蒸馏也呈现出不同的层级特征。微调事件既涉及 L1 核心架构技术的候选携带，也涉及 L3 后训练与适配技术；适配关系更集中于 L3；合并关系则更多体现为既有模型或微调结果之间的组合；蒸馏事件规模较小，但同样主要围绕模型能力和行为的再表达展开。由此可见，衍生模型的再加工不是单一机制，而是由微调、适配、量化、合并、蒸馏等多种关系共同构成。

不过，图 4 的解释必须保留口径边界。该图未剔除候选携带，因此反映的是主样本中“关系类型 × 技术层级”的技术事件分布。尤其是 L1 中的大量事件，很可能包含父模型技术随权重和配置进入下游的候选携带，而不能直接解释为下游主体在核心架构层进行了主动创新。

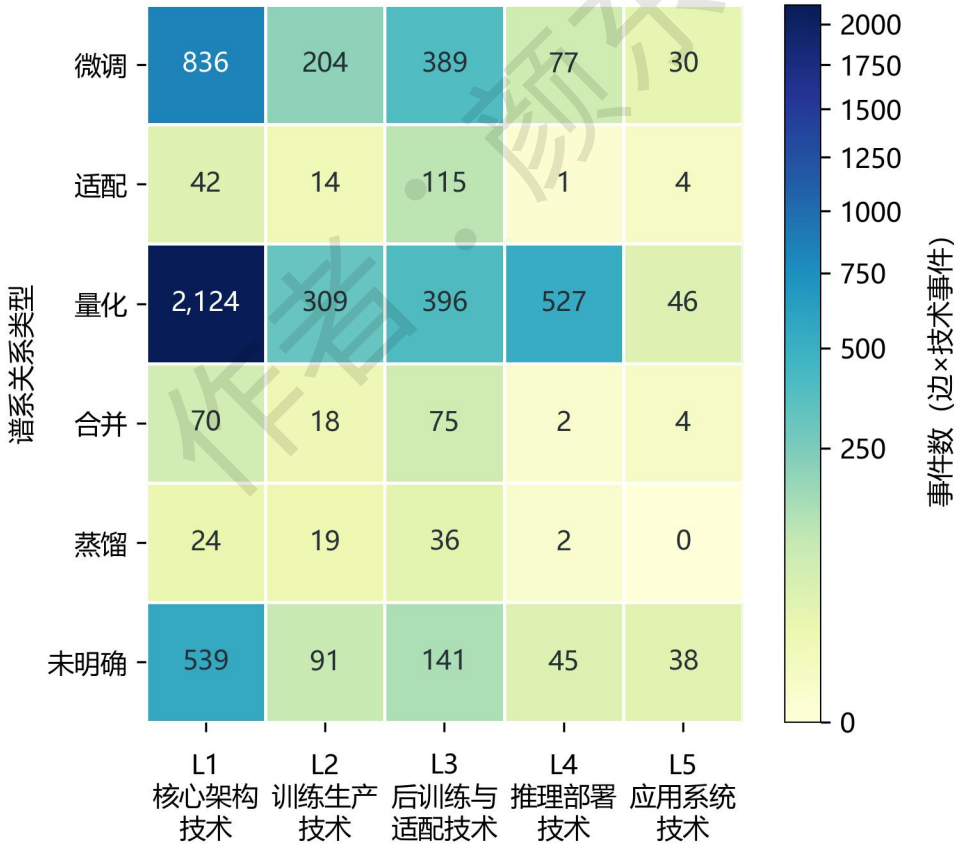

图 4 谱系关系类型与技术层级分布

图 5 则把传播分类重新放回主流层级流向中，说明不同扩散路径中的证据结构。T1→T2、T1→T3、T2→T2 和 T2→T3 四类主流流向中，候选携带都占据较大比例，分别为 986、858、626 和 566 条。这说明，开放模型向下扩散中有相当一部分技术事件来自父模型端的显式声明和子模型端的继承关系，反映的是模型工件可能携带的技术规模。

与此同时，新增采用在各层级流向中也占有重要位置。T1→T2、T1→T3、T2→T2 和 T2→T3 中新增采用分别为 421、369、461 和 295 条。这说明，下游模型并非只是被动继承父模型技术，而是在继承开放制品的基础上继续加入新的后训练、适配、量化、部署或组合方式。尤其是 T2→T2 和 T2→T3 中新增采用数量较高，进一步表明 T2 层不是简单中转层，而是在开放模型再衍生和再分发过程中承担了重要加工功能。

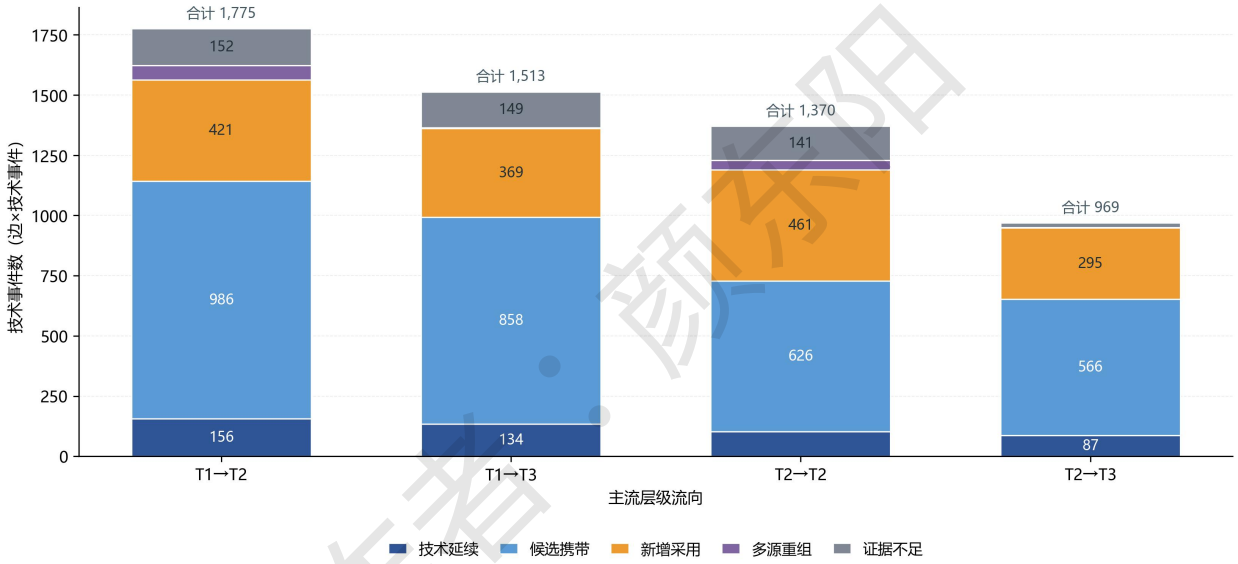

图 5 主流层级流向中的传播分类构成

综合考虑，可以得到一个较稳妥的判断：开放权重确实扩大了下游主体围绕模型制品进行再加工的机会，但这种再加工主要集中在外围层。L1 核心架构技术更多表现为候选携带，说明其可能随模型工件进入下游；而 L3 后训练与适配、L4 推理部署中的新增采用更突出，说明下游主体主动披露的技术活动更多发生在后训练、适配、量化、部署和再分发环节。因此，衍生模型不是上游模型的简单复制品，但其创新也不能被直接等同于核心架构突破。开放权重释放的是强外围再加工能力，而不是对核心知识生产过程的完全开放。

3.3 核心层的非谱系式共同采用：弱连接广泛、强连接内聚

3.3.1 从谱系扩散转向共同采用网络

前文基于可靠谱系边说明，开放权重模型在生态中形成了较强的向下扩散通道：T1 和 T2 模型主要作为上游来源，T2 和 T3 模型则在继承、改写、压缩和部署过程中承担了大量外围再加工功能。由此产生的进一步问题是：如果可观察的技术扩散主要发生在向下游流动的谱系边中，那么 T1 层内部是否也存在技术路线变化？如果存在，这种变化应当被理解为头部模型之间的有向传播，还是另一种更弱、更间接的共同采用现象？

现有数据并不支持将 T1 层写成一个可观察的有向扩散网络。全量可靠谱系边中，T1→T1 显式谱系边仅有 3 条，难以支撑关于头部模型之间“谁影响谁”的传播路径分析。因此，本节不构造 T1→T1 的有向谱系网络，而是将分析对象转向模型级技术共同采用网络。也就是说，本节关注的不是模型之间是否存在父子继承关系，而是不同 T1 模型是否在技术标签组合上出现重叠，以及这种重叠在模型家族之间如何分布。

具体而言，本节以 T1 主力模型为节点，以模型之间共享的实质技术标签为无向边，构造模型级共同采用网络。为避免技术标签过少导致偶然重叠，只有技术标签数不少于 3 项的 T1 模型进入候选节点集合，共得到 144 个候选模型。两个模型之间若共享技术标签，则可形成共同采用关系。为区分浅层重叠与较高强度重叠，本节进一步将边划分为两类：弱连接指两个模型恰好共享 2 项技术标签，强连接指两个模型共享 3 项及以上技术标签。

需要强调的是，共同采用网络中的边只表示两个模型的技术标签集合存在重叠，不表示因果影响。由于 T1 模型之间缺少足够的显式父子关系，模型级共同采用网络只能用于描述头部模型之间的技术组合相似性，不能用于推断某一模型家族向另一模型家族发生了直接技术转移。换言之，本节的分析边界是“非谱系式共同采用”，而不是“谱系式技术扩散”。这一转换有助于更稳妥地讨论 T1 层的技术变化：T1 层并非完全静止，但其可观察变化主要表现为模型层和家族层的技术组合重叠，而不是可识别方向的传播链条。

图 X 展示了 T1 模型级技术组合重叠网络。图中节点代表 T1 模型，边代表模型对之间共享技术标签；不同边强度对应不同共享技术数量。该图的目的不是寻找扩散源头，而是观察共同采用关系在模型家族内部和家族之间的分布结构。后续分析将进一步说明，T1 层共同采用并不是一个均质网络，而是呈现出明显的强弱分层：弱连接更广泛、更容易跨越模型家族；强连接则显著集中在同家族内部。

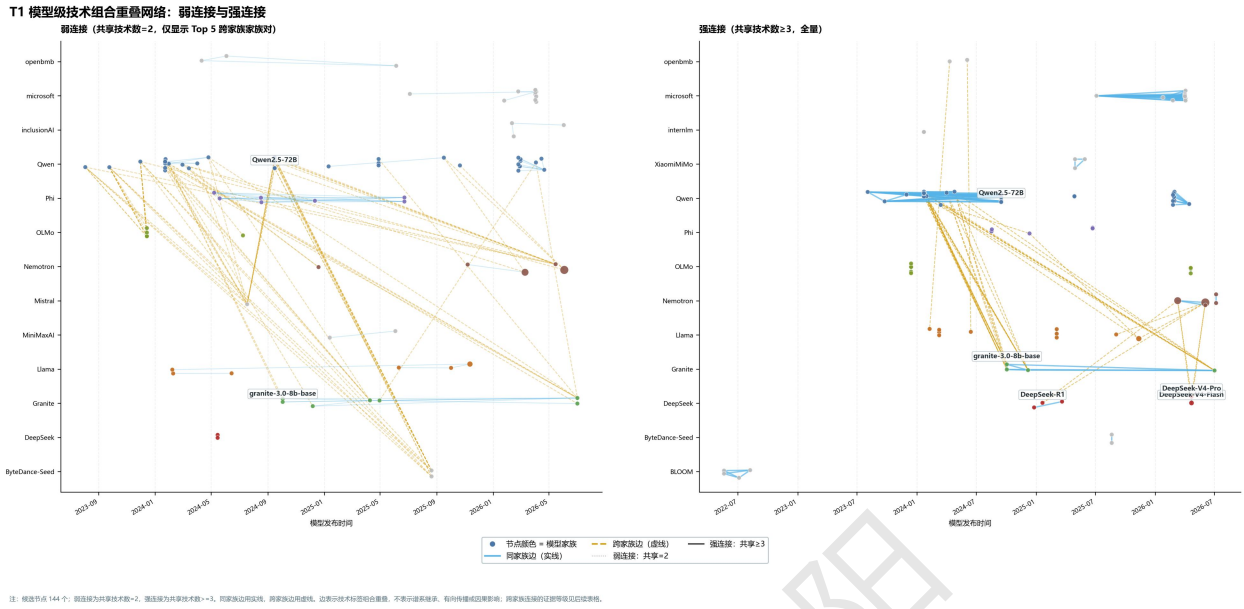

图 6

3.3.2 弱连接广泛，强连接内聚

模型级共同采用网络显示，T1 核心模型之间确实存在非谱系式技术组合重叠，但这种重叠并不是均匀分布的。根据共享技术数量的不同，弱连接和强连接呈现出截然不同的结构特征。弱连接覆盖面更广，并更多跨越模型家族；强连接数量较少，但更集中于同一家族内部。这一结果说明，T1 层共同采用的主要形态并不是头部模型之间普遍发生高强度技术路线重合，而是一种“弱连接广泛、强连接内聚”的分层结构。

从弱连接看，恰好共享 2 项技术标签的模型对共有 489 条边，涉及 130 个模型、21 个模型家族和 65 个家族对。其中，跨家族边达到 398 条，占弱连接总数的 81.4%；同家族边为 91 条，占 18.6%。这说明，在较低共享阈值下，不同 T1 模型家族之间存在相当广泛的技术标签交集。换言之，头部模型之间并非彼此隔绝，而是在若干架构、训练、后训练或部署标签上形成了广泛但较浅的共同采用关系。

表 10　弱连接 vs 强连接结构对比表

	弱连接（共享=2）	强连接（共享≥3）
边数	489	241
涉及模型数	130	92
涉及家族数	21	13
涉及家族对数	65	19
同家族边数	91	200
跨家族边数	398	41

跨家族边占比	81.4%	17.0%
平均共享技术数	2.00	3.04
最大共享技术数	2	13

但是，弱连接的广泛性不能直接解释为强技术路线趋同，更不能解释为有向扩散。由于弱连接只要求模型对共享 2 项技术标签，它更容易捕捉到通用技术范式、常见披露标签或基础建模选择带来的低强度重叠。这类边能够说明不同家族之间存在可观察的技术组合交集，却不足以证明它们共享了完整的具体技术路线。因而，弱连接更适合作为 T1 层共同采用广度的描述性证据，而不是具体路线传播的证据。

与此相对，强连接表现出明显的内聚性。共享 3 项及以上技术标签的模型对共有 241 条边，涉及 92 个模型、13 个模型家族和 19 个家族对。强连接中的同家族边达到 200 条，占 83.0%；跨家族边仅 41 条，占 17.0%。这意味着，当共享技术数量门槛提高后，模型之间的技术组合重叠迅速向同家族内部收缩。强连接主要发生在同系列模型之间，可能反映同一模型家族内部的架构复用、版本延展、训练与披露模板相似，以及同一发布主体对技术路线的连续使用。

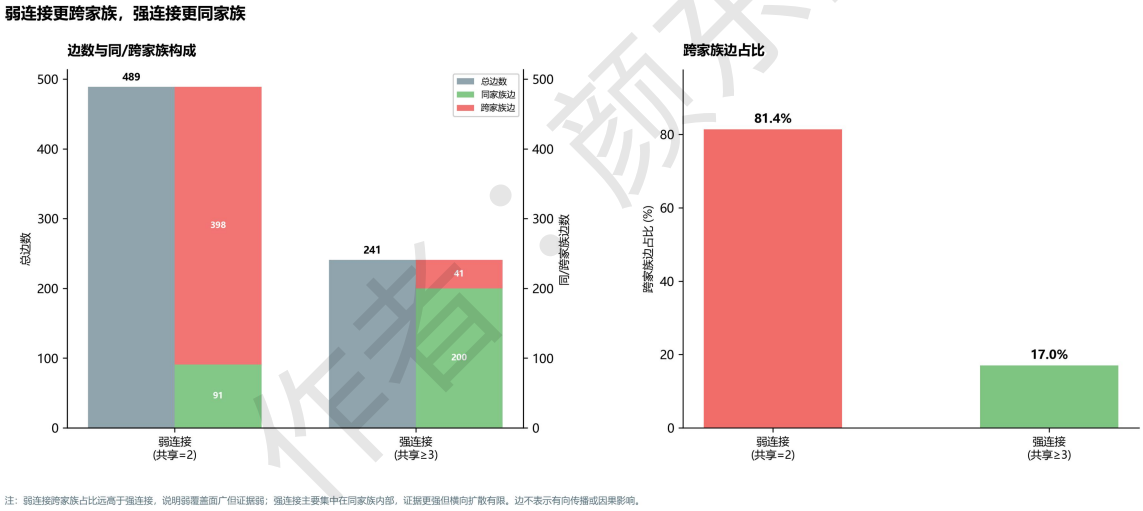

图 7 弱连接更跨家族，强连接更同家族

这一强弱差异具有重要解释意义。如果 T1 层存在普遍的跨家族技术路线趋同，那么在强连接中也应当观察到大量跨家族边；但结果恰恰相反。弱连接中的跨家族边占比很高，说明头部模型家族之间存在广泛的浅层共同采用；强连接中的跨家族边大幅减少，说明高强度技术组合重叠主要仍被家族内部结构吸收。也就是说，T1 层的共同采用不是一个强跨家族扩散网络，而更像是由广泛浅层重叠和家族内部深层重叠共同构成的分层结构。因此，本节将 T1 层共同采用概括为“弱连接广泛、强连接内聚”。

这一发现也为后续解释奠定了基础。仅观察共同采用网络本身，还不足以判断这些跨家族连接究竟代表具体技术路线共现，还是基础架构范式上的一般趋同。尤其是在大模型

领域，许多 T1 模型都会采用 decoder-only 架构、RoPE、GQA、SwiGLU、SentencePiece 等通用技术标签，这些标签可能在不同家族之间制造大量共同采用边。因而，下一步需要进一步拆解共同采用边的技术层级构成，并检验去除通用 L1 架构标签后，跨家族连接是否仍然稳健存在。

3.3.3 共同采用首先表现为基础架构范式趋同

弱连接与强连接的结构差异说明，T1 模型之间存在共同采用关系，但还不能说明共同采用的技术内容是什么。为了避免将一般性架构重叠误读为具体技术路线趋同，需要进一步拆解共同采用边所包含的技术层级。结果显示，无论弱连接还是强连接，T1 模型共同采用都首先表现为基础架构范式上的趋同，而不是具体技术路线的大规模跨家族重合。

从技术层级看，L1 核心架构标签在共同采用边中占据主导位置。弱连接中，L1 标签出现 656 次，占 67.1%；L3 后训练适配标签出现 269 次，占 27.5%；L2 训练生产和 L4 推理部署占比较低，分别为 3.9% 和 1.5%。强连接中，L1 标签的主导性进一步增强，出现 581 次，占 79.3%；L2 占 12.0%，L3 占 7.0%，L4 占 1.8%。这表明，当模型之间共享技术数量增加时，共享内容反而更集中于核心架构和基础建模范式。

表 11　弱连接 vs 强连接技术层级构成表

技术层级	弱连接出现次数	弱连接占比	强连接出现次数	强连接占比
L1	656	67.1%	581	79.3%
L2	38	3.9%	88	12.0%
L3	269	27.5%	51	7.0%
L4	15	1.5%	13	1.8%

注：技术层级包括 L1 核心架构、L2 训练生产、L3 后训练适配、L4 推理部署、L5=应用系统。

这一结果需要谨慎解释。L1 标签虽然被归入核心架构层，但其中相当一部分是大模型领域的通用架构范式或常见结构选择，例如 decoder_only、grouped_query_attention、swiglu、RoPE、rms_norm、sparse_moe 等。它们能够说明不同 T1 模型在基础架构选择上具有相似性，却不必然意味着某一具体技术路线在模型家族之间发生了可观察传播。尤其是 decoder_only 、 grouped_query_attention 、 swiglu 、 causal_language_modeling 、 sentencepiece 等标签，往往具有较强的通用性和范式属性，更适合被解释为基础架构趋同，而不是具体路线扩散。

强连接中的跨家族家族对进一步说明了这一点。以 Granite↔ Qwen 为例，该家族对在强连接中形成 28 条跨家族边，是强连接跨家族边最多的家族对；但其共享技术主要集中在 decoder_only、grouped_query_attention 和 swiglu，L1 占比达到 100%。因此，Granite↔ Qwen 虽然在网络上表现为高频跨家族强连接，但其含义更接近基础架构范式趋同，而不

是具体技术路线共同采用。换言之，边的数量较多并不自动意味着证据更强，还必须结合共享技术的层级和通用性来判断。

相比之下，少数家族对包含更多 L3 后训练适配或 L4 推理部署技术标签。例如 DeepSeek ↔ Nemotron 的强连接涉及 supervised_finetuning、grpo、vllm_runtime、sglang_runtime 等技术；Llama ↔ Nemotron 的部分连接涉及 vllm_runtime、tensor_parallel_inference、speculative_decoding、eagle_decoding 等技术。这类连接比单纯由 decoder_only、GQA 或 SwiGLU 驱动的连接更接近具体技术路线共同采用的候选线索。不过，即便如此，它们仍然只是模型标签层面的共现关系，而不是传播证据。

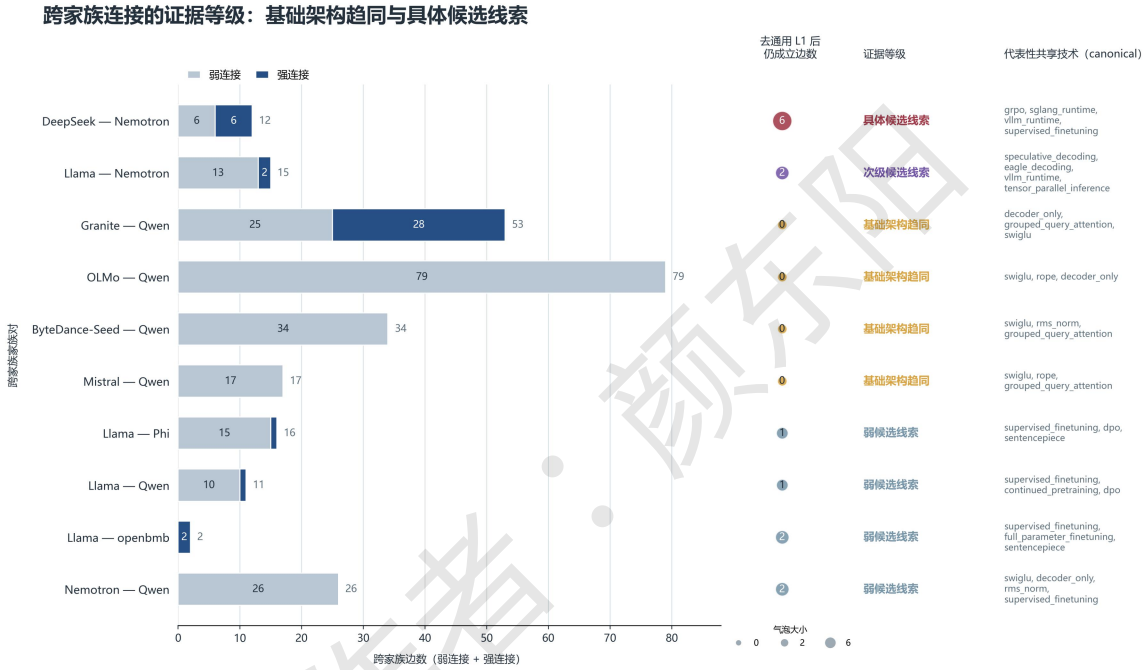

图 12　跨家族连接的证据等级：基础架构趋同与具体候选线索

注：跨家族边仅表示技术标签组合重叠，不表示谱系继承、有向传播或因果影响。L1 驱动连接主要反映基础架构范式趋同；去除通用 L1 标签后仍成立、且包含 L3/L4 具体技术的连接，才作为"具体技术路线共同采用候选线索"处理。

因此，T1 模型级共同采用网络需要区分两类含义。第一类是基础架构范式趋同，即多个头部模型家族在通用架构、基本建模结构和常见训练表示上选择相似方案；第二类是具体技术路线共同采用，即不同家族在后训练、推理部署、解码策略或特定训练方法上出现标签重叠。现有网络中，前者占据主要地位，后者数量较少且需要更谨慎解释。只有在完成这一层区分之后，T1 层共同采用才不会被误读为头部模型之间的有向技术扩散。

3.3.4 去除通用 L1 标签后的网络收缩

为了进一步检验共同采用网络在多大程度上依赖通用基础架构标签，本节对网络进行敏感性分析：剔除若干最具通用性的 L1 或基础建模标签后，重新观察模型级共同采用关

系是否仍然成立。剔除的标签包括 causal_language_modeling、decoder_only、grouped_query_attention、sentencepiece 和 swiglu。这些标签在大模型中广泛出现，能够反映基础架构范式或常见建模选择，但并不适合作为具体技术路线共同采用的强证据。

敏感性分析显示，去除通用 L1 标签后，共同采用网络明显收缩，尤其是跨家族强连接大幅减少。在强连接网络中，总边数从 241 条下降到 57 条，下降 76.3%；跨家族强连接从 41 条下降到 8 条，下降 80.5%。弱连接也出现明显收缩，跨家族弱连接从 398 条下降到 163 条，下降 59.0%。这说明，原始网络中的大量跨家族共同采用关系依赖于通用基础架构标签重叠，而不是由更具体的技术路线共现所驱动。

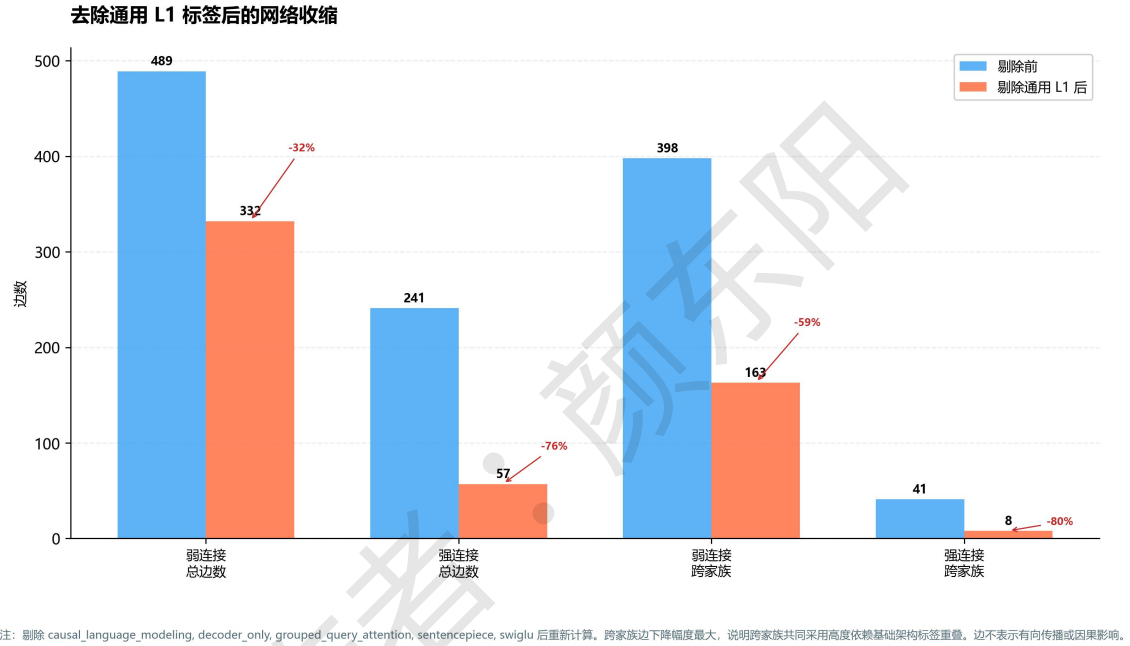

图 13　去除通用 L1 标签后的网络收缩

这一结果对解释 T1 层共同采用非常关键。如果不剔除通用 L1 标签，网络中会出现较多跨家族边，容易让人误以为头部模型家族之间存在广泛的具体路线趋同。但剔除这些通用标签后，跨家族强连接显著收缩，说明许多看似较强的跨家族关系，其实主要来自基础架构范式的一般共享。换言之，T1 层共同采用的主要证据并不是“具体技术路线在头部家族之间广泛传播”，而是“头部模型在基础架构范式上趋于相似”。

强连接的收缩尤其值得注意。强连接原本要求两个模型共享至少 3 项技术标签，因此比弱连接具有更高的组合重叠强度。然而，即便在这一较高阈值下，去除通用 L1 标签后，跨家族强连接仍从 41 条下降到 8 条。这表明，强连接并不天然等同于具体路线趋同；只有当强连接在剔除通用架构标签后仍然存在，并且包含 L3 后训练适配、L4 推理部署或其他更具体的技术标签时，才可以被视为具体技术路线共同采用的候选线索。

这一敏感性结果也重新界定了跨家族连接的解释等级。像 Granite↔ Qwen 这类原本较突出的强连接家族对，在剔除通用 L1 标签后解释强度明显下降，因为其共享技术主要由 decoder_only、grouped_query_attention 和 swiglu 等通用标签构成。相反，DeepSeek↔ Nemotron、Llama↔ Nemotron 等少数连接在剔除通用标签后仍保留部分强连接，并包含 grpo、sglang_runtime、vllm_runtime、speculative_decoding、eagle_decoding 等更具体的技术标签，因而更适合作为具体技术路线共同采用的候选线索。

表 12 去除通用 L1 后仍成立的跨家族具体候选线索

边类型	家族对	模型 A	模型 B	共享技术明细	层级构成
强连接	DeepSeek↔Nemotron	deepseek-ai/DeepSeek-R1	nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-NVFP4	sglang_runtime, supervised_finetuning, vllm_runtime	L3:1, L4:2
强连接	DeepSeek↔Nemotron	deepseek-ai/DeepSeek-R1	nvidia/NVIDIA-Nemotron-3-Ultra-550B-A55B-NVFP4	sglang_runtime, supervised_finetuning, vllm_runtime	L3:1, L4:2
强连接	DeepSeek↔Nemotron	deepseek-ai/DeepSeek-V4-Flash	nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-NVFP4	grpo, sparse_moe, supervised_finetuning	L1:1, L3:2
强连接	DeepSeek↔Nemotron	deepseek-ai/DeepSeek-V4-Flash	nvidia/NVIDIA-Nemotron-3-Ultra-550B-A55B-NVFP4	grpo, sparse_moe, supervised_finetuning	L1:1, L3:2
强连接	DeepSeek↔Nemotron	deepseek-ai/DeepSeek-V4-Pro	nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-NVFP4	grpo, sparse_moe, supervised_finetuning	L1:1, L3:2
强连接	DeepSeek↔Nemotron	deepseek-ai/DeepSeek-V4-Pro	nvidia/NVIDIA-Nemotron-3-Ultra-550B-A55B-NVFP4	grpo, sparse_moe, supervised_finetuning	L1:1, L3:2
弱连接	DeepSeek↔Nemotron	RedHatAI/DeepSeek-V3-BF16	nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-NVFP4	multi_token_prediction, sparse_moe	L1:1, L2:1
弱连接	DeepSeek↔Nemotron	RedHatAI/DeepSeek-V3-BF16	nvidia/NVIDIA-Nemotron-3-Ultra-550B-A55B-BF16	multi_token_prediction, sparse_moe	L1:1, L2:1
弱连接	DeepSeek↔Nemotron	RedHatAI/DeepSeek-V3-BF16	nvidia/NVIDIA-Nemotron-3-Ultra-550B-A55B-NVFP4	multi_token_prediction, sparse_moe	L1:1, L2:1
弱	DeepSeek↔Nemotron	deepseek-	nvidia/NVIDIA-	multi_token_prediction,	L1:1,

连接	ai/DeepSeek-V3	Nemotron-3-Super-120B-A12B-NVFP4	sparse_moe	L2:1

因此，去除通用 L1 标签后的网络收缩支持一个更稳妥的判断：T1 层确实存在非谱系式共同采用，但其主体部分是基础架构范式趋同，而不是具体技术路线的大规模横向扩散。共同采用网络能够揭示头部模型之间的技术组合重叠，却不能单独证明技术从一个家族传播到另一个家族。在这一前提下，下一步需要保留的只是少数剔除通用架构标签后仍然存在、且包含较具体技术标签的跨家族连接。

3.3.5 少量具体技术路线共现候选线索

在剔除通用 L1 标签后仍然保留的跨家族强连接中，DeepSeek↔ Nemotron 是最值得关注的候选线索。该家族对涉及 supervised_finetuning、grpo、sglang_runtime、vllm_runtime 等技术标签。这些标签不只是一般架构描述，而更多指向后训练策略、强化学习方法和推理部署框架。因此，相比 Granite ↔ Qwen 这类主要由 decoder_only、grouped_query_attention 和 swiglu 驱动的连接，DeepSeek↔ Nemotron 更接近具体技术路线层面的共同采用。

Llama↔ Nemotron 可以作为次级候选线索。该家族对在去除通用 L1 标签后仍保留少量强连接，涉及 supervised_finetuning、tensor_parallel_inference、vllm_runtime、speculative_decoding、eagle_decoding 等标签。这些技术更多分布在后训练、推理部署和解码优化环节，提示不同模型家族在若干具体路线上的共现。不过，与 DeepSeek↔ Nemotron 相比，Llama↔ Nemotron 的边数更少，涉及模型范围更有限，因此解释强度也应相应降低。

这些剩余连接的意义在于，它们提醒我们不能把 T1 层共同采用完全化约为基础架构趋同。头部模型之间确实存在少量更细粒度的技术标签重叠，尤其是在后训练、推理部署和解码优化等环节。但是，这些线索仍然不能被写作有向传播证据。共同采用网络中的边是无向边，只表示模型对之间存在技术标签交集；技术标签共现不等于实际技术继承；公开数据中的首次观察时间也不等于技术发明时间或组织间采用时间。因此，DeepSeek↔ Nemotron、Llama↔ Nemotron 等连接更稳妥的表述是“具体技术路线共同采用候选线索”，而不是“技术传播路径”。

由此看，T1 层的技术变化具有双重面貌：一方面，大量跨家族共同采用来自基础架构范式趋同；另一方面，少数连接显示出更具体的技术路线共现。两者合在一起，构成了 T1 层非谱系式共同采用的主要图景：范式趋同是主体，具体路线共现是少数候选线索。

综上，T1 层不宜被描述为通过可观察谱系边发生有向扩散的网络。全量可靠谱系边中 T1→T1 显式边极少，无法支撑头部模型之间的方向性传播分析；模型级共同采用网络揭示的，是 T1 模型之间在技术标签组合上的非谱系式重叠。这种重叠呈现明显分层：弱连接更广泛，并更多跨越模型家族；强连接则主要集中在同家族内部，反映同系列模型的架构复用、版本延展和披露模板相似。进一步剔除通用 L1 架构标签后，跨家族强连接大幅收缩，仅剩少数以 DeepSeek↭ Nemotron、Llama↭ Nemotron 为代表的具体技术路线共同采用候选线索。因此，T1 层更稳妥地表现为“弱连接广泛、强连接内聚”：它显示的是基础架构范式趋同和少量具体路线共现，而不是可观察的有向技术传播。

3.4 MoE 技术路线的复合扩散形态

前文说明，T1 层缺少足够的 T1→T1 显式谱系边，因此不宜被解释为头部模型之间的有向技术传播；模型级共同采用网络更稳妥地显示为基础架构范式趋同和少量具体技术路线共现。为了进一步说明这种判断，本节以 MoE 技术路线为例，观察同一技术路线在不同层级中的扩散形态。MoE 是一个适合展开的案例，因为它一方面属于 T1 层多个头部模型家族共同采用的架构路线，另一方面又在 T2/T3 层通过微调、量化、适配、蒸馏和再发布进入更明确的模型谱系扩散过程。

从采用总览看，MoE 并不只存在于头部模型层。固定 6 项 MoE 相关技术后，T1 层共有 184 条模型×技术采用事件，涉及 171 个模型、21 个发布账号和 6 项技术；T2 层共有 79 条事件，涉及 77 个模型、18 个账号和 6 项技术；T3 层共有 96 条事件，涉及 89 个模型、44 个账号和 4 项技术。这说明，MoE 作为一条技术路线，在 T1、T2 和 T3 层均有可观察采用，并非只停留在少数头部模型内部。

表 13　MoE 采用总览表

样本框	事件数	模型数	账号数	技术项
T1	184	171	21	6
T2	79	77	18	6
T3	96	89	44	4

在 T1 层，MoE 的关键事实不是可观察的 T1→T1 谱系传播，而是多个头部发布账号的非谱系式共同采用。T1 层中共有 21 个账号出现 MoE 相关技术采用记录，时间上覆盖 DeepSeek、Qwen、BAAI、OLMo、Granite、MiniMax、Phi、Hunyuan、ERNIE、Kimi、Gemma 等多个模型家族或发布主体。但是，在 MoE 相关技术事件中，并没有形成可支持

T1 MoE→T1 MoE 的显式谱系边。因此，T1 层的 MoE 更适合被解释为领域级共同采用或架构范式趋同，而不是头部模型之间已经被观察到的有向传播。

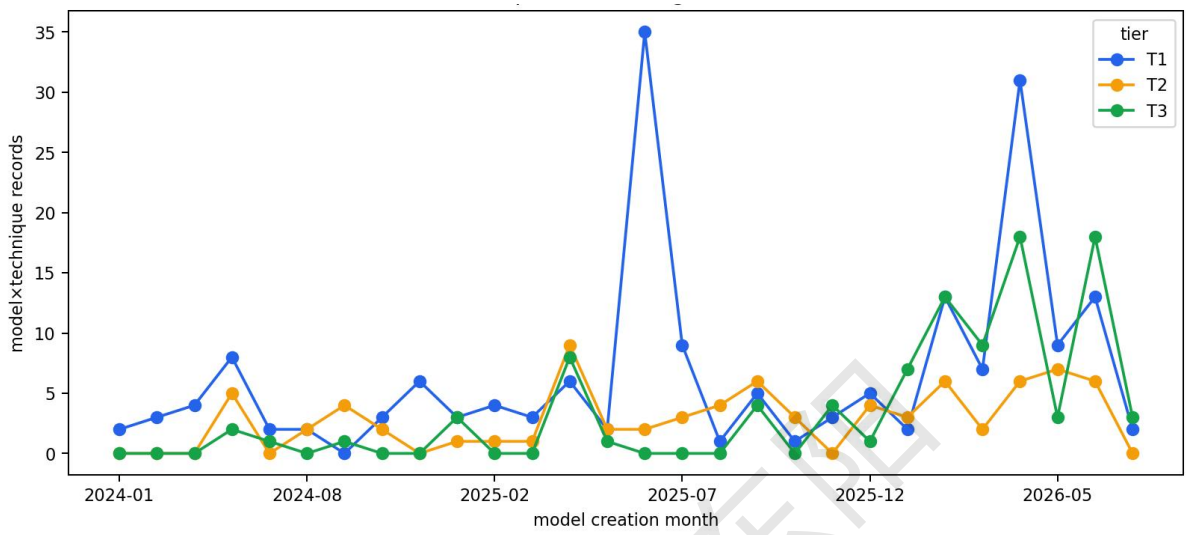

图 14　MoE 技术在不同层级中的采用趋势

这一点与 3.3 的模型级共同采用网络结论一致。MoE 在 T1 层的多账号出现，说明头部模型家族之间确实围绕稀疏专家、专家并行、细粒度专家等方向形成了共同技术选择；但这种共同选择本身并不能说明某一 T1 家族向另一 T1 家族发生了技术转移。公开数据中的首次观察时间也不等于技术发明时间或组织间采用时间。因此，T1 MoE 的证据意义应被限定为“非谱系式共同采用”，而不是“核心层有向扩散”。

与 T1 层不同，MoE 进入下游层后，更多表现为可观察谱系关系中的再加工和再分发。MoE 相关谱系事件中，T1→T2 和 T1→T3 是最主要的向下流向。按关系类型看，T1→T2 中包括量化 98 条、微调 37 条、蒸馏 1 条、合并 2 条，以及未明确类型 39 条；T1→T3 中包括量化 145 条、微调 19 条、适配 2 条、蒸馏 1 条，以及未明确类型 14 条。与此同时，T2→T2 和 T2→T3 也存在较多 MoE 相关事件，说明下游层并非只接收 T1 模型，而会继续围绕 MoE 模型进行二次加工和再分发。

表 14 MoE 谱系扩散中的主要层级流向与关系类型

层级流向	主要关系类型	事件数	解释
T1→T2	量化、微调、合并、蒸馏、未明确	177	头部 MoE 模型进入主要衍生模型层，以下游再加工为主
T1→T3	量化、微调、适配、蒸馏、未明确	181	头部 MoE 模型进入高影响社区模型层，量化和再发布尤其突出

T2→T2	量化、微调、合并、未明确	57	主要衍生模型之间继续发生 MoE 相关再加工
T2→T3	量化、微调、适配、蒸馏、未明确	96	T2 层进一步向社区高影响模型扩散

这些谱系事件说明，MoE 在下游层的扩散形态与 T1 层不同。在 T1 层，MoE 主要表现为缺少显式父子边的共同采用；而一旦进入 T2/T3 层，MoE 相关模型更频繁地通过可观察谱系关系被量化、微调、适配、蒸馏、合并和再发布。尤其是量化事件在 T1→T2 和 T1→T3 中数量较高，说明许多下游采用并不是重新生产 MoE 架构本身，而是在已有开放模型制品基础上进行部署压缩、格式转换和再分发。这与前文关于“强外围再加工”的判断一致。

因此，MoE 展示的是一种复合扩散形态，而不是单一方向的技术传播。在核心层，MoE 更接近非谱系式共同采用：多个 T1 发布主体在相近时期选择或披露 MoE 相关架构路线，但缺少可观察的 T1→T1 谱系传播证据。在下游层，MoE 则进入更明确的谱系式再加工网络，通过量化、微调、适配、蒸馏和再发布扩散到 T2/T3 模型。换言之，同一技术路线在不同层级中具有不同扩散机制：核心层表现为范式趋同和共同采用，下游层表现为围绕开放模型制品的谱系式再加工扩散。

这一案例进一步强化了本章的总体判断。开放权重大模型生态中的技术流动并不是单一的“从核心到外围复制”，也不是头部模型之间可以直接观察到的横向有向传播。更稳妥的理解是：T1 层形成若干架构范式和技术路线的共同采用，下游层则围绕已经开放的模型制品展开大规模再加工、压缩、适配和再分发。MoE 正好展示了这两种机制如何叠加：它在 T1 层提供共同采用线索，在 T2/T3 层呈现谱系式扩散证据，从而构成开放模型生态中“核心共同采用 + 外围再加工扩散”的复合形态。

3.5 下游回流的候选线索与证据边界

开放技术系统是否形成完整创新闭环，关键不只在于技术是否向下扩散，也在于下游再加工能否重新进入核心层。对于开放权重大模型而言，前文已经说明，现有公开数据能够较清楚地支持两个判断：第一，T1/T2 模型通过可靠谱系边形成较强的向下扩散；第二，T2/T3 层围绕开放模型制品形成活跃的微调、量化、适配、部署和再分发活动。但是，能否进一步证明下游创新反过来进入 T1 核心层，则需要更高证据标准。

本节将“下游回流”限定为一种更严格的现象：某项技术、方法或模型加工实践先在下游模型中出现，随后被官方或核心层模型采用，并且能够通过明确文本、代码依赖、致谢、引用、版本说明或其他材料建立方向性联系。仅有时间先后和技术相似，并不足以证

明回流；同一组织内部的版本序列，也不能直接写成外部下游对核心层的反馈。因此，本节采用证据漏斗而不是确认计数来处理回流问题，将可能线索、同组织版本序列、README 引用候选和已确认强证据区分开来。

现有结果显示，回流候选数量并不少，但能够升级为强证据的案例尚未形成。基于当前快照，时间先后型回流候选共有 2,524 条，即下游或社区模型中某项技术的首次观察时间早于某个官方模型采用时间；同组织版本序列候选共有 20,623 条，主要反映同一组织或同一模型家族内部的后续版本变化；README 引用候选共有 72 条，来自模型卡或 README 中对其他模型、权重、配置或方法的文本引用。但是，在人工核验后的证据等级中，已确认强证据 A、强候选 B、描述性证据 C 和排除项 D 目前均未赋值，其中已确认强证据 A 为 0。这意味着，当前材料可以支持“存在一批值得核查的回流候选”，但不能支持“已经观察到明确的下游知识回流”。

从候选内容看，时间先后型线索主要包括两类。一类是社区模型或下游模型较早出现某项技术标签，随后官方或较核心模型出现相同标签，例如分组查询注意力、特殊聊天标记、AWQ、llama.cpp 运行时、监督微调等。另一类是下游分发账号围绕某些模型形成量化、运行时适配或格式转换实践，随后在官方或较核心账号中出现相似技术标签。这些线索具有启发意义，因为它们提示下游生态可能不仅是被动接收者，也可能在部署、量化、格式转换、推理运行时和对齐适配等环节率先形成实践。

但是，这类时间先后型线索的证据强度很低。首先，公开平台中的模型创建时间或首次观察时间并不等于技术发明时间，也不等于组织内部实际采用时间。其次，许多时间差只有数小时或数天，可能反映发布节奏、模型卡更新、元数据抓取时间或同一基础模型的同步分发，而不是下游影响上游。再次，相同技术标签可能来自共同基础设施、论文、开源库或行业范式，而不必然来自某个下游模型。因此，时间领先只能作为人工核验队列，不能直接写成知识回流。

README 引用候选提供了另一类较具体的线索，但同样不能直接升级为回流证据。这些候选通常来自模型卡中对其他模型、权重、配置、适配器或量化来源的引用，例如使用某个基础模型、加载 LoRA、合并权重、引用推测解码模型、说明量化来源等。它们能够帮助识别模型之间的显式文本关联，但多数情况下更像是下游对上游的引用、派生或复用说明，而不是上游吸收下游创新。只有当 README 或相关材料明确表明官方模型采纳了下游方法、代码、权重、数据处理流程或部署实践时，才可能构成较强回流证据。

同组织版本序列也需要单独处理。现有结果中，同组织版本序列候选数量很高，达到 20,623 条，但这部分更可能反映同一发布主体内部的模型迭代、配置更新、版本扩展或系列化发布。它们对于理解核心模型家族内部的技术演进有价值，却不能简单归入开放生态

中的“下游回流”。如果不加区分，容易把组织内部研发迭代误写成外部社区对核心层的反馈。

因此，当前材料对下游回流的支持应当被表述为“候选线索存在，但确认强度不足”。下游层确实在后训练、量化、部署、格式转换和推理运行时等方面形成大量实践，这些实践理论上可能通过工具链、模型卡、社区反馈、推理框架、评测压力或分发标准影响核心模型生产者；但在本研究当前数据中，这种影响更多停留在可疑时间序列、标签相似和文本引用候选层面，尚未形成经过人工核验的方向性证据。

这一结果并不削弱前文关于开放权重扩散的判断，反而进一步说明开放技术系统的层级不对称。开放权重降低了下游主体围绕模型制品再加工的门槛，因此能够观察到强向下扩散和活跃外围创新；但由于训练数据、训练流程、组织内部实验、模型设计决策和商业部署反馈并不充分公开，下游创新如何进入核心模型生产过程更难被平台数据捕捉。换言之，开放权重释放了下游再加工能力，却没有同等程度地开放核心知识生产过程的反馈链条。

综上，现有公开数据尚不能证明开放权重大模型生态已经形成明确的“下游创新回流核心层”闭环。更稳妥的判断是：下游回流存在若干候选线索，尤其集中在量化、推理部署、运行时适配、格式转换和部分后训练实践中；但这些线索目前仍需要人工核验，不能写成已确认回流。就本章证据而言，开放模型生态呈现的是强向下扩散、强外围再加工、弱可观察核心回流。由此，开放技术系统不是一个完全对称的双向知识循环，而是一个以开放模型制品为中心、向下扩散清晰、向上反馈间接且证据较弱的分层创新体系。

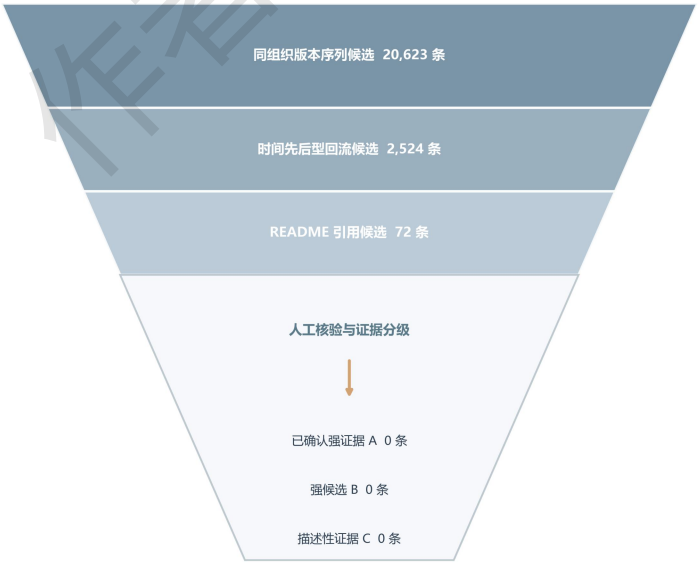

图 15　下游知识回流候选证据漏斗

附录

T1/T2 样本框白名单机构列表

本附录列出用于界定 T1/T2 样本框的官方机构白名单，共 89 个 HuggingFace 发布账号。白名单来源为 config.yaml 的 official_orgs 字段，判定逻辑为：若模型的 author 字段（或 model_id 的命名空间部分）在白名单中（大小写不敏感），则该模型被标记为 official=True，归入 T1（主力模型）或 T2（官方衍生模型）；否则归入 T3（社区高活跃）、T4（社区长尾）或 T5（社区原创）。白名单的构建依据包括：原 19 个头部公司与研究机构、中国头部公司新增、海外头部公司新增、研究/嵌入/社区类、同公司不同账号补充、高影响大模型账号摸排、以及 A 类高影响主要玩家补充。

表 白名单机构列表（共 89 个账号）

序号	HF 账号	机构中文名	类别	说明
1	meta-llama	Meta	海外头部公司	Llama 系列
2	facebook	Meta（旧账号）	海外头部公司	Meta 早期发布账号
3	qwen	阿里巴巴通义千问	中国头部公司	Qwen 系列
4	deepseek-ai	深度求索	中国头部公司	DeepSeek 系列
5	mistralai	Mistral AI	海外头部公司	Mistral/Mixtral 系列
6	google	Google	海外头部公司	Gemma 系列
7	microsoft	Microsoft	海外头部公司	Phi 系列
8	allenai	Allen AI	海外研究机构	OLMo/Tulu 系列
9	ibm-granite	IBM	海外头部公司	Granite 系列
10	ibm	IBM	海外头部公司	IBM 通用账号
11	nvidia	NVIDIA	海外头部公司	Nemotron 系列
12	bigscience	BigScience	海外研究项目	BLOOM 系列
13	tiiuae	TII	海外研究机构	Falcon 系列
14	huggingface	Hugging Face	海外平台/社区	官方账号
15	huggingfacetb	Hugging Face	海外平台/社区	TB 团队账号
16	openai	OpenAI	海外头部公司	GPT 系列（开源权重部分）
17	baai	北京智源研究院	中国研究机构	BGE/CogView 系列
18	bigcode	BigCode	海外研究项目	StarCoder 系列
19	cohere	Cohere	海外头部公司	Command 系列
20	cohereforai	Cohere for AI	海外研究机构	Cohere 研究分支
21	eleutherai	EleutherAI	海外研究机构	GPT-NeoX/Pythia 系列
22	sentence-	Sentence	海外研究项目	嵌入式模型

	transformers	Transformers		
23	stabilityai	Stability AI	海外头部公司	Stable LM 系列
24	zai-org	智谱 AI	中国头部公司	GLM 系列
25	moonshotai	月之暗面	中国头部公司	Kimi 系列
26	minimaxai	MiniMax	中国头部公司	MiniMax 系列
27	tencent	腾讯	中国头部公司	Hunyuan/Hy3 系列
28	bytedance	字节跳动	中国头部公司	Doubao 系列
29	alibaba-pai	阿里巴巴 PAI	中国头部公司	PAI 系列模型
30	tongyi-mai	通义实验室	中国头部公司	通义系列衍生
31	wan-ai	Wan AI	中国头部公司	Wan 系列
32	01-ai	零一万物	中国头部公司	Yi 系列
33	opengvlab	OpenGVLab	中国研究机构	视觉多模态
34	openbmb	OpenBMB	中国研究项目	MiniCPM 系列
35	anthropic	Anthropic	海外头部公司	Claude 系列（开源部分）
36	apple	Apple	海外头部公司	OpenELM 系列
37	amazon	—	—	—
38	intel	Intel	海外头部公司	量化转换分发
39	black-forest-labs	Black Forest Labs	海外头部公司	FLUX 系列
40	genmo	Genmo	海外公司	Mochi 系列
41	liquidai	Liquid AI	海外公司	LFM 系列
42	nomic-ai	Nomic AI	海外研究机构	嵌入式模型
43	nousresearch	Nous Research	海外研究机构	Hermes 系列
44	comfy-org	Comfy Org	海外社区	ComfyUI 生态
45	google-bert	Google	海外头部公司	BERT 发布账号
46	FacebookAI	Meta	海外头部公司	Meta AI 旧账号（xlm-roberta 等）
47	openai-community	OpenAI	海外头部公司	社区账号（gpt2 等）
48	llava-hf	LLaVA	海外研究项目	LLaVA 官方 HF 镜像
49	intfloat	IntFloat	海外研究者	e5 embedding 系列
50	jinaai	Jina AI	海外公司	Jina 嵌入式模型
51	Snowflake	Snowflake	海外公司	Arctic Embed 系列
52	Salesforce	Salesforce	海外头部公司	BLIP 系列
53	mixedbread-ai	Mixedbread AI	海外公司	mxbai-embed 系列
54	pyannote	pyannote	海外研究项目	语音分割/说话人识别
55	tinyllama	TinyLlama	海外研究项目	TinyLlama 系列
56	unslothai	Unsloth	海外公司	Unsloth 官方账号
57	baidu	百度	中国头部公司	Unlimited-OCR 等
58	alibaba-nlp	阿里巴巴 NLP	中国头部公司	gte embedding 系列
59	h2oai	H2O.ai	海外公司	h2ovl-mississippi 系列
60	stable-diffusion-v1-5	Stability AI	海外头部公司	SD 官方镜像账号

61	stepfun-ai	阶跃星辰	中国头部公司	Step 系列
62	ByteDance-Seed	字节跳动	中国头部公司	Seed 系列模型
63	inclusionAI	蚂蚁集团	中国头部公司	Ant 系列
64	CohereLabs	Cohere	海外头部公司	Cohere 实验室
65	internlm	上海 AI Lab	中国研究机构	书生系列
66	XiaomiMiMo	小米	中国头部公司	MiMo 系列
67	LGAI-EXAONE	LG	海外头部公司	EXAONE 系列
68	PaddlePaddle	百度飞桨	中国头部公司	飞桨生态模型
69	ATH-MaaS	Athene	海外公司	Athene 系列
70	naver	Naver	海外头部公司	韩国最大搜索引擎
71	Falconsai	Falcon（UAE）	海外研究机构	UAE Falcon 系列
72	cross-encoder	Cross Encoder	海外研究项目	句子相似度官方
73	distilbert	—	—	DistilBERT 官方
74	ggml-org	—	—	GGML/gguf 官方
75	RedHatAI	—	—	Red Hat
76	speechbrain	—	—	SpeechBrain 官方
77	WhereIsAI	—	—	bge 微调作者
78	base	—	—	--- 补充：A 类高影响主要玩家/研究机构（基于前 0.5%下载摸排）---
79	instruct	—	—	—
80	chat	—	—	—
81	reasoning	—	—	—
82	code	—	—	—
83	embedding	—	—	—
84	tiny-random	—	—	—
85	tiny_random	—	—	—
86	dummy	—	—	—
87	toy-model	—	—	—
88	test-model	—	—	—
89	debug	—	—	—

注：1) 白名单匹配为大小写不敏感，HuggingFace 的 author 字段大小写敏感（如 Qwen / BAAI / Alibaba-NLP 等），统一转小写后比对。2) 同一机构可能有多个账号（如 Google 有 google 和 google-bert，Meta 有 meta-llama、facebook 和 FacebookAI），均分别列入白名单以避免主力模型被遗漏。3) 部分账号为研究项目或社区组织（如 BigScience、BigCode、OpenBMB），因其产出具有重大影响的开源模型而纳入。