

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Learnware and AI Model Management System

Zhi-Hua Zhou 南京大学 arXiv 2026

Background

AI 的未来：三个问题与作者的判断

一个「英雄模型」
解决所有任务？

→ **多个模型协作**
临床诊断与金融风控难由一个模型都做好

模型由同一个
开发者提供？

→ **来自众多开发者**
没有谁拥有所有领域的知识与数据

开发者的训练数据
会公开吗？

→ **大多保持私有**
病历、交易记录等敏感数据难以公开

现状：模型「存储」系统



Hugging Face 等模型池
≈ 模型文件 + 手写描述（model card）

核心问题：不访问开发者训练数据（最好也不访问用户数据），能否为新任务找到有用的模型？

Motivation

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

从「存储」到「管理」：缺的是对模型能力的可计算刻画

数据	文件存储 只有创建者能按预想方式使用	→ 数据库管理系统 schema + 索引：可查询、可索引、可组合
AI 模型	模型池（如 Hugging Face） 模型文件 + 文字描述	→ AI 模型管理系统 ？ 识别、复用、组装模型

只改进「存储」为什么不够？

① 文字描述不可靠
模型复杂到开发者自己都说不清；
用户也说不清自己的需求

② 只能用于预想任务
模型通常只被用在
开发时预想到的任务上

③ 无法组装
不能把为不同任务开发的模型
组合起来解决新任务

硬约束：这种能力刻画必须在不泄露开发者与用户数据的前提下得到

作者的答案：把管理的基本单元从「模型」升级为「学件」，学件 = 模型 + 规格

规格由机器学习过程生成（而非手写）· 在本地生成 · 跨开发者可比较 · 理论上不泄露训练数据

Specification

学件规格：用机器可处理的形式描述模型能力

规格 = 语义规格 + 统计规格

语义规格 → 过滤：能不能比？

记录任务类别、输入输出接口、数据格式、单位、标签含义

例：数据类型 table · 任务类型 回归 · 场景 商业

统计规格 → 排序：谁更合适？

刻画与模型能力相关的统计性质，如模型擅长的数据分布

同一任务的模型也可能适合不同人群，手写标签难以区分

生成协议：系统下发，本地计算

学件坞系统 LDS

下发：共享函数 F + 规模预算 n

提交：模型 f + 规格 Ω

开发者本地：用 F 处理模型与训练数据，生成规格 Ω

训练数据不出本地 · 同一协议 ⇒ 规格之间可比较

三种统计规格：区别在于 F 是什么（按模型类型选择）

RKME F = 核函数（显式）

适用：常规预测模型

规格：（输入, 模型输出）的带权合成原型

比较：MMD 距离

PAVE F = 参考模型（隐式）

适用：大语言模型等复杂模型

规格：模仿微调参考模型得到的参数增量 Δθ

比较：余弦相似度

构造式规格 F = 参考任务 + 参考模型

适用：黑盒 API / 智能体

规格：微调参考模型得到的分层参数向量

比较：余弦相似度

共同点：用户用同一协议生成任务规格 · 比较只用规格，不调用模型、不接触原始数据

Specification

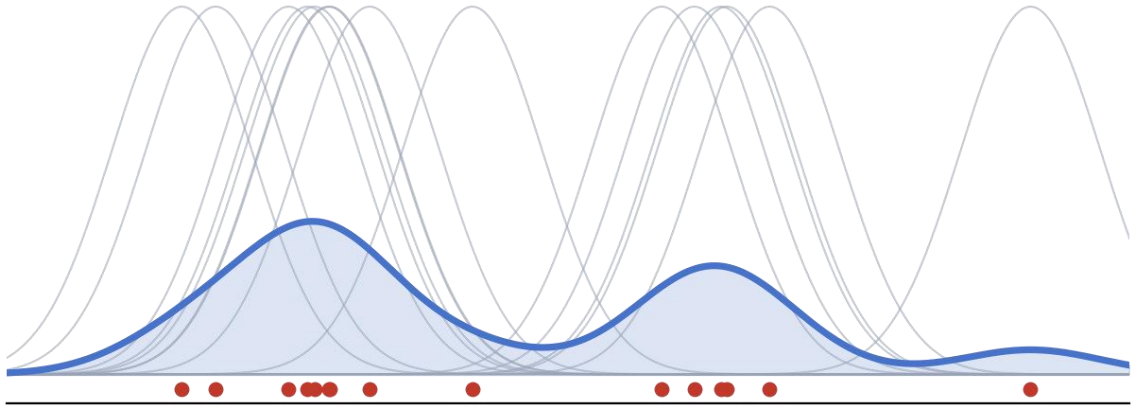
RKME（约简核均值嵌入）①：把模型的「行为分布」变成一个点

① 记录行为：z = (x, f(x))

开发者用模型 f 预测每个训练样本 x_i，得到 $z_i = (x_i, f(x_i))$
用的是模型输出而非训练标签：既记录模型**在哪里**开发，也记录它在那里**怎么判断**

只嵌入输入 x	✗ 分不出「同一人群、判断不同」的模型
只嵌入输出 f(x)	✗ 丢掉了「判断发生在哪里」
嵌入 (x, f(x))	✓ 两者都保留

② 核均值嵌入（KME）：鼓包平均成「地形图」



● 样本 z_i　— 单个鼓包 $k(z_i, \cdot)$　— 平均后的地形图 $\hat{\mu}$

灰线：横轴各位置与样本 z_i 的相似度；蓝线：m 条灰线逐点平均（i 为样本编号）

高斯核 $k(z, z') = \exp(-\gamma\|z - z'\|^2)$　　$\hat{\mu} = (1/m)\,\Sigma_i\, k(z_i, \cdot)$

关键性质：用特征核（如高斯核）时，分布不同 ⇔ 地形图不同，这张图能代表整个行为分布，而不只是均值、方差

问题：$\hat{\mu}$ 每个训练样本占一项（共 m 项），既太大，又贴近原始数据

→ 下一页：压缩成 n 个带权合成原型（RKME），并用 MMD 比较

Specification

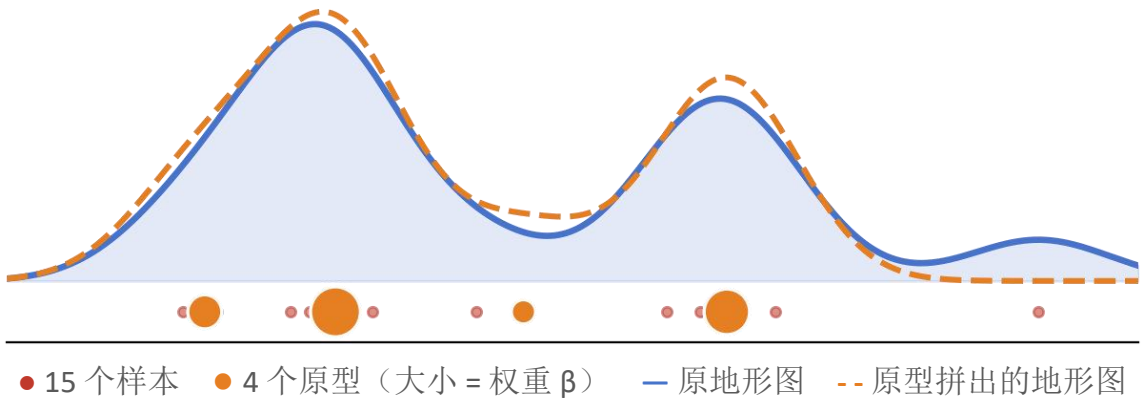
RKME ②：约简成 n 个带权原型，用 MMD 比较

③ 约简：用 n 个带权合成点拼出同一张地形图

$$\min_{\beta,\, t} \left\| \frac{1}{m}\sum_{i=1}^{m} k(z_i, \cdot) - \sum_{j=1}^{n} \beta_j k(t_j, \cdot) \right\|_{\mathcal{H}}^2$$

规格 Ω：n 个合成点 t 及其权重 β，$n \ll m$

合成点不是真实样本；目标是保留分布信息，而不是还原单个样本



④ 比较：MMD（最大均值差异）

$$d^2(a,b) = \|\tilde{\mu}_a - \tilde{\mu}_b\|_{\mathcal{H}}^2$$

展开后只需两组原型之间的核函数值

不调用模型、不碰原始数据；计算量取决于 n，而不是 m

⑤ 用户任务规格：同一个核，同样的形式

系统把同一个核函数发给用户，用户用少量带标签样本在本地生成

只有无标签数据时，则只刻画输入分布（适用于协变量偏移）

⑥ 规模预算 n：信息量、速度与安全的平衡

n 越大，信息越多，但比较越慢、泄露风险越高

理论保证：合成点与真实记录重合的概率为 0；$n \le \sqrt{m}$ 时近似误差与泄露风险均渐近趋于 0

Specification

PAVE（参数向量规格）：用「模仿之后的参数变化」刻画大模型

使用场景

复杂模型（如大语言模型）： 开发者手里有模型和数据，但模型规模大、输入是开放文本，难以像 RKME 那样在原始输入上显式构造并压缩嵌入

对比：常规针对性任务预测模型 → RKME

生成与比较

① 系统下发
预训练参考模型 F_θ
如 0.5B 小模型，所有人共用

→

② 开发者本地
用自有数据询问模型 f
得到 (x, f(x))

→

③ 模仿微调
微调参考模型去模仿 f
模仿的是 f 的输出，而非训练标签

→

④ 提交规格
$\Omega = \Delta\theta$
微调前后参数的变化量

用户： 同一个参考模型 + 自己的少量带标签数据 → 用户的 $\Delta\theta$　　**比较：** 余弦相似度 $\cos(\Delta\theta_1, \Delta\theta_2)$，方向越一致越匹配

为什么必须用同一个参考模型

初始化、损失、输出接口、微调方式都相同，
$\Delta\theta$ 才在同一坐标系里；
底座不同则无法比较 → 参考模型是「契约」

与 RKME 的关系

神经正切核视角：$\Delta\theta \approx$ 逐样本梯度特征之和，
相当于一个隐式的核均值嵌入；
与 RKME 给候选模型的排序一致

压缩与安全

低秩分解 $\Delta W = B \cdot A$：A 为公开固定的随机矩阵，
只训练并提交 B → 存储 < 参考模型的 1%；
PAVE 自带 (ε, δ)-差分隐私保证

直观理解：同一个「空白学生」被不同老师各教一遍，学生的参数被推向哪个方向，就反映这位老师擅长什么

Specification

构造式规格：用「在公共考卷上做对了哪些题」刻画黑盒服务

使用场景

以托管服务形式提供的智能体 / 大模型：训练数据和内部参数都拿不到，只能「提问—看回答」

RKME 需要训练数据，PAVE 需要开发者本地数据 → 都用不了，只能从可观察的行为出发

生成流程

① 公共考卷
系统维护一组参考任务
每个 benchmark 的题即探针

→

② 预算内测评
只挑一部分题让服务作答
如只考 300 题，不考整套

→

③ 分层编码
用做对的题微调参考模型
难题权重更大，得到参数向量

→

④ 提交规格
任务级 + 簇级向量
比较时用余弦相似度

为什么不能只记一个总分

服务 A ■■■■■■■■□□□□ 70 分

服务 B □□□■■■■■■■■ 70 分

同分不同对：总分一样，擅长的题完全不同
用户的问题未必正好落在某个参考任务上

预算内测评：省着考

先用固定的基线模型把题分成易 / 难：易题多且重复 → 聚类抽代表；难题区分度高 → 全保留；乐观估计（UCB 思路）：附近的题都没做对，就连乐观估计都不达标 → 这题直接跳过

分层编码：只看做对的题

任务级向量概括整个参考任务；簇级向量刻画任务内各语义簇，区分「同分不同对」；基线做错、而该服务做对的难题 → 权重更大，更能区分模型

即插即用：每个服务只对照公共考卷和参考模型生成自己的规格，新增 / 下线无需全局更新　**能力区域：**可按查询分派

Specification

智能体学件：按「单个查询」分工，而不是给整个任务只选一个

预测模型
一个任务，选一次学件就够了

智能体
同一任务里，不同智能体擅长不同的查询 → 要按查询识别

难点
单个查询太短，信息太少，没法直接生成可靠的规格

解法：把系统维护的参考任务当作「上下文库」

① **找参考任务**
查询先匹配到最相关的参考任务

② **取邻近题目**
从对应 benchmark 取出相似的题，凑成上下文

③ **生成查询规格**
按和智能体规格相同的协议生成

④ **两层匹配**
在任务级和簇级比较，聚合最高的几个相似度

⑤ **分派**
交给最擅长这类查询的智能体

补充信号：聚合多个高相似度，减少偶然的局部误匹配；参考任务覆盖不到这个查询时，再看智能体在「难题」上做对了多少

✓ 已经做到：按能力分工的基本协作

- 同一任务的不同查询，可以交给不同的智能体
- 只需「提问—看回答」，不访问智能体内部
- 新智能体加入时，只需生成它自己的规格，不用重训一个中心路由器

? 尚未解决：从「识别」走向「编排」

- 目前只看最终回答；更一般的智能体会执行代码、改文件、调工具、相互通信
- 能力要在执行轨迹、资源消耗、安全性上评估，还要随任务推进反复分派
- 需要「可组合规格」：不仅说明能做什么，还说明通过哪些工具和交互来组合

LDS

学件坞系统（LDS）：不是模型仓库，而是以规格为接口的学件管理系统

四个模块：规格是控制接口，模型代码隔离执行

开发者提交：模型 + 规格 ↓

用户提交：任务规格 ↓

检查器 Checker
准入把关
能否运行、语义接口是否一致
是否有可用的性能证据
模型一改 → 规格须重新生成

→

组织器 Organizer
管理学件群体
维护群体证据与索引
构建元先验与组织结构
（覆盖树、锚点等，后面讲）

→

搜索器 Searcher
找候选
用户任务规格 → 候选
返回预算内的短名单
并附上匹配证据

→

复用器 Reuser
落地到用户任务
把选中的候选变成
适配用户任务的实现
（直接用 / 适配 / 组装）

一次查询：分阶段决策，而不是一次最近邻查找

语义规格过滤
输入输出、部署约束兼容

→

统计规格排序
在兼容集合里比能力

→

短名单 + 证据
借助元先验与组织结构

→

选择复用路线
直接/适配/增强/路由/组装

→

用户侧验证
覆盖不足时系统可弃权

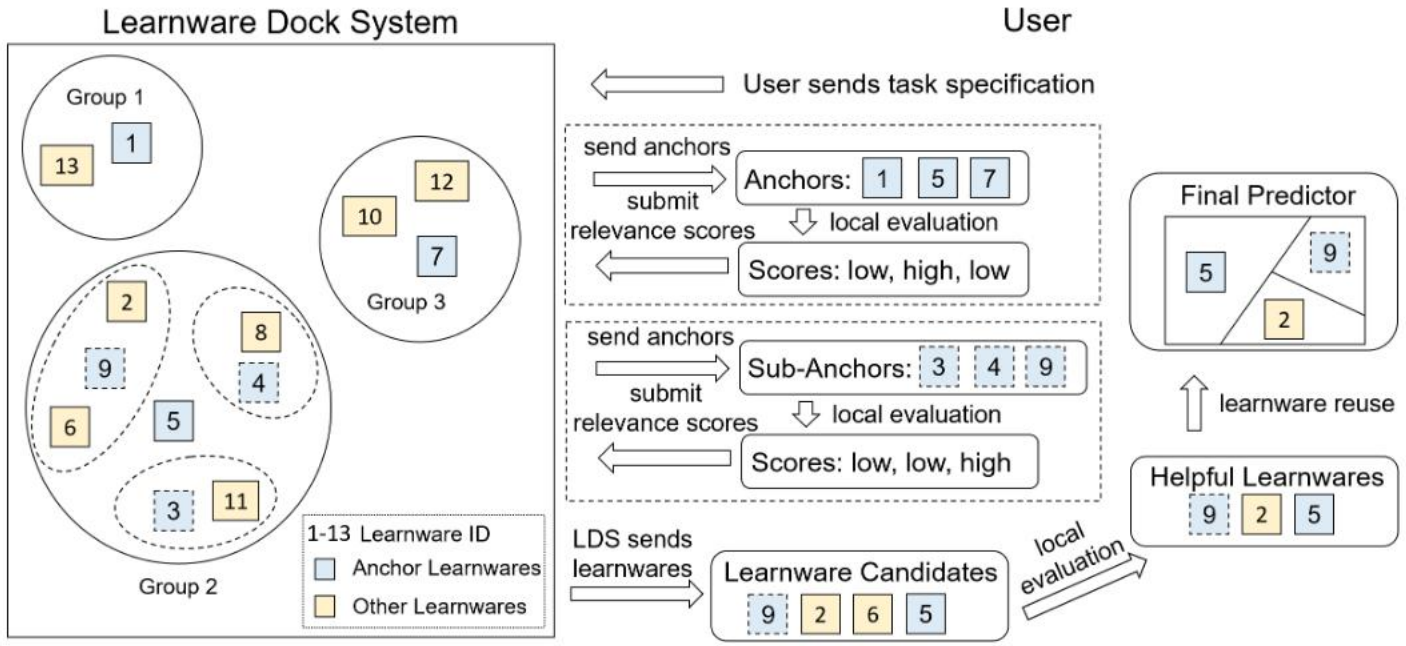
为什么要分阶段：防止「统计上很像、接口却不兼容」的模型被选中；区分「能运行」和「真合适」

规格相近 ≠ 性能保证：最接近的候选总是存在；群体覆盖不足时应公开匹配分数、拒绝推荐

全过程不接触原始数据：模型代码是隔离执行的「载荷」，规格是贯穿生命周期的持久接口协议

LDS

锚点检索：学件太多时，不必和全库逐一比对



多轮相关性反馈

① **系统给出少量锚点**
学件先分组，每组的代表（medoid）当锚点

② **用户本地评估**
在自己的私有数据上给锚点打相关分

③ **只回传高层统计**
例如「第 5 号那组最相关」，不传原始数据

④ **缩小范围再发一批**
几轮后锁定有用学件，避免穷举

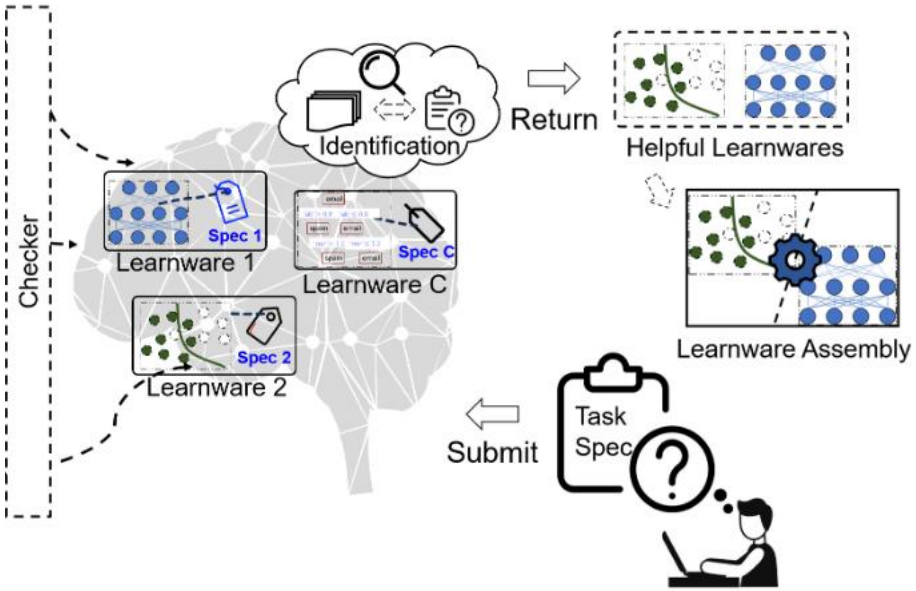
为什么需要： 学件上万时，用户规格和每个学件规格都比一次太贵；锚点让每轮只看几个代表

额外的好处： 锚点按组组织学件，相当于给学件库存提供了一层「逻辑视图」

尚未解决： 面向多学件识别与组合复用的锚点设计，论文列为未来问题

LDS

复用与组装：按「学件能覆盖用户任务多少」选择方式



覆盖程度	复用方式	什么情况 / 怎么做	适用规格
单个够用	单学件复用	任务落在某学件能力范围内；直接部署、输出校准、特征增强或参数适配	RKME / PAVE / 构造式
多个都好	多学件集成	多个学件都与用户规格匹配；简单平均或加权组合（如混合专家）	通用
单个不够 → 组装	填分布组装	用户分布 ≈ 若干学件分布的混合；二次规划定权重 → kernel herding → 选择器路由	仅 RKME
	预测链	接口互补，A 的输出接 B 的输入；如天气模型 → 作物产量模型	靠语义规格
	树组装	用少量带标签数据切分输入空间，验证提升才继续切	与统计规格无关
	按查询分派	智能体：同一任务内每个查询交给最擅长的服务	构造式

单学件复用走哪条路，取决于三件事：接口是否一致 · 用户有没有标签 · 能否访问模型参数

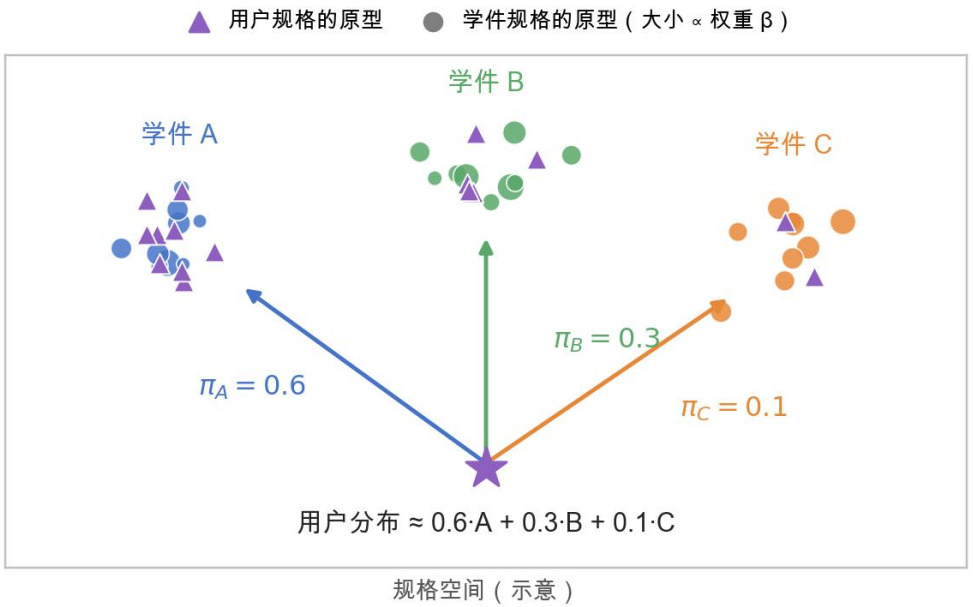
assemble ≠ ensemble：集成是同一任务的模型投票；组装是把为不同任务开发的模型拼起来，解决单个都解决不了的任务

LDS

基于 RKME 的填分布组装：按比例把几个学件「拼」成用户分布

使用场景

没有单个学件能覆盖用户任务，但用户分布 ≈ 若干学件分布的混合，例如用户人群 = 60% 像 A + 30% 像 B + 10% 像 C



规格空间（示意）

① 二次规划求混合比例 π

只用规格之间的核值：让各学件 RKME 的加权组合尽量贴近用户 RKME

$$\min_{\pi} \left\| \tilde{\mu}_u - \sum_k \pi_k \tilde{\mu}_k \right\|_{\mathcal{H}}^2, \quad \pi_k \geq 0, \ \sum_k \pi_k = 1$$

② kernel herding 生成拟样本

按 π 从各学件的原型中逐个挑点，使拟样本整体逼近用户分布；
每个点都带着「来自哪个学件」的标号

③ 训练选择器 g(x)

输入拟样本的特征，学习预测它的学件标号

④ 路由新样本

硬路由：只交给 g(x) 最大的学件；软路由：按 g(x) 加权多个学件的输出

注意： π 是整批用户数据的比例；g(x) 是单条样本的路由分数，随样本变化

局部救回： 整体不像、但局部有用的宽学件，会被整体匹配错误地拒掉 → 挑出相关原型并重新加权，找回这段重叠

为什么仅限 RKME： kernel herding 要从规格里取出原型点；PAVE 和构造式规格是参数向量，没有样本可取

增长、收缩与索引：学件越多，系统越懂行，但不越臃肿

三层状态分离

学件库存
已准入的模型

→

系统派生证据
如跨任务损失画像

→

索引
如覆盖树、LSH

准入立刻改变库存，但要等检查和交叉评估更新了证据与索引，才影响检索

下线不抹掉证据和出处，索引增量修复 → 每次变动可审计、可回滚

增长：可演化规格

- 每个学件既贡献一种能力，也贡献一个「任务」：用它的规格粗估其他模型在这个任务上的表现
- 全部算一遍 → 跨任务损失画像，不需要真实标签和原始数据
- 识别从「分布像不像」变成「在已有任务的混合上表现如何」，更能发现模型间的互补
- 学件越多样，用户任务越可能落在已有任务附近，近似误差越小

收缩：冗余过滤

- 无限准入 → 存储变大、搜索变慢，还可能遭遇重复提交攻击
- 同任务的模型表现可能不同，不同任务的模型也可能互相覆盖 → 不能只看是否重复
- 判据是「可替代」：在新学件擅长的每个任务上，老学件都擅长且输出一致
- 下线的学件在一段时间内可恢复，证据变多后可以重新判断

索引：提效

- 前提：规格距离能反映模型性能差异 → 一个代表能代表一片邻域
- 覆盖树：按规格距离建层级，像「省 → 市 → 县」逐层缩小，检索为对数复杂度
- LSH：在演化后的画像空间里哈希分桶，剪掉不必算的部分
- 替代全库扫描，并支撑上一节的分层搜索（锚点）

异构学件：接口对不上，不等于帮不上忙

先用规格学出特征空间 / 标签空间之间的映射，再走原来的识别与复用流程 —— 是扩展现有流水线，而不是另起一套检索系统

	标签空间相同	标签空间不同
特征相同	**同构：两者都一致** · 规格可以直接比较，模型可以直接接收用户输入 · 即前面几页讲的识别、复用与组装，本页作为参照	**标签异构** · 单个模型覆盖不了用户全部类别，几个合起来可以 · 按类规格：每个类别一个分量，刻画模型在该类上的能力 · 模型类别 ↔ 用户类别做匹配，用最小费用最大流，偏好互补覆盖 · 类别对应从用户数据学出，不要求类名一致
特征不同	**特征异构** · 各任务观察到不同的特征块子集 · 只凭提交的规格学一个统一子空间，不用开发者数据，也不用辅助数据 · 再用标签信息监督子空间 → 按输入–输出行为匹配 · 在 LDS 空闲时更新，像搜索引擎更新网页索引	**联合异构与改派** · 两套机制组合：投影到统一子空间，再给每个用户类别找学件 · 连语义对应都没有时：按类性能矩阵 → 标签映射 → 引导特征映射 · 把「看似无关」的模型当作黑盒决策边界来改用（结果尚属初步）

特征异构方法的演进

借助辅助数据
需要一份额外的公共数据，才能学出统一子空间

只用规格对齐
去掉辅助数据，只靠各方提交的 RKME 规格来对齐

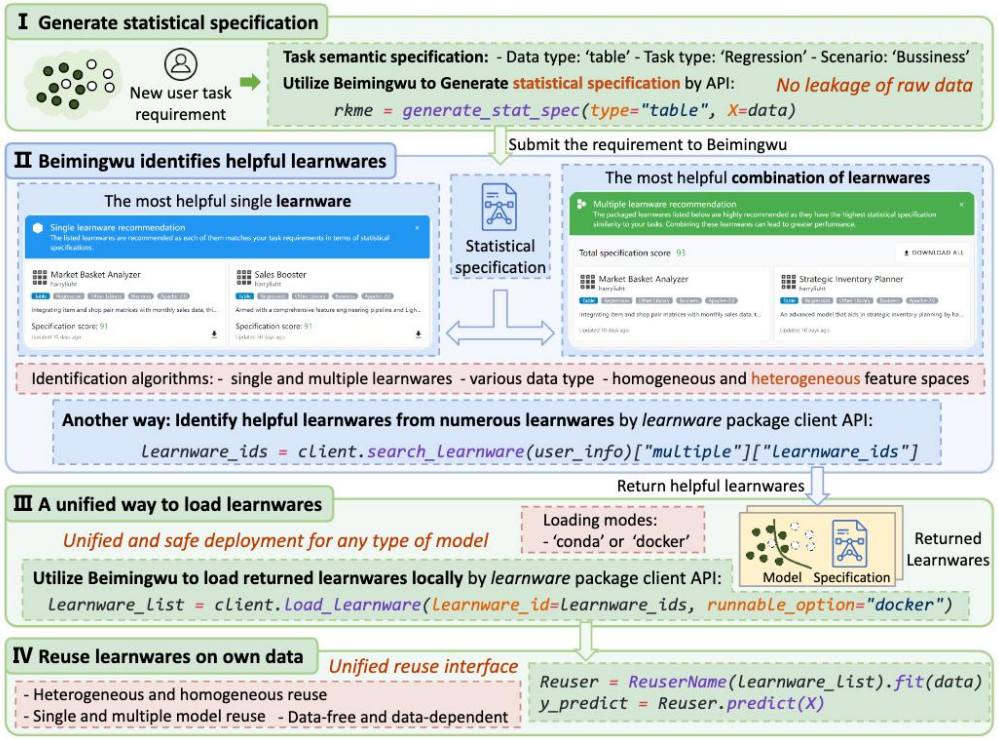
引入标签信息
把模型输出写进规格，比较输入–输出关系，而非只比输入

多学件组装
多个异构学件按样本空间的局部区域分工（树组装）

> **“特征或标签空间不匹配，说明模型在原始形式下与用户任务不兼容，而不是无法为这个任务做贡献。**
>
> 规格既能发现这种潜在能力，也提供了构造映射所需的统计信息。

Experiments

北冥坞与实验总览：从开源原型系统到三组实验



生成规格 → 识别学件 → 统一加载 → 在本地复用

≈1200 个学件已入驻北冥坞，并在持续增长

第一个开源的学件坞系统：学件包 + 核心引擎 + 后端 / 网页 / 命令行

识别与执行分离

匹配规格时不运行任何候选代码；选中后才在容器里实例化学件包

证据带出处

开发者自报的证据和系统准入时自己生成的证据分开存放

三组实验 — 通用性逐步提高

RKME	PAVE	构造式规格
I 表格学件	II 大模型学件	III 智能体学件
学件群体增长时，规格能否支撑识别、演化、收缩、高效搜索与异构	101 个 8B 专家模型，与大得多的通用模型比	136 个只能调用的黑盒智能体，按查询分派

普通模型池

按模型名查找、下载

vs

北冥坞：提交的是任务规格，而不是模型名

在不断演化的规格群体中搜索 → 返回学件包，可能是多个学件的组装

Experiments

实验一 · 表格学件：规格能否撑起整个生命周期

Table 1: Empirical evidence on tabular learnwares.

Capability	Representative evidence
Identification	Specification matching tracks the best single model; in privacy-constrained deployment it gives the lowest error among methods that do not exchange source data [26].
Evolution	Across six scenarios with 3,090 models*, evolved specifications improve every scenario; in one, error falls from 23.43% to 11.08% with one learnware and 10.35% with multiple [21].
Shrinking	Across more than 10,000 models*, capability-based filtering removes 27.91–52.73% of the population while preserving the best attainable performance and reducing search time [13].
Efficient search	In four populations of 450–1,050 models, anchors enable to examine only 11.8–21.4% of the population while remaining comparable to full traversal [30].
Heterogeneity	Specification-based mechanisms achieve the best average ranks across 30 tasks with heterogeneous feature spaces, while class-wise assembly performs best across all ten tabular populations with mismatched label spaces [23, 14].
Repurposing	Across 79 binary tasks, identified models from apparently irrelevant tasks achieve the best performance, surpassing training from scratch.

读表要点

23.43% → 11.08%

演化：误差减半

删 28%–53%

收缩：性能不降

只看 12%–21%

锚点：接近全量遍历

近 80%

异构：用户500 个标注样本胜过在本地训练

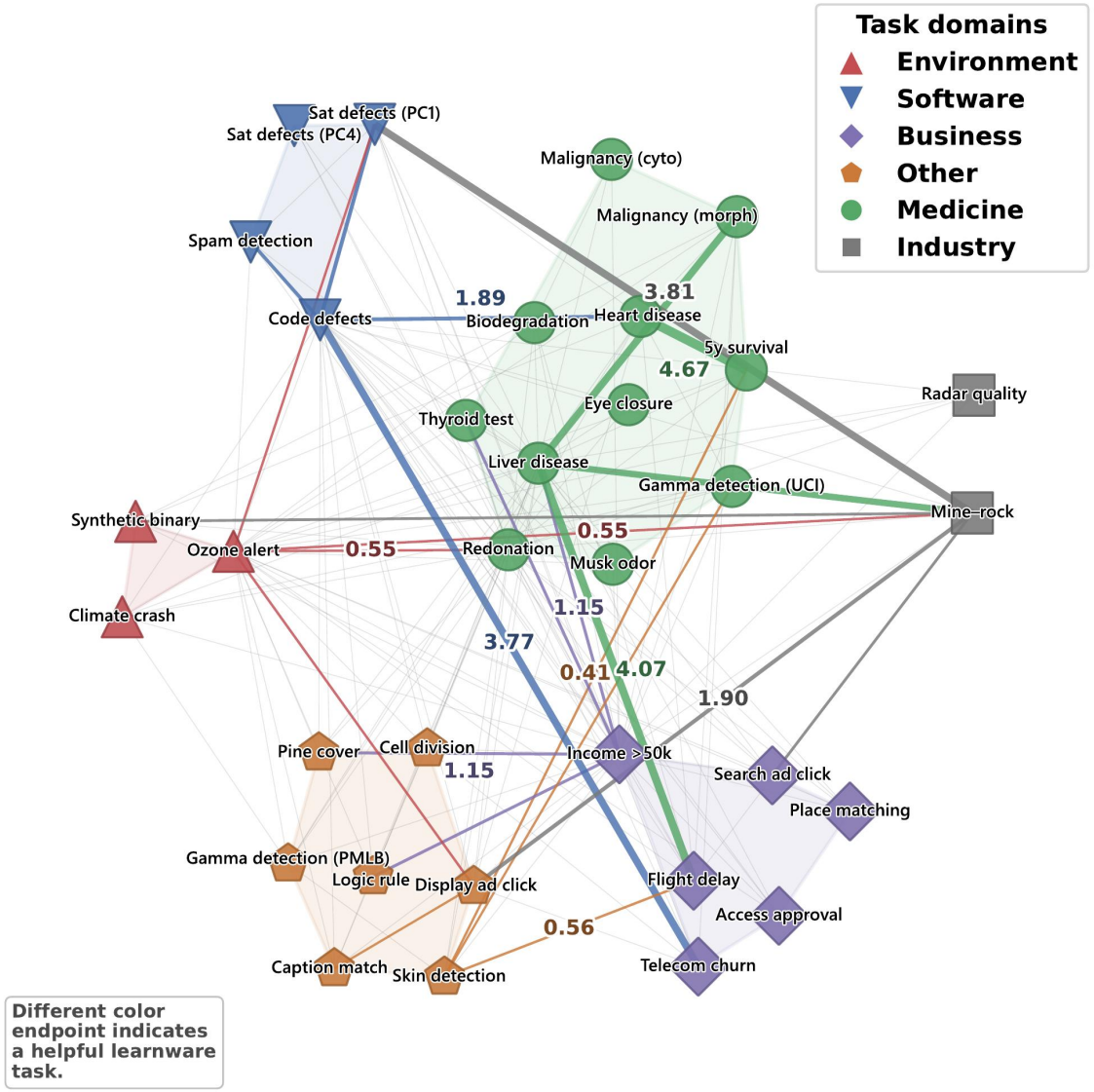
79 个任务

看似无关的模型反而最好

* 演化、收缩用的是实验构造的模型群体

Experiments

看似无关的任务：模型也能互相帮忙



图解释

节点	一个任务；形状和颜色表示所属领域
连线	任务 A 的模型能帮到任务 B；线的颜色 = B 的颜色
线宽 / 数字	帮助有多大，越粗越大

两个例子

航班延误预测 → 肝病诊断

4.07 商业 → 医学

电信用户流失 → 代码缺陷检测

3.77 商业 → 软件

有用的不是任务名字，而是模型学到的决策边界

按名字或领域检索找不到这些模型，只有比较规格才能发现

Experiments

实验二 · 大模型学件：一群 8B 专家，对比 70B 以上大模型

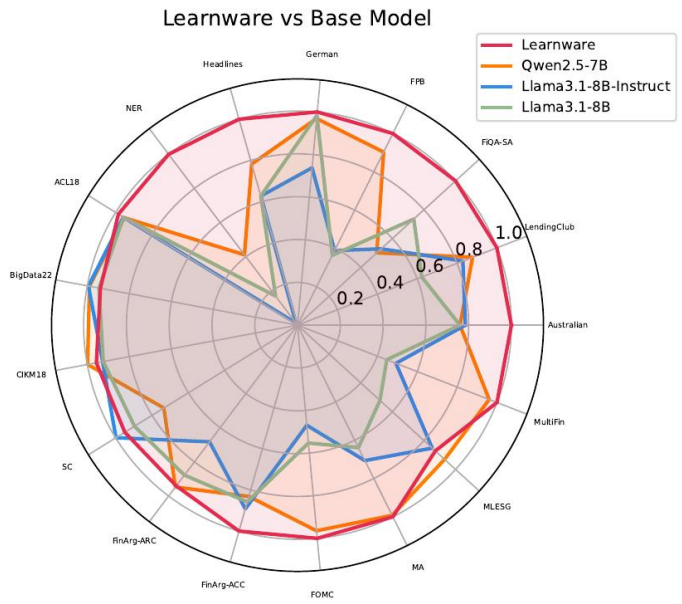
	Finance	Healthcare	Math
Best base model (7–8B)	57.61	74.02	51.68
Best large model (>70B)	58.35	74.70	**57.99**
Random specialist	56.25	74.09	51.10
Best single specialist	59.69	74.83	52.08
Consolidated single model	63.87	–	–
Learnware (ours)	**66.60**	**75.13**	52.62
Oracle (8B-scale)	67.79	75.77	54.89

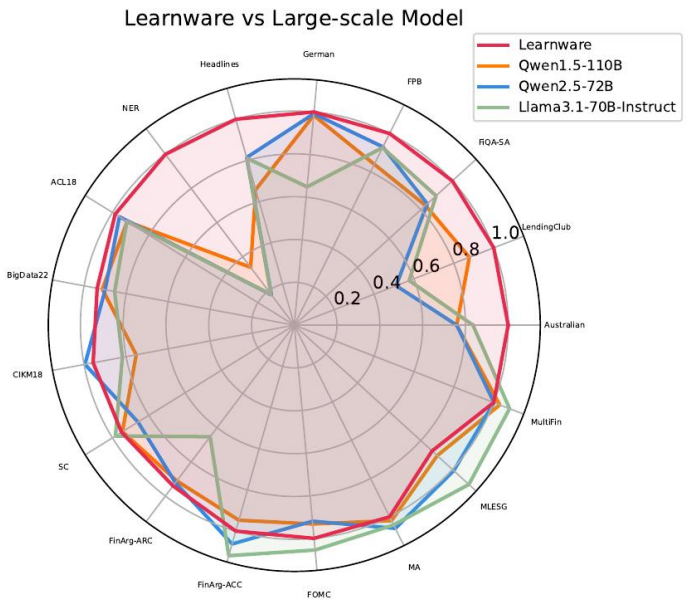
101 个 8B 专家（金融 75 / 医疗 13 / 数学 13），PAVE 规格，每个任务只挑 1 个

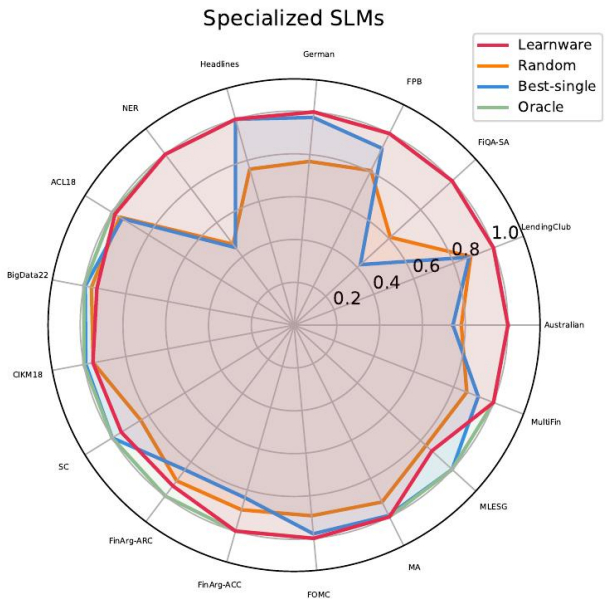
金融	医疗	数学
66.60	**75.13**	**52.62**
vs 70B+ 的 58.35	vs 70B+ 的 74.70	vs 70B+ 的 57.99，专家仅 13 个

始终超过最好的单个专家，金融、医疗上超过 70B 以上大模型，接近 Oracle 上界

图 7 读法：每条轴是一个金融子任务，越靠外越好；红线是学件方案



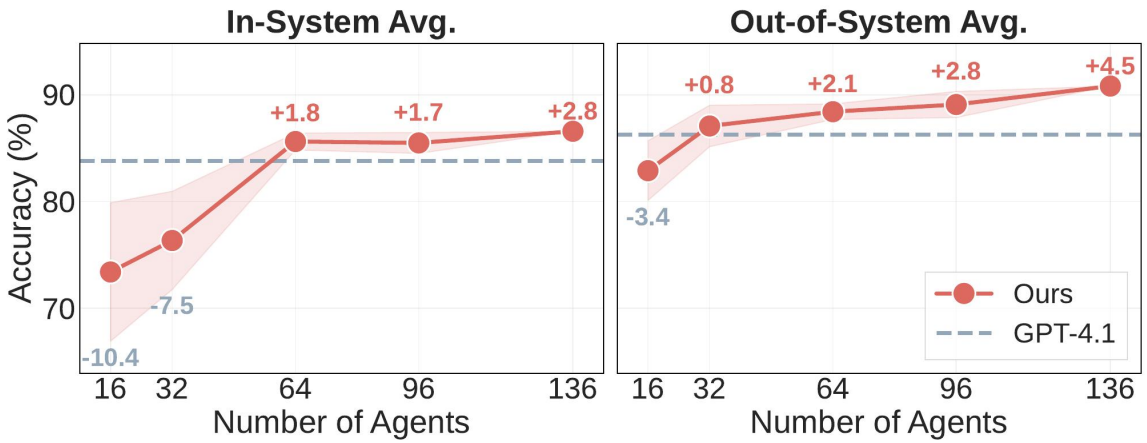




Experiments

实验三 · 智能体学件：136 个黑盒智能体，按查询分派

Method	In-system Acc.	In-system Rank	Out-of-system Acc.	Out-of-system Rank
Random selection	70.78	7.17	78.36	7.07
Description similarity	70.40	7.17	78.70	6.40
LLM chooses from descriptions	74.16	6.00	83.15	4.73
Nearest-neighbour routing [6]	80.46	4.67	82.96	5.07
Capability summaries [33]	81.89	4.33	83.92	4.87
LLM Learned agent embeddings [38]	85.05	2.67	86.82	2.60
Single general model	83.82	2.83	86.29	2.80
Agentic learnware (ours)	**86.59**	**1.00**	**90.83**	**1.47**



设置： 40 个调用在线服务 + 96 个「基座模型 + 外部工具」；系统内 6 个基准做主动测评，系统外 12 个基准子集测泛化

86.59 / 90.83

系统内 / 系统外准确率
两种设置都排第一

70.40 ≈ 70.78

按描述选 ≈ 随机选
描述写不清真实能力

+2.8 / +4.5

136 个时超过 GPT-4.1
智能体越多，准确率越高

新智能体加入只需按协议构造规格，匹配规则不变、不用重训中央选择器；构造查询规格 < 3 秒，匹配 < 0.1 秒

Thanks