

电视ABR：T阿布拉德乙炔磷大号盈利米电气电子测试系统 否伊瑞斯特否邻居

尤里·戈里什尼*† 伊万·鲁巴切夫*† 尼古拉·卡尔塔舍夫*†
丹尼尔·施伦斯基† 阿基姆·科捷尔尼科夫*† 阿特姆·巴本科†‡

一个摘要

目前，针对表格数据问题（例如分类、回归）的深度学习 (DL) 模型正受到研究人员越来越多的关注。然而，尽管最近做出了许多努力，但基于梯度提升决策树 (GBDT) 的非 DL 算法仍然是解决这些问题的有力方案。旨在提高表格 DL 地位的研究方向之一涉及设计所谓的检索增强模型。对于目标对象，此类模型从可用的训练数据中检索其他对象（例如最近的邻居），并使用它们的特征和标签做出更好的预测。

在这项工作中，我们介绍了 TabR——本质上是一个前馈网络，中间有一个自定义的类似 k-Nearest-Neighbors 的组件。在一组包含多达数百万个对象的数据集的公共基准测试中，TabR 标志着表格深度学习向前迈出了一大步：它在表格深度学习模型中展示了最佳的平均性能，成为多个数据集上的新前沿，甚至在最近提出的“GBDT 友好型”基准测试中胜过 GBDT 模型（参见图 1）。在为 TabR 提供支持的重要发现和技术细节中，主要在于类似注意力机制，该机制负责检索最近邻居并从中提取有价值的信号。除了更高的性能之外，与之前基于检索的表格 DL 模型相比，TabR 更简单且效率更高。源代码已发布：[关联](#)。

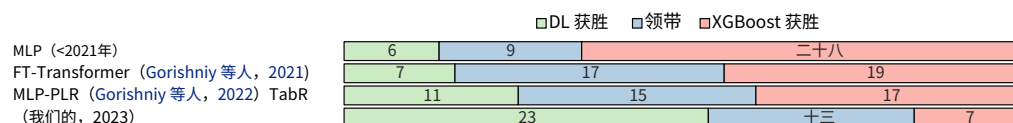


图 1：深度学习模型与 XGBoost 的比较（陈和格斯特林，2016）在 43 个中等规模的回归和分类任务上（≤50 维对象）摘自“为什么基于树的模型在典型的表格数据上仍然优于深度学习？”Grinsztajn 等人（2022）。与之前的表格深度学习模型相比，TabR 向前迈出了重要一步，并延续了该领域的积极趋势。

1 我引言

表格数据的机器学习 (ML) 问题（其中对象由一组异构特征描述）在医学、金融、制造业和其他领域的工业应用中无处不在。从历史上看，对于这些任务，基于梯度提升决策树 (GBDT) 的模型长期以来一直是首选解决方案。然而，最近，表格深度学习 (DL) 模型越来越受到关注，而且它们的竞争力也越来越强（Klambauer 等人，2017；Popov 等人，2020；王等人，2020；Hazimeh 等人，2020；黄等人，2020；Gorishniy 等人，2021；Somepalli 等人，2021；Kossen 等人，2021；Gorishniy 等人，2022）。

特别是，最近有人尝试设计一个检索增强表格深度学习模型（Somepalli 等人，2021；秦等人，2021；Kossen 等人，2021）对于目标对象，检索增强模型会从训练集中检索其他对象（例如，目标对象的最近

邻域，甚至整个训练集）并使用它们来改进对目标对象的预测。事实上，检索技术在其他领域也非常流行，包括自然语言处理（Das 等人，2021；王等人，2022；Izacard 等人，2022）、计算机视觉（贾等人，2021；Isken 等人，2022；Long 等人，2022）、CTR 预测（秦等人，2020；2021；杜等人，2022）等。与纯参数（即无检索）模型相比，基于检索的模型可以实现更高的性能，并且还表现出几个实际重要的特性，例如增量学习的能力和更好的鲁棒性（Das 等人，2021；贾等人，2021）。

虽然存在多种针对表格数据问题的检索增强模型，但在我们的实验中，我们表明，与经过适当调整的多层感知器（MLP；最简单的参数模型）相比，它们仅提供微小的优势，同时复杂度和成本却要高得多。尽管如此，在这项工作中，我们表明，考虑到某些以前被忽视的设计方面，可以获得基于检索的表格架构，该架构功能强大、简单且比以前的基于检索的模型效率高得多。我们总结了我们的主要贡献如下：

1. 我们设计了 TabR——一个简单的检索增强表格 DL 模型，该模型在一系列公共基准测试中，展示了 DL 模型中最佳的平均性能，在多个数据集上达到了新的最先进水平，并且比之前基于深度检索的表格模型效率高得多。
2. 特别地，TabR 在最近提出的中等规模任务基准上超越了 GBDT，为表格深度学习树立了一个显著的里程碑（Grinsztajn 等人，2022），最初用于说明基于决策树的模型优于深度学习模型。而基于树的模型则是一种更有效的解决方案。
3. 我们强调了注意力机制（基于检索的模型中经常使用的模块）的重要自由度，这使得设计更好的基于检索的表格模型成为可能。

2 R 兴高采烈的工作

梯度增强决策树（GBDT）。基于 GBDT 的 ML 模型是针对表格数据监督问题的非 DL 解决方案，由于其强大的性能和高效率而在社区中广受欢迎。通过采用现代 DL 构建块，特别是检索技术，我们的新模型成功与 GBDT 竞争，特别是证明了 DL 模型可以在非大数据上表现出色，在最近提出的中小型任务基准上的表现优于 GBDT（Grinsztajn 等人，2022）。

表格深度学习。Tabular DL 是一个快速发展的领域，其最新进展涵盖了参数化架构（Klambauer 等人，2017；王等人，2020；Gorishniy 等人，2021；2022）（以及许多其他），正则化（Jeffares 等人，2023）、预训练（Bahri 等人，2021）和其他方法（Hollmann 等人，2023）。在这项工作中，我们特别关注架构。特别是，最近的研究表明，类似 MLP 的主干仍然具有竞争力（Kadra 等人，2021；Gorishniy 等人，2021；2022），并且连续特征的嵌入（Gorishniy 等人，2022）显著缩小了 tabular DL 和 GBDT 之间的差距。在这项工作中，我们表明，设计合理的检索组件可以进一步提高 tabular DL 的性能。

总体而言，检索增强模型。通常，基于检索的模型设计如下。对于输入对象，首先，它们从可用（训练）数据中检索相关样本。然后，它们将输入对象与检索到的实例一起处理，以生成输入对象的最终预测。设计基于检索的方案常见动机之一是局部学习范式（博图瓦普尼克，1992），这种模型最简单的例子是 k-最近邻（kNN）算法（詹姆斯等人，2013）。基于检索的方法的前景已在各个领域得到证实，例如自然语言处理（刘易斯等人，2020；Guu 等人，2020；Khandelwal 等人，2020；Izacard 等人，2022；Borgeaud 等人，2022）、计算机视觉（Isken 等人，2022；Long 等人，2022）、CTR 预测（秦等人，2020；2021；杜等人，2022）等。此外，检索增强模型通常具有有用的特性，例如更好的可解释性（王和萨本库，2023），鲁棒性（赵和赵，2018）等。

针对表格数据问题的检索增强模型。非深度检索表格模型的经典示例是基于邻域的方法和核方法（詹姆斯等人，2013；Nader 等人，2022）还有一些基于深度检索的模型适用于（或直接设计用于）表格数据问题（Wilson 等人，2016；Kim 等人，2019；Ramsauer 等人，2021；Kossen 等人，2021；

Somepalli 等人, 2021)。值得注意的是, 其中一些方法省略了检索步骤并使用全部训练数据点作为“检索到的”实例 (Somepalli 等人, 2021; Kossen 等人, 2021; Schäfl 等人, 2022)。然而, 我们表明, 现有的基于检索的表格深度学习模型仅比简单的参数化深度学习模型略胜一筹, 而且这通常需要使用繁重的 Transformer 类架构。与之前的研究相比, 其中多层多个多头原始注意力模块经常使用 (Ramsauer 等人, 2021; Kossen 等人, 2021; Somepalli 等人, 2021), 我们的模型 TabR 仅使用一个单头注意力模块, 定制使其更适合表格数据问题。因此, TabR 的性能大大优于现有的基于检索的 DL 模型, 而且效率也更高。

3吨ABR

在本节中, 我们针对表格数据问题设计了一种新的检索增强深度学习模型。

3.1 磷补充

符号。对于给定的表格数据的监督学习问题, 我们将数据集表示为 $\{(x_i, y_i)\}_{i=1}^n$ 在哪里 $x_i \in \mathbb{R}^d$ 代表我第 i 个对象的特征和 $y_i \in \mathbb{R}$ 是代表我-日对象的标签。根据上下文, y_i 索引可以省略。我们考虑三类任务: 二分类 $Y = \{0, 1\}$ 多类别分类 $Y = \{1, \dots, C\}$ 和回归 $Y = \mathbb{R}$ 。为了简单起见, 在大多数地方, 我们假设 x_i 仅包含数值 (连续) 特征, 我们将提供额外的通信耳鼻喉科 必要时, 对二进制和分类特征进行分类。数据集被分成三个不相交的部分: $1, n = n_{\text{我火车}} \cup n_{\text{我瓦尔}} \cup n_{\text{我测试}}$ 其中“训练”部分用于训练, “验证”部分用于提前停止和超参数调整, “测试”部分用于最终评估。给定模型进行预测的输入对象称为“输入对象”或“目标对象”。

当对给定的目标对象使用检索技术时, 检索是在“上下文候选”或简称为“候选”的集合内执行的: $n_{\text{坎德}} \subseteq n_{\text{我火车}}$ 。检索到的对象又称为“上下文对象”或简称为“上下文”。目标对象也可以包含在其自己的上下文中。在本研究中, 除非另有说明, 我们对所有输入对象使用同一组候选对象, 并将 $n_{\text{坎德}} = n_{\text{我火车}}$ (即从所有训练对象中检索)。

实验设置。我们在以下文章中详细描述了我们的调整和评估协议: D.6 小节最重要的一点是, 对于任何给定的算法, 在每个数据集上, 遵循 Gorishniy 等人 (2022), (1) 我们使用验证集; (2) 对于最佳超参数, 在正文中, 我们报告了测试集合对 15 个随机种子进行平均, 并提供标准差附录 E; (3) 在比较任何两种算法时, 我们会考虑标准差, 如 D.6 小节; (4) 为了获得同一类型的模型集成, 我们将 15 个随机种子分成三个不相交的组 (即三个集成), 每组包含五个模型, 每组内取平均预测值, 并报告所得三个集成的平均性能。

在本研究中, 我们主要使用先前文献中的数据集, 并在表 1 (有时, 我们将这组数据集称为“默认基准”)。此外, 在第 4.2 节, 我们使用最近引入的中等规模任务基准 (≤ 50 对象) (Grinsztajn 等人, 2022) 据报道, GBDT 优于 DL 解决方案。

表 1: 数据集属性。“RMSE”表示均方根误差, “Acc.”表示准确度。

	中	加州	何	广告	直接投票	时赛你好	基本法	我们	一氧化碳	MI
# 对象	10000	20640	22784	48842	53940	61878	98049	166821	397099	581012
# 特征数量	7	8	16	6	6	93	28	4	118	10
# bin.features						131	3	00	1	0
# cat.features	1	00	7	3	0	04			1	四十四
度量	累计误差	RMSE	RMSE	累计误差	RMSE	累计误差	RMSE	累计均方根误差	RMSE	账户。
# 类	2	-	-	2	-	9	2	-	-	7
多数阶级	79%	-	-	76%	-	26%	52%	-	-	48%

3.2 一个建筑结构

为了构建基于检索的表格 DL 模型，我们选择了一种增量方法，从一个简单的无检索架构开始，一步一步地添加和改进检索组件。

让我们考虑一个通用的前馈无检索网络 $f(\cdot) = \text{磷}(\text{埃}(\cdot))$ 非正式划分

分为两部分：编码器埃： $\mathbf{x} \rightarrow \mathbf{R}_d$ 和预测器磷： $\mathbf{R}_d \rightarrow \hat{\mathbf{y}}$ 。为了逐步使其成为基于检索的，我们添加了检索模块 R 在剩余分支之后 埃如图所示图 2，在哪里

负责 \mathbf{R}_d 是目标对象的中间表示， $\{\tilde{\mathbf{x}}_i\}_{i \in I_{\text{cand}}}$ 是

候选人的陈述和 $\{y_i\}_{i \in I_{\text{cand}}}$ 是候选人的标签。

$\mathbf{C} \subseteq \mathbf{R}_d$ 是中间

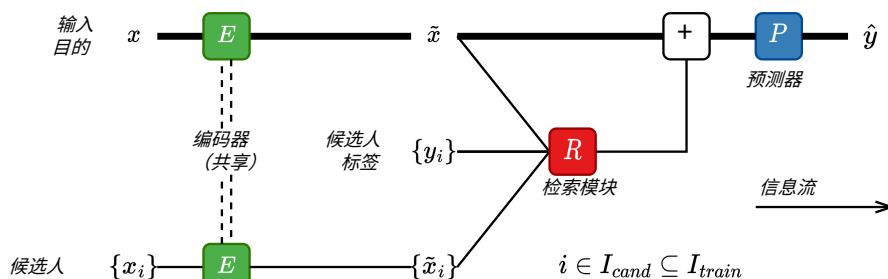


图 2：2017 年引入的基于通用检索的架构第 3.2 节并用于构建 TabR。首先，使用相同的编码器对目标对象及其检索候选对象进行编码埃。然后，检索模块 R 通过从候选中检索和处理相关对象来丰富目标对象的表示。最后，预测器磷做出预测。粗体路径突出显示了添加检索模块之前前馈无检索模型的结构 R 。

编码器和预测器。编码器埃和预测器磷模块（图 2）不是本文的重点，因此我们将其保持简单，如图所示图 3。

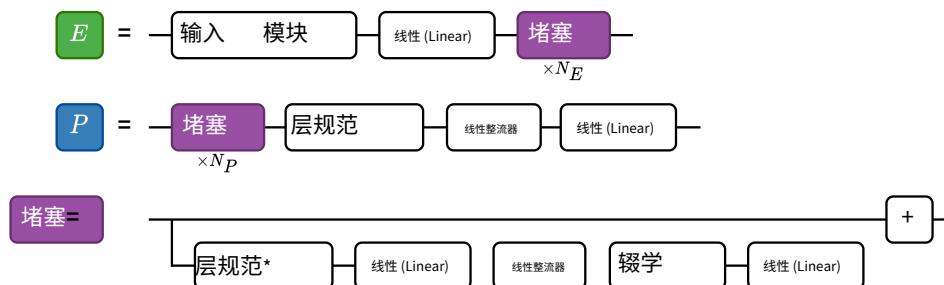


图 3：编码器埃和预测器磷引入图 2。否埃和否磷表示

堵塞模块埃和磷，分别。输入模块封装输入处理例程（特征归一化、独热编码等），并为后续的线性层组装向量输入。具体来说，输入模块可以包含连续特征的嵌入（格

奥里什内等等，2022）。（层规范在第一个中被省略堵塞的埃。

雷特曼尔莫德。我们完善检索模块 R 本着 埃如图所示，最近的邻居在图 4 图中为了清晰起见省略了以下正式细节：

1. 如果编码器埃至少包含一个 乙锁 ($\text{IE} \text{否埃} > 0$)，然后，在传递给 R ，负责 和 阿尔 升 负责 是 n 用 sha 规范化 r 层归一化 (Ba 等人，2016)。
2. 选择 离 目标 o 比 本身 可以无条件地 (即忽略顶部米手术) 添加为 (米+1)-第个对象与具有相似度得分的上下文对象集年代 ($\tilde{\mathbf{x}}, \tilde{\mathbf{x}}$)。
3. 对 softmax 函数产生的权重应用 Dropout。
4. 我们使用 米=96 (消融于 A.6 小节)，除非另有说明，我坎德=我火车。

现在，是 刺率 超过可能相似性模块的合理设计年代和价值模块五 (引入过程中，我们不使用数在图 4 图 4)。完成这个值特征的嵌入 (Gorishniy 等人，杜勒的埃 埃德埃并设置 否埃=0，否磷=2022) 在 日埃在放莫 1 (见图 3)。数控

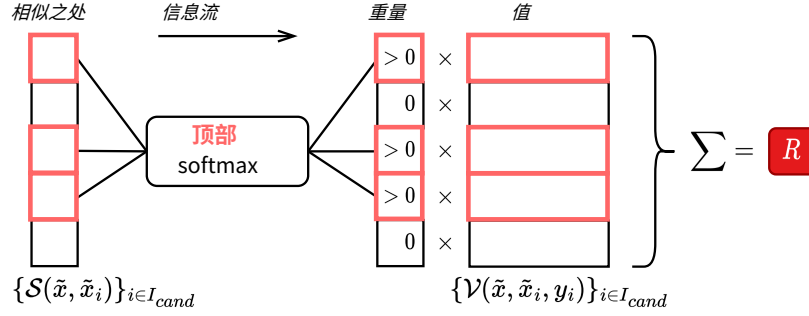


图4: 检索模块的简化图示 R (省略的细节在正文中提供)。对于目标对象的表示 \tilde{x} , 该模块采用 m 个候选人中的最近邻居 $\{\tilde{x}_i\}$ 根据相似性模块年代: $(R_d, R_d) \rightarrow R$ 并汇总价值模块产生的价值五: $(R_d, R_d, \text{是}) \rightarrow R_d$.

步骤 0. 类似香草注意力的基线。自注意力操作 (Vaswani 等人, 2017) 在先前的工作中经常被用来模拟目标对象和候选/上下文对象之间的交互 (Somepalli 等人, 2021; Kossen 等人, 2021; Schäfl 等人, 2022) 然后, 实例化检索模块 R 作为香草自我注意 (模顶米操作) 是合理的基线:

$$\text{年代}(\tilde{x}, \tilde{x}_{\text{我}}) = \text{西间}(\text{负责}) \cdot \text{电视西}(\text{负责我}) \cdot d^{-1/2} \quad \text{五}(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}}) = \text{西五}(\text{负责我}) \quad (1)$$

在哪里 西间 , 西 , 和 西五 是线性层, 目标对象被添加为 $(m+1)$ -第个对象到它自己的上下文 (即忽略顶部米操作)。据报告表 2, Step-0 配置的表现与 MLP 类似, 这意味着使用 vanilla self-attention 是一种次优策略。

步骤 1. 添加上下文标签。改进 Step-0 配置的自然尝试是利用上下文对象的标签, 例如, 通过将它们合并到价值模块中, 如下所示:

$$\text{年代}(\tilde{x}, \tilde{x}_{\text{我}}) = \text{西间}(\text{负责}) \cdot \text{电视西}(\text{负责我}) \cdot d^{-1/2} \quad \text{五}(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}}) = \text{西是}(\text{是我}) + \text{西五}(\text{负责我}) \quad (2)$$

区别在于等式 1 是下划线 添加 西是 : $\text{是} \rightarrow R_d$, 它是分类任务的嵌入表和回归任务的线性层。表 2 使用标签并没有带来任何改进, 这是违反直觉的。也许, 相似性模块年代取自 v 阿尼拉注意 做不高的咬 受益于标签这样有价值的信号。

步骤 2. 改进相似度模块年代经验上, 我们观察到, 删除查询的概念 (即删除 西间) 并使用大号₂ 距离代替点积显著提高了在几个数据集上的性能表 2 (A.1 小节提供更多讨论):

$$\text{年代}(\tilde{x}, \tilde{x}_{\text{我}}) = -\| \text{西}(\text{负责}) - \text{西}(\text{负责我}) \|_2 \cdot d^{-1/2} \quad \text{五}(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}}) = \text{西是}(\text{是我}) + \text{西五}(\text{负责我}) \quad (3)$$

区别在于等式 2 是下划线。这一变化是我们故事的一个转折点, 在之前的工作中被忽视了。至关重要, 在 A.3 小节, 我们表明, 删除三个成分中的任何一个 (上下文标签、仅键表示, 大号₂距离) 会导致性能下降到 MLP 的水平。而大号₂距离不太可能是普遍的最佳选择 (即使在表格 做主要的), 对于表格数据问题, 它似乎是一个合理的默认选择。

步骤 3. 改进价值模块五. 改进后年代在第二步, 我们转向价值模块五。激励编辑网络 R (纳德 等人, 2022) - 最近提出的 KNN 算法的推广, 我们制造五通过采用目标对象的表示来更具表现力负责考虑到:

$$\text{年代}(\tilde{x}, \tilde{x}_{\text{我}}) = -\| \text{西}(\text{负责}) - \text{西}(\text{负责我}) \|_2 \cdot d^{-1/2} \quad \text{五}(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}}) = \text{西是}(\text{是我}) + \text{电视}(\text{西}(\text{负责}) - \text{西}(\text{负责我})) \quad (4)$$

电视 $(\cdot) = \text{李} \cdot \text{德} \cdot \text{近} \quad \text{思特} \quad \text{双} \quad \text{作为} \quad (\text{Dropout}(\text{ReLU}(\text{线性}(\cdot))))$

区别在于等式 3 是下划线。表 2 表明新的价值模块进一步提高了多个数据集上的性能。直观地说, 术语 $\text{西是}(\text{是我})$ (上下文对象标签的嵌入) 可以看作是我第个上下文对象。术语 $\text{电视}(\text{西}(\text{负责}) - \text{西}(\text{负责我}))$ 可以看作是“修正”术语, 其中模块电视将 ace 转换为标签嵌入空间中不同 在关键 sp 的差异。我们提供进一步的 分析 在这新价值 模块吸部分 AS。

表 2: 检索模块实现的性能 R , 描述于第 3.2 节。如果某个数字被划线, 那么它至少比上一步的相应数字好一个标准差。与 MLP 相比, 从第 2 步开始, MLP 有了明显的改进。符号: \downarrow 对应于 RMSE, \uparrow 对应于准确性。

	中 \uparrow 加州 \downarrow 何 \downarrow 广告 \uparrow 直接投资 \downarrow 加时赛 \uparrow 你好 \uparrow 基本法 \downarrow 我们 \downarrow 一氧化碳 \uparrow															
多传感器处理器	0.	854 0.	499 3.	112 0.	853 0.	140 0.	816 0.	719 0.	697 1.	905 0.	963					
(步骤 0) 原始注意力基线	0.	8550.	484 3.	2340.	857 0.	142 0.	814 0.	719 0.	699 1.	903 0.	957	(步骤 1) + 上下文标签	(步骤 2) 新的相似性模块年代	(步骤 3) 新价值模块年代	(步骤 4) 技术调整 = 泰伯	
													860 0.	418 3.	153 0.	8580.
													140 0.	813 0.	7200.	692 1.
													804 0.	972 0.	8590	
													408 3.	1580.	863 0.	135 0.
													810 0.	722 0.	692 1.	8140.
													975 0.	8600.	403	
													3.	067 0.	8650.	133 0.
													818 0.	7220.	690 1.	747 0.
													973			

步骤 4. TabR。最后, 从经验上看, 我们观察到省略缩放项 $d_{-1/2}$ 在相似性模块中, 不将目标对象包括在其自己的上下文中, 平均而言会产生更好的结果, 如表 2。这两个方面都可以视为超参数, 上述注释可以看作我们的默认建议。我们将得到的模型称为 “TabR” (Tab-表格, R-检索)。TabR 如何实现检索模块的正式完整描述 R 如下:

$$\text{钾} = \text{西钾}(\text{负责}), k_{\text{我}} = \text{西钾}(\text{负责我}) \text{年代}(\tilde{x}, \tilde{x}_{\text{我}}) = -\|k - k_{\text{我}}\|_2 \text{五}(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}}) = \text{西是}(\text{是我}) + \text{电视}(k - k_{\text{我}}) \quad (5)$$

在哪里 西钾 是线性层, 西是 是分类任务的嵌入表和回归任务的线性层, (默认情况下) 目标对象不包含在其自己的上下文中, (默认情况下) 相似度分数不缩放, 并且 $\text{电视}(\cdot) = \text{线性无偏差}(\text{Dropout}(\text{ReLU}(\text{线性}(\cdot))))$ 。

局限性。TabR 具有检索增强模型的标准限制, 我们在附录 B。我们鼓励从业者在实践中使用 TabR 之前先检查一下其局限性。

4 埃公共基准测试

在本节中, 我们比较了 TabR (在第 3 节) 与现有的基于检索的解决方案和最先进的参数模型相结合。除了 TabR 的完全配置 (所有自由度都可用于埃和磷如下所述图 3), 我们还使用 TabR-S (“S”代表“简单”) ——一种简单的配置, 不使用特征嵌入 (Gorishniy 等人, 2022), 具有线性编码器 (否埃=0) 和一个单块预测器 (否磷=1)。我们指定了 TabR-S 何时仅用于表格、图片和标题, 而不适用于正文。有关 TabR 的其他详细信息 (包括超参数调整), 请参阅 D.8 小节。

4.1 埃评估检索-表格数据的增强深度学习模型

在本节中, 我们比较了 TabR (第 3 节) 以及现有的具有全参数深度学习模型的检索增强解决方案 (见附录 D 了解所有算法的实现细节)。表 3 表明 TabR 是唯一一款在许多数据集上比 MLP 性能显著提升的基于检索的模型。尤其是 TabR 的完整变体优于 MLP-PLR (现代参数化深度学习模型, 在 2017 年的 1000 多项式平均排名中位列第一)。Gorishniy 等人 (2022) 在几个数据集 (CA、OT、BL、WE、CO) 上取得了不错的成绩, 并且在除 MI 数据集之外的其余数据集上的表现与之相当。关于先前的基于检索的解决方案, 我们面临各种技术限制, 例如与分类问题不兼容和扩展问题 (例如, 正如我们在 A.4.1 小节, 训练 TabR 所需的时间比 NPT 少得多 (Kossen 等人, 2021) – 基于检索的最接近的竞争对手表 3) 值得注意的是, 检索组件并不是对所有数据集都具有普遍的益处。

获得的结果突出了数值特征的检索技术和嵌入 (Gorishniy 等人, 2022) (用于 MLP-PLR 和 TabR) 是两种强大的架构元素, 可改善表格 DL 模型的优化属性。有趣的是, 这两种技术并不完全正交, 但它们都无法恢复另一种技术的全部功能, 并且取决于给定的数据集, 一个人应该更喜欢检索、嵌入还是两者的组合。

主要的收获。TabR 成为表格数据问题的一种新的强大的深度学习解决方案, 并展示了基于检索的方法的良好潜力。TabR 展示了强大的平均性能, 并在多个数据集上实现了新的最先进水平。

表 3：将 TabR 与现有的检索增强表格模型和参数化 DL 模型进行比较。符号如下表 2 粗体条目是性能最佳的算法，其定义考虑了标准差，如 D.6 小节。

	中	↑	加州	↓	何	↓	广告	↑	直接投资	↓	加时赛	↑	你好	↑	基本法	↓	我们	↓	一氧化	↑	碳	↑	MI	↓	平均排名
kNN	0.	837	0.	588	3.	744	0.	834	0.	256	0.	774	0.	665	0.	712	2.	296	0.	927	0.	764	6.	0	±1. 7
DNNR (Nader 等人, 2022)	—	0.	430	3.	210	—	0.	145	—	—	—	0.	704	1.	913	—	0.	765	—	—	—	—	—	—	4. 8±1. 9
DKL (Wilson 等人, 2016)	—	0.	521	3.	423	—	0.	147	—	—	—	0.	699	—	—	—	—	—	—	—	—	—	—	—	6. 2±0. 5
素 (Kim 等人, 2019)	—	0.	472	3.	162	—	0.	140	—	—	—	0.	705	1.	902	—	—	—	—	—	—	—	—	—	4. 6±2. 5
圣人 (Somepalli 等人, 2021)	0.	8600.	468	3.	242	0.	860	0.	137	0.	812	0.	724	0.	693	1.	933	0.	964	0.	763	3.	8	±1. 5	不扩散核武器条约 (Kossen 等人, 2021)
	0.	858	0.	474	3.	175	0.	853	0.	138	0.	815	0.	721	0.	692	1.	947	0.	966	0.	753	3.	6	±1. 0
多源感知处理	0.	854	0.	499	3.	112	0.	853	0.	140	0.	816	0.	719	0.	697	1.	905	0.	963	0.	748	3.	7	±1. 三 0.
多层线性规划	860	0.	4763.	056	0.	8700.	134	0.	8190.	7290.	687	1.	860	0.	9700.	7442.	0	±1. 0	—	—	—	—	—	—	—
TabR-S	0.	8600.	403	3.	0670.	8650.	1330.	818	0.	722	0.	690	1.	747	0.	973	0.	750	—	9	±0. 7	0.	7	0.	862
泰伯	0.	4003.	1050.	870	0.	133	0.	825	0.	729	0.	676	1.	690	0.	9760.	750	—	3	±0. 6	—	—	—	—	—

4.2 碳比较电视ABR带渐变-增强决策树

在本节中，我们将 TabR 与基于梯度提升决策树 (GBDT) 的模型进行比较：XGBoost (陈和格斯特林, 2016)、LightGBM (柯等人, 2017) 和 CatBoost (Prokhorenkova 等人, 2018)。具体来说，我们比较集成（例如 TabRs 集成与 XGBoosts 集成）以进行公平比较，因为梯度提升已经是一种集成技术。

默认基准。表 4 表明，在默认基准测试中，经过调整的 TabR 在几个数据集（CH、CA、HO、HI、WE、CO）上比经过调整的 GBDT 有显著的改进，同时在除 MI 数据集之外的其余数据集上也具有竞争力。该表还表明 TabR 具有竞争力的默认配置（定义在 D.8 小节）。

基准来自 Grinsztajn 等人 (2022)。现在，我们更进一步，使用最近提出的中小型任务基准 Grinsztajn 等人 (2022) 重要的是，这个基准最初是为了说明 GBDT 在数据集上优于参数化 DL 模型，≤50 个对象，这对 TabR 来说是一个有趣的挑战。我们根据我们的调整和评估协议调整基准（参见 C.2 小节详情）并将结果报告于表 5 虽然 MLP-PLR（最好的参数化 DL 模型之一）在这组任务上确实略逊于 GBDT，但 TabR 取得了显著的进步，平均表现优于 GBDT。

附加分析：A.5 小节，我们尝试使用检索组件来增强 XGBoost；A.4 小节，我们比较了 TabR 和 GBDT 模型的训练时间和批量推理效率。

主要的收获。与 GBDT 进行比较后，TabR 确认了其作为表格数据问题新强解决方案的地位：它提供了强大的平均性能，并且可以在某些数据集上比 GBDT 提供显著的改进。

表 4：TabR 集成与 GBDT 模型集成的比较。参见 D.8 小节了解如何获得“默认”TabR-S。符号如下表 3。

	中	↑	加州	↓	何	↓	广告	↑	直接投资	↓	加时赛	↑	你好	↑	基本法	↓	我们	↓	一氧化	↑	碳	↑	MI	↓	平均排名
调整超参数																									
XGBoost	0.	861	0.	432	3.	1640.	8720.	1360.	8320.	726	0.	680	1.	769	0.	971	0.	741	2.	5	±0. 9	0.	9	0.	9
CatBoost	859	0.	426	3.	1060.	8720.	133	0.	827	0.	727	0.	681	1.	773	0.	9690.	7412.	5	±1. 1	—	—	—	—	—
轻量级GBM0.	860	0.	434	3.	1670.	8720.	1360.	8320.	726	0.	679	1.	761	0.	971	0.	741	2.	4	±0. 9	泰伯	—	—	—	—
	0.	865	0.	391	3.	025	0.	872	0.	131	0.	831	0.	733	0.	674	1.	661	0.	9770.	748	1.	3	±0. 9	—
默认超参数																									
XGBoost	0.	856	0.	471	3.	368	0.	871	0.	143	0.	817	0.	716	0.	683	1.	920	0.	966	0.	750	三.	4	±0. 9
CatBoost	861	0.	432	3.	1080.	8740.	132	0.	8220.	7260.	684	1.	886	0.	9240.	7442.	1	±0. 8	—	—	—	—	—	—	—
轻量级GBM0.	856	0.	449	3.	222	0.	869	0.	1370.	8260.	720	0.	681	1.	817	0.	899	0.	744	2.	5	±0. 9	TabR-S	—	—
S	0.	864	0.	398	2.	9710.	8590.	1310.	824	0.	724	0.	6881.	721	0.	9740.	752	2.	0	±1. 3	—	—	—	—	—

表 5：在基准上比较 DL 模型集成与 GBDT 模型集成 Grinsztajn 等人 (2022)（例如，MLP 集合与 XGBoost 集合；注意在 图 1，我们比较单身的型号不同，因此编号也不同）。请参阅 D.8 小节有关“默认”TabR-S 的详细信息。将 TabR-S 的默认配置与 GBDT 模型的默认配置进行比较。比较以成对方式进行，并考虑标准差，如中所述 D.6 小节。

	与 XGBoost 相比			与 CatBoost 相比			与 LightGBM 相比		
	胜 / 平 / 负			胜 / 平 / 负			胜 / 平 / 负		
	调整超参数								
多层感知处理器	6	11	二十六	6	8	二十九	5	11	二十七
多层线性规划	12	17	14	10	11	22	14	15	14
TabR-S	21	十三	九	17	11	15	21	15	7
泰伯	二十	六	三	23	十三	七	二十	十四	三
	默认超参数								
TabR-S	二十	十	五	17	16	10	二十	五	9

五 一分析

5.1 弗重新定义上下文以加快训练电视ABR

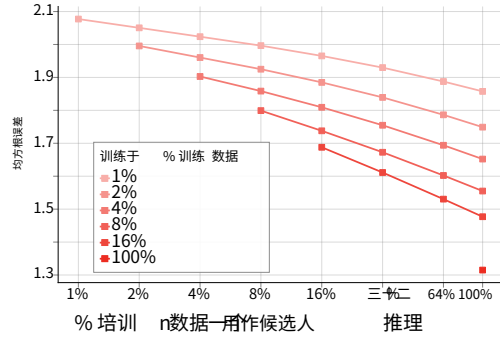
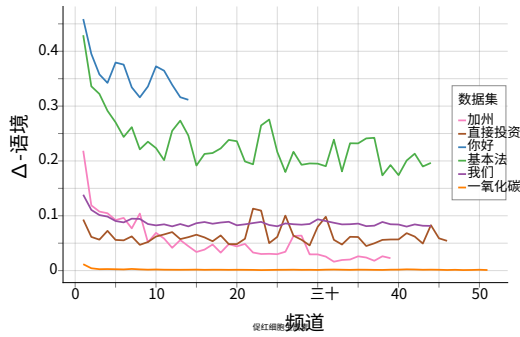
在 TabR 的普通配方中 (第 3 节)，对于每个训练批次，通过对所有候选进行编码并计算与所有候选的相似性来挖掘最新的上下文，这在大型数据集上可能非常慢。例如，在完整的“天气预报”数据集上训练单个 TabR 需要超过 18 个小时 (Malinin 等人, 2021) (300 多万对象；使用来自表 4) 然而，正如我们在图 5 对于一个普通的训练对象，它的上下文（即顶部米候选集及其根据相似性模块的分布年代）在训练过程中逐渐“稳定”，这为简单的优化提供了机会。也就是说，经过固定数量的 epoch 后，我们可以执行“上下文冻结”：即最后一次计算所有训练（但不包括验证和测试）对象的最新上下文，然后在其余的训练中重复使用这些上下文。表 6 表明，在某些数据集上，这种简单的技术可以加速 TabR 的训练，而不会造成太多指标损失，在较大的数据集上，加速效果更明显。特别是在完整的“天气预报”数据集上，我们实现了近七倍的加速（从 18 小时 9 分钟到 3 小时 15 分钟），同时保持了有竞争力的 RMSE。参见 D.2 小节了解实施细节。

表 6：TabR-S 的“上下文冻结”性能第 5.1 节。TabR-S (CF-否) 表示 TabR-S 在之后应用了上下文冻结否时期。在括号中，我们提供了与不冻结训练（最后一行）相比，训练所花费的时间比例。

	加州 ↓	直接投资 ↓	你好 ↑	基本法 ↓	我们 ↓	一氧化氮 ↑	我们 (全部) ↓
TabR-S (CF-1)	0.414 (0.72)	0.137 (0.47)	0.718 (0.80)	0.692 (0.61)	1.770 (0.57)	0.973 (0.49)	1.325 (0.13)
TabR-S (CF-4)	0.409 (0.71)	0.136 (0.51)	0.717 (0.73)	0.691 (0.62)	1.763 (0.56)	0.973 (0.59)	—
TabR-S	0.406 (1.00)	0.133 (1.00)	0.719 (1.00)	0.691 (1.00)	1.755 (1.00)	0.973 (1.00)	1.315 (1.00)

5.2 单位约会电视ABR使用新的训练数据，无需重新训练

获取新的未见过的训练数据后训练机器学习模型（例如，在收集应用程序的另一部分日志之后）是一种常见的实际场景。从技术上讲，TabR 允许利用新数据无需再培训通过将新数据添加到检索候选集来解决这个问题。我们在完整的“天气预报”数据集上测试了这种方法 (Malinin 等人, 2021) (300 万+ 对象)。图 6 表明这种“在线更新”可能是将新数据合并到已训练的 TabR 中的可行解决方案。此外，通过在数据子集上训练模型并从完整数据中检索，此方法可用于将 TabR 扩展到大型数据集。总体而言，我们认为进行的实验是初步探索，并将持续更新的系统研究留待未来工作。参见 D.3 小节了解实施细节。



图古第5:	Δ -上下文 (例如	贝洛平原	瓦 平均-
年龄	orb 训练对象	你直到凌晨	奥平 克
尽管	特拉伊bR-S.O	n 特定时期	时长为
给定	对象t, Δ -上下文	拥有该部分	其中的
孔特	七 (时) 颈部米坦率	一个及其	瓦(\ s)
昌	编辑相比于	上一期	丙 (即,
罗	厄瓦价值,	米阿勒尔河	格 埃
子	内第奥D.2用于表格	十详细信息)。	时表 0
节目	蒂 上下文更新	变得不那么	张力 埃
杜林	克日埃列车运行	ing, 其中m	蒂瓦特
操作	蒂米描述的	在小节	5.1。

图 6: Trai n英 Tab-R-S 上的变化 o美国部分
培训 d艾塔完整的“我们-其他预测
”数据集 dgr对偶添加 t时刻余
看不见的训练 数据到可以集合 迪达泰斯和-
再培训一个 s 记录在子秒 5.2。为了
每条曲线 左进正确的点 池塘不
添加任何新的 数据到最右点的集 之后
培训, d这 合 对应
添加所有 u n看 n 个训练数据 吨集合
候选人。

5.3 F鸟R热分析 年代

在 应用程序 我们提供 dea 更有洞察力 響鳴 分析。非

- 我苏A.1节，我们分析 k 埃 on 是大号相似
 of 苏基节 n3.2, w 我是一只 tur 宁 页我们的故事
 s 佩奇财务信息 神分裂症 在年代和 pe 改革 个点。深入
 吨吨 尼拉 张力 (点积 介于 查询和关键
 • 我苏基节 nA.2, w 埃分析 v 价值米 模块五引。
 r 埃格雷以公 问题，我们确认 埃 正确答案 时间的
 • 我苏子集 丙 on A.4.1, 我们 一重新交易
 b 阿基内斯 显示 帽子比较 到 pri o 基于检索
 s 特普奥沃德在任期 效率。 虽然埃 abR 相对
 米外 撞，机智 在 k o 考虑范围 页 至东 dataset 尺
 一个可 r 最 p 实际场景 ios。
 • 我苏基节 nA.8, w 埃突出显示添加 意大利技术特性

一个示例列表 es:

是莫杜勒年代介绍埃及步骤-2
西EP提供直觉 这背后
s在瓦与之相似 y 模块
s) 。
在年代 3子秒 3.2。在
时数模块电视从埃及公式 4。
R 无训练时间 埃及所有
b乌拉模型, TabR 大
s低的简单的r 埃及无需检索
l o卢梭训练时间 TabR 的
的电视。

6 碳我否联合疗捷& F优特尔 哦RK

在此苏重新模性能
吾疾翼莱斯马
rk,我们建增量吃掉题米埃堡心踢的
时大达玛中国埃堡塔布尔的张
道米学习一个架构师n一些数据d详细b
证明问题题信息检索

重赏基于val的深度学习
米开启表格数据。名称
尤里 提供强大的
我 米重要的是，我们有
特滕前 机制
公司 o网点。

n英 模型有 gr
ly,瓦我们有设计
五时俄绩效
高的点亮类似
有 显著的i

处于潜在
TabR - a
一个实现
吨和值
米公约

阿尼米波儿吗蚁目 r 未来的工作
给妈 锤等很快 呃呃 n 以及在宾夕法尼亚州
观点 s 一个另外, 在篇文章, 我们专注
普罗俄系 TabR 剩余 n 探索不足
通过上下文对象的影响来解释模型的
(我们在第 2.2 节) 架构细节方面, 可
与检索到的实例进行交互。

simp考察效率
网状结构用于十
= 莫尔埃在ta方面
d. 对于 例如，重检索

是r检索增强
一个数十万
斯克性能，所
五氮abRp的

n特德模型
数据离子
其他一些
主 提供新的
供更好的支持

R可生产性年代声明

为了使结果和模型可重复、可验证，我们提供完整的代码库、所有结果以及分步使用说明：[关联](#)。尤其是，（1）[论文报告的结果和超参数只是对提供的 URL 上提供的结果的摘要](#)（少数例外）；（2）TabR 的实现和所有基线（NPT 除外）都可用；（3）超参数调整、训练和评估管道可用；（4）超参数可用；（5）使用的数据集和分割可用；（6）超参数调整和训练时间可用；（7）使用的硬件可用；（8）[在固定环境（即固定的硬件和软件版本）内，大多数结果都是按位可重现的。](#)

R参考文献

- T. Akiba, S. Sano, T. Yanase, T. Ohta 和 M. Koyama。Optuna：下一代超参数优化框架。*知识发现*，2019年。[23](#)，[24](#)，[二十七](#)，[二十八](#)，[二十九](#)，[三十](#)
- JL Ba、JR Kiros 和 GE Hinton。层规范化。*论文集*，1607.06450v1，2016年。[4](#)
- D. Bahri、H. Jiang、Y. Tay 和 D. Metzler。Scarf：使用随机特征损坏的自监督对比学习。*国际肾病研究联合会*，2021年。[2](#)
- P. Baldi、P. Sadowski 和 D. Whiteson。利用深度学习寻找高能物理中的奇异粒子。*自然通讯*，5，2014。[20](#)
- JA Blackard 和 DJ Dean。人工神经网络和判别分析在根据制图变量预测森林覆盖类型方面的比较精度。*农业中的计算机和电子技术*，24(3): 131–151, 2000。[20](#)
- S. Borgeaud、A. Mensch、J. Hoffmann、T. Cai、E. Rutherford、K. Millican、G. van den Driessche、J. Lespiau、B. Damoc、A. Clark、D. de Las Casas、A. Guy、J. Menick、R. Ring、T. Hennigan、S. Huang、L. Maggiore、C. Jones、A. Cassirer、A. Brock、M. Paganini、G. Irving、O. Vinyals、S. Osindero、K. Simonyan、JW Rae、E. Elsen 和 L. Sifre。通过检索数万亿个标记来改进语言模型。*国际激光医学联合会*，2022年。[2](#)
- L. Bottou 和 V. Vapnik。局部学习算法。*神经计算*，4，1992。[2](#)
- T. Chen 和 C. Guestrin。Xgboost：可扩展的树提升系统。*知识发现*，2016 年。[1](#)，[7](#)
- R. Das、M. Zaheer、D. Thai、A. Godbole、E. Perez、JY Lee、L. Tan、L. Polymenakos 和 A. McCallum。基于知识库的自然语言查询案例推理。*增强型神经网络LP*，2021年。[2](#)
- K. Du、W. Zhang、R. Zhou、Y. Wang、X. Zhao、J. Jin、Q. Gan、Z. Zhang 和 DP Wipf。通过邻域传播学习表格数据的增强表示。*在神经成像与成像系统*，2022年。[2](#)
- JR Gardner、G. Pleiss、D. Bindel、KQ Weinberger 和 AG Wilson。Gpytorch：使用 GPU 加速的黑盒矩阵高斯过程推理。*在神经信息处理系统的进展*，2018 年。[二十六](#)
- Y. Gorishniy、I. Rubachev、V. Khruikov 和 A. Babenko。重新审视表格数据的深度学习模型。*神经成像与成像系统*，2021年。[1](#)，[2](#)，[20](#)，[二十八](#)，[二十九](#)
- Y. Gorishniy、I. Rubachev 和 A. Babenko。关于表格深度学习中数值特征的嵌入。*神经成像与成像系统*，2022年。[1](#)，[2](#)，[3](#)，[4](#)，[6](#)，[14](#)，[20](#)，[22](#)，[24](#)，[二十八](#)，[二十九](#)
- L. Grinsztajn、E. Oyallon 和 G. Varoquaux。为什么基于树的模型在典型的表格数据上仍然优于深度学习？*NeurIPS*，“数据集和基准”轨道，2022年。[1](#)，[2](#)，[3](#)，[7](#)，[8](#)，[20](#)，[23](#)，[24](#)，[三十](#)，[33](#)
- K. Guu、K. Lee、Z. Tung、P. Pasupat 和 M. Chang。检索增强语言模型预训练。*国际激光医学联合会*，2020年。[2](#)

- H. Hazimeh, N. Ponomareva, P. Mol, Z. Tan 和 R. Mazumder。树集成层：可微分性满足条件计算。在*国际激光医学联合会*，2020年。1
- N. Hollmann, S. Müller, K. Eggensperger 和 F. Hutter。Tabpfn：一款可在一秒钟内解决小型表格分类问题的转换器。在*国际肾病研究联合会*，2023年。2
- X. Huang, A. Khetan, M. Cvitkovic 和 Z. Karnin。Tabtransformer：使用上下文嵌入的表格数据建模。论文集，2012.06678v1，2020年。1
- A. Iscen, T. Bird, M. Caron, A. Fathi 和 C. Schmid。用于增量学习的记忆变换网络。论文集，abs/2210.04485v1，2022年。2
- G. Izacard, PSH Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel 和 E. Grave。使用检索增强语言模型进行小样本学习。论文集，abs/2208.03299v3，2022年。2
- G. James, D. Witten, T. Hastie 和 R. Tibshirani。统计学习简介。施普林格出版社，2013年。<https://www.statlearning.com/>。2
- A. Jeffares, T. Liu, J. Crabbé, F. Imrie 和 M. van der Schaar。Tangos：通过梯度正交化和特殊化来规范表格神经网络。在*国际肾病研究联合会*，2023年。2
- M. Jia, B.-C. Chen, Z. Wu, C. Cardie, S. Belongie 和 S.-N. Lim。重新思考用于视觉分类的最近邻。arXiv 预印本 arXiv:2112.08459，2021年。2
- A. Kadra, M. Lindauer, F. Hutter 和 J. Grabocka。经过良好调整的简单网络在表格数据集上表现出色。在*神经成像与成像系统*，2021年。2
- G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye 和 T.-Y. Liu。Lightgbm：一种高效的梯度提升决策树。神经信息处理系统的进展, 30: 3146–3154, 2017。7
- R. Kelley Pace 和 R. Barry。稀疏空间自回归。统计与概率快报, 33(3): 291–297, 1997。19
- U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer 和 M. Lewis。通过记忆进行泛化：最近邻语言模型。在*国际肾病研究联合会*，2020年。2
- H. Kim, A. Mnih, J. Schwarz, M. Garnelo, SMA Eslami, D. Rosenbaum, O. Vinyals 和 YW Teh。注意力神经过程。在*国际肾病研究联合会*，2019年。2, 7, 二十六
- G. Klambauer, T. Unterthiner, A. Mayr 和 S. Hochreiter。自归一化神经网络。神经信息处理系统, 2017年。1, 2
- R. Kohavi。扩大朴素贝叶斯分类器的准确率：决策树混合。在*知识发现*，1996年。20
- J. Kossen, N. Band, C. Lyle, AN Gomez, T. Rainforth 和 Y. Gal。数据点之间的自我注意：超越深度学习中的单个输入输出对。在*神经成像与成像系统*，2021年。1, 2, 3, 5, 6, 7, 16, 二十七
- PSH Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel 和 D. Kiela。知识密集型 NLP 任务的检索增强生成。神经成像与成像系统, 2020年。2
- A. Long, W. Yin, T. Ajanthan, V. Nguyen, P. Purkait, R. Garg, A. Blair, C. Shen 和 A. van den Hengel。长尾视觉识别的检索增强分类。在*计算机视觉与图像处理*，2022年。2
- I. Loshchilov 和 F. Hutter。解耦权重衰减正则化。在*国际肾病研究联合会*，2019年。22
- A. Malinin, N. Band, G. Chesnokov, Y. Gal, MJF Gales, A. Noskov, A. Ploskonosov, L. Prokhorenkova, I. Provilkov, V. Raina, V. Raina, M. Shmatova, P. Tigas 和 B. Yangel。Shifts：跨多个大规模任务的实际分布转移数据集。论文集，abs/2107.07455v3，2021年。8, 20

- Y. Nader, L. Sixt 和 T. Landgraf. Dnnr: 差分最近邻回归。在*国际激光医学联合会*, 2022年。2, 5, 7, 14
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D.库纳波 (Cournapeau), M. 布鲁彻 (M. Brucher), M. 佩罗 (M. Perrot) 和 E. Duchesnay. Scikit-learn: Python 中的机器学习。 *机器学习研究杂志《国际神经病学杂志》* 12: 2825–2830, 2011。22
- S. Popov, S. Morozov 和 A. Babenko. 用于表格数据深度学习的神经无意识决策集成。 *国际肾病研究联合会*, 2020年。1
- L. Prokhorenkova, G. Gusev, A. Vorobev, AV Dorogush 和 A. Gulin. Catboost: 使用分类特征的无偏提升。在*神经成像与成像系统*, 2018 年。7
- J.Qin, W.Zhang, X.Wu, J.Jin, Y.Fang, 和 Y.Yu. 用户行为检索用于点击率预测。 *信号*, 2020年。2
- J.Qin, W.Zhang, R.Su, Z.Liu, W.Liu, R.Tang, X.He, 和 Y.Yu. 表格数据预测的检索和交互机。 *知识发现*, 2021年。1, 2
- T. Qin 和 T. Liu. 介绍 LETOR 4.0 数据集。 *论文集*, 1306.2597v1, 2013年。20
- H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, T. Adler, DP Kreil, MK Kopp, G. Klambauer, J. Brandstetter 和 S. Hochreiter. Hopfield 网络就是您所需要的。在*国际肾病研究联合会*, 2021年。2, 3
- B. Schäfl, L. Gruber, A. Bitto-Nemling 和 S. Hochreiter. Hopular: 用于表格数据的现代 Hopfield 网络。 *论文集*, abs/2206.00664, 2022 年。3, 5, 二十七
- G. Somepalli, M. Goldblum, A. Schwarzschild, CB Bruss 和 T. Goldstein. SAINT: 通过行注意和对比预训练改进表格数据的神经网络。 *论文集*, 2106.01342v1, 2021年。1, 3, 5, 7, 二十七
- J. Vanschoren, JN van Rijn, B. Bischl 和 L. Torgo. Openml: 机器学习中的网络科学。 *论文集*, 1407.7722v1, 2014年。20
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, AN Gomez, L. Kaiser 和 I. Polosukhin. 注意力就是你所需要的一切。 *神经信息处理系统*, 2017年。5
- AQ Wang 和 MR Sabuncu. 灵活的 nadaraya-watson 头可以提供可解释和校准的分类。在*颞叶皮质激素受体*, 2023年。2
- R. Wang, R. Shivanna, DZ Cheng, S. Jain, D. Lin, L. Hong 和 EH Chi. Dcn v2: 改进的深度和交叉网络以及用于网络规模学习排名系统的实践课程。 *论文集*, 2008.13535v2, 2020年。1, 2
- S. Wang, Y. Xu, Y. Fang, Y. Liu, S. Sun, R. Xu, C. Zhu, 和 M. Zeng. 训练数据比你想象的更有价值: 一种从训练数据中检索的简单有效的方法。 *arXiv 预印本 arXiv:2203.08773*, 2022年。2
- AG Wilson, Z. Hu, R. Salakhutdinov 和 EP Xing. 深度核学习。 *美国统计局*, 2016 年。2, 7
- J. Zhao 和 K. Cho. 检索增强卷积神经网络可提高对抗样本的鲁棒性。 *arXiv 预印本 arXiv:1802.09502*, 2018 年。2

年代补充材料

一个一个额外分析

A.1 年代相似模型电视ABR

A.1.1 米动机

回想一下，在第 2 步中第 3.2 节，从 vanilla 注意力的相似性模块转变为新的 key-only 大₂驱动的相似性模块是我们故事的一个转折点，其中基于检索的模型开始在几个数据集上显示出比 MLP 明显的改进。事实上，除了实证结果（在表 2 和 A.3 小节），这个特定的相似性模块具有合理的直观动机，我们现在提供它。

- 第一的，对齐目标和候选对象的两个（查询和键）表示是优化过程的额外挑战，并且在我们的案例中没有明确的动机引入这一挑战。并且，正如 A.3 小节，避免这一挑战不仅是有益的，而且是必要的。
- 第二，在设计过程中第 3.2 节，相似性模块年代经营范围 线性变换输入（因为此时编码器埃只是一个线性层，因为我们固定了否埃=0）。然后，在原来的特征空间可能保持合理转型特征空间。对于表格数据，大₂通常比点积更好的相似度测量原来的特征空间。请注意，浅层/线性编码器的情况是特殊但非常重要的情况：因为埃在每个训练步骤中应用于许多候选者，埃最好是轻量级的以保持足够的效率。

综合起来，以上两点促使我们删除查询表示并切换到大₂距离，这导致了第 2 步中引入的相似性模块第 3.2 节。

A.1.2 一个分析候选人的注意力模式

在本节中，我们分析相似度模块年代在第 2 步中引入第 3.2 节，这大大提高了表 2。

形式上，对于给定的输入对象，相似性模块定义了候选对象的分布（“权重”图 4）恰好米+1 非零项（米是上下文大小；+1 来自在步骤 2 中将目标对象添加到其自己的上下文中）。直观地讲，这种分布的平均多样性越低，发生频率就越高不同的输入对象增强了类似的背景。在表 7，我们证明，与 vanilla 注意力相比，使用新的相似性模块，此类分布平均而言更加多样化。实现细节在 D.4 小节。

表 7：候选对象的平均分布熵（对测试对象的单个分布进行平均）。分布由相似性模块生成，如 A.1.2 小节训练好的 Step-1 和 Step-2 模型直接取自表 2。在第 2 步引入的相似性模块第 3.2 节产生更加多样化的背景。

	CH	CA	HO	AD	DI	OT	嗨我们
步骤1	6.	66.	17.	07.	15.	85.	三8. 五8. 98
步骤2	.	49.	09.	39.	710.	310.	110. 59. 5
制服8.	89.	59.	610.	210.	410.	611.	012. 6

A.1.3 目录ASE 研究

在本节中，我们考虑三个数据集，其中过渡到仅密钥大₂来自查询和键之间的普通点积的相似性模块表现出了最令人印象深刻的性能。正式地说，这是从“步骤 1”到“步骤 2”的过渡表 2 对于这三个数据集中的每一个，首先，我们注意到，对于给定的输入对象，存在一个特定于领域的“好邻居”概念，即从人类角度来看与输入对象非常相关的邻居

并为更好地预测输入对象提供强有力的提示。然后，我们展示了新的相似性模块允许查找和利用这些自然提示。

加州住房 (CA)。在这个数据集上，从“普通”的查询和键之间的点积相似性模块到仅有键的大号相似性模块带来了显著的性能提升，如“步骤 1”和“步骤 2”之间的差异所示表 2。在这个数据集上，任务是估算加利福尼亚州的房价。直观地说，对于测试集中的给定房屋，地理邻域内训练房屋的价格应该是解决任务的有力提示。此外，特征之间有坐标（经度和纬度），这应该可以简化寻找好邻居的过程。“步骤 2”模型成功做到了这一点，而“步骤 1”模型则并非如此。具体来说，对于一个普通测试对象，“步骤 2”模型将大约 7% 的注意力集中在对象本身上（回想一下，“步骤 2”将目标对象包含在上下文对象中），将大约 77% 的注意力集中在 10 公里半径内的上下文对象上。“步骤 1”模型的相应数字分别为 0.07% 和 1%。

天气预报 (WE)。这里的故事看似与上一段分析的 CA 数据集的故事相似，但实际上有很大的不同。同样，对于给定的测试数据点，数据集包含来自训练集的地理邻居形式的自然提示，这些提示可以为测试查询做出更好的天气预报；而“第 2 步”模型（表 2）成功利用了这一点，而“步骤 1”模型无法对这些提示给予任何有意义的关注。具体来说，对于一个普通物体，“步骤 2”模型将大约 29% 的注意力集中在物体本身上（回想一下，“步骤 2”将目标物体包含在上下文物体中），将大约 25% 的注意力集中在 200 公里半径内的上下文物体上。“步骤 1”模型的相应数字分别为 0.25% 和 0.5%。然而，与 CA 案例有一个关键的区别：在我们使用的数据集 WE 版本中，该特征不包含坐标换句话说，为了进行分析，后在训练中，我们从原始数据集中恢复了每一行的原始坐标，并观察到模型学到了“好邻居”的“正确”观念与其他特征的区别。

Facebook 评论量 (FB)。在本文中，我们第一次提到这个数据集，它曾在先前的工作中使用过 (Gorishniy 等人, 2022)，我们也在这个项目中使用了一段时间。值得注意的是，在这个数据集上，TabR 表现出了比竞争对手（包括 GBDT 和一流的参数化 DL 模型）不可思议的进步。然后我们注意到一个奇怪的模式：通常，对于给定的输入，TabR 会将异常高比例的注意力集中在一个上下文对象上（每个输入对象都有一个不同的上下文对象）。这就是我们发现从 Gorishniy 等人 (2022) 包含一个“泄漏”：粗略地说，对于许多对象，可以在训练集中找到它们的几乎完全相同的副本，并且通过这种提示，任务变得非常简单。实际上，对于 TabR 来说，这非常简单，但对于其他模型来说并非如此。具体来说，对于一个普通对象，“步骤 2”模型将大约 20% 的注意力集中在对象本身上（回想一下，“步骤 2”在上下文对象中包含目标对象），将大约 35% 的注意力集中在其泄漏的几乎副本上。“步骤 1”模型的相应数字分别为 0.5% 和 0.09%。

A.2 一个分析价值模块电视ABR

在本节中，我们分析价值模块五TabR（参见等式 5）。

A.2.1 主键动机

正式地，我们注意到价值模块的输出五香草注意力（定义在等式 3，那是，前步骤 3 修改）不依赖于目标对象表示负责这为五更具表现力负责考虑到这一点。虽然有很多技术方法可以利用这个机会，但我们决定从 DNNR 中汲取灵感 (Nader 等人, 2022) – 最近提出的针对回归问题的 kNN 算法的泛化。

从概念上来说，虽然 kNN 仅捕获局部标签分布，但 DNNR 还捕获局部趋势（正式称为导数）。技术与 kNN 相反，在 DNNR 中，邻居不仅对预测有贡献，还包括一个额外的校正项，该校正项取决于原始特征空间中目标对象和邻居之间的差异。

这就是我们到达**等式 4**—新价值模块五。与 DNNR 类似，该模块由两个项构成其输出：

1. 上下文对象的标签的嵌入。
2. “修正”一词，其中模块 *电视* 将密钥空间中的差异转化为标签嵌入空间中的差异。

A.2.2 问量化“更正”语义

回想一下价值模块的正式定义 **五**TabR（参见**等式 5**）：

$$\text{五}(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}}) = \text{西}_{\text{是}}(\text{是我}) + \text{电视}(k - k_{\text{我}}) = \text{西}_{\text{是}}(\text{是我}) + \text{电视}(\Delta \text{钾}_{\text{我}}) \quad (6)$$

直观地讲，对于给定的上下文对象，其标签 *是我* 可以成为预测的重要贡献部分。让我们考虑回归问题，其中**等式 6**， $\text{是我} \in \mathbb{R}$ 嵌入于 *西是到*

$\tilde{Y} \subset \mathbb{R}^d$ 。自从 *西是* 是线性层， \tilde{Y} 只是一条线，这条线上的每个点都可以映射回到相应的标签 *R*。然后，校正项的投影 *电视*($\Delta \text{钾}_{\text{我}}$) \tilde{Y} 可以翻译为上下文标签的修正 *是我*：

$$\text{五}(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}}) = \text{西}_{\text{是}}(\text{是我}) + \text{项目} \tilde{Y} \text{电视}(\Delta \text{钾}_{\text{我}}) + \text{项目} \tilde{Y} \perp \text{电视}(\Delta \text{钾}_{\text{我}}) = \text{西}_{\text{是}}(\text{是我} + \Delta \text{是我}) + \text{项目} \tilde{Y} \perp \text{电视}(\Delta \text{钾}_{\text{我}}) \quad (7)$$

检查下划线校正项 $\text{proj}_{\tilde{Y}} \text{电视}(\Delta \text{钾}_{\text{我}})$ （或者 $\Delta \text{是我}$ ）很重要，我们采取 *受过训练*TabR，并重新评估它 *无需再培训* 忽略这个投影（这相当于设置 $\Delta \text{是我} = 0$ ）。作为基准，我们还尝试忽略 *电视*($\Delta \text{钾}_{\text{我}}$) 随机

一维子空间而不是 \tilde{Y} 。**表 8** 表明沿 \tilde{Y} 的校正对模型起着至关重要的作用。实现细节在 **D.5 小节**。

表 8：评估训练后的 TabR-S 的 RMSE，同时忽略 *电视*($\Delta \text{钾}_{\text{我}}$) 在不同的一维子空间上，如下所述**A.2 小节**第一列显示了从中移除一维子空间的投影 *电视*($\Delta \text{钾}_{\text{我}}$)。第一行对应的不是

删除任何投影（即未修改的 TabR-S）。忽略 \tilde{Y} （标签嵌入空间）上的投影会破坏模型，而忽略随机投影则不会产生太大影响。

	加州 ↓ 何 ↓ 直接投资 ↓ 基本法 ↓ 我们 ↓
—	0. 403 3. 067 0. 133 0. 690 1. 747
随机的	0. 403 3. 071 0. 133 0. 690 1. 754
\tilde{Y}	0. 465 3. 649 0. 364 0. 695 2. 003

对于分类问题，我们测试了类似的假设，但没有得到任何有趣的结果。也许，价值模块五尤其是 *电视* 模块应该采用不同的设计来更好地模拟分类问题的性质。

A.3 一个膨胀研究

回想一下，在第 2 步**第 3.2 节**，我们提到，至关重要的是 *全部* 与步骤 0 相比，步骤 2 发生了变化（使用标签 + 不使用查询 + 使用大 ℓ_2 距离而不是点积）对于在多个数据集上提供比 MLP 更明显的改进非常重要。请注意，不使用查询相当于共享 *西问* 和 *西钾*： $\text{西问} = \text{西钾}$ 。**表 9** 包含相应实验的结果，并且确实证明了步骤 2 的配置不能在不损失指标的情况下轻易简化（参见 CH、CA、BL、WE 数据集）。

总的来说，我们假设两件事都很重要：附加信号有多有价值（步骤 1）以及我们如何测量从目标物体到该有价值信号源的距离（步骤 2）。

A.4 埃效率

A.4.1 碳比较训练时间

虽然 TabR 表现出色，但这些好处并非毫无代价，因为与所有检索增强模型一样，TabR 的检索组件也会带来额外的开销。在这种情况下

表 9：消融研究A.3 小节。西问=西种表示仅使用键而不使用查询。步骤 2 是唯一在 CH、CA、BL、WE 数据集上提供比 MLP 明显改进的变体。

大号, 宽问=西种, 宽是			中	加州	主任	直接投	时赛你好	基本法我们	平均排名							
多层感知处理器			0.	854 0.	499 3.	112 0.	853 0.	140 0.	816 0.	719 0.	697 1.	905	2.	4 ±1.	4	
步骤-0	✗	✗	✗	0.	855 0.	484 3.	234 0.	857 0.	142 0.	814 0.	719 0.	699 1.	903	2.	4 ±0.	9
步骤1	✗	✗	✓	0.	855 0.	489 3.	205 0.	857 0.	142 0.	814 0.	719 0.	698 1.	906	2.	4 ±1.	2
	✗	✓	✗0.	853 0.	495 3.	178 0.	857 0.	143 0.	808 0.	719 0.	698 1.	903	2.	9 ±0.	8	
	✗	✓	✓	0.	857 0.	495 3.	217 0.	857 0.	141 0.	808 0.	717 0.	698 1.	881	2.	7 ±0.	7
	✓	✗	✗0.	855 0.	488 3.	170 0.	857 0.	143 0.	813 0.	719 0.	698 1.	901	2.	3 ±1.	0	
	✓	✗	✓	0.	856 0.	498 3.	206 0.	858 0.	142 0.	812 0.	721 0.	699 1.	900	2.	4 ±1.	1
	✓	✓	✗0.	856 0.	442 3.	154 0.	856 0.	141 0.	811 0.	722 0.	698 1.	896	2.	0 ±0.	7	
步骤2	✓	✓	✓	0.	860 0.	418 3.	153 0.	858 0.	140 0.	813 0.	720 0.	692 1.	804	1.	2 ±0.	4

部分中，我们旨在通过比较 TabR 的训练时间与所有基线的训练时间来确定这种开销。表 10 显示了两件重要的事情：

- 首先，TabR 比之前的检索增强表格模型效率更高（即在下游性能和训练时间之间提供了更好的权衡）。特别是，TabR 比 NPT 效率更高（有时甚至更高）。Kossen 等人, 2021) – 根据检索得出的最接近的竞争对手表 3。
- 其次，在考虑的数据集大小范围内，TabR 的绝对训练时间在实践中是可以承受的。此外，报告的执行时间是使用我们的简单实现实现的，甚至缺少一些基本的优化。

综上所述，与基于检索的表格深度学习的先前工作相比，TabR 在效率方面取得了很大进步。TabR 相对于简单模型（GBDT、参数化深度学习模型）相对较慢，提高其效率是一个重要的研究方向。然而，考虑到技术优化的空间以及类似于上下文冻结的技术（第5.1节），基于检索的表格深度学习的前景看好。

表 10：调整模型的训练时间（来自表 3，表4和表 6）对随机种子取平均值。格式为时:分:秒。TabR-S (CF-4) 是具有上下文冻结功能的 TabR-S（第5.1节）在四个时期后应用。颜色描述了以下非正式层级：

■ <5 分钟 ■ <30 分钟 ■ <2小时 ■ <10 小时 ■ >10 小时

	中	加州	何	广告	直接投	时赛	你好	基本法	我们	一氧化碳	MI
XGBoost	0:00:01	0:00:20	0:00:05	0:00:05	0:00:02	0:00:35	0:00:15	0:00:08	0:02:02	0:01:55	0:03:43
轻量级GBM	0:00:00	0:00:04	0:00:01	0:00:01	0:00:03	0:00:34	0:00:10	0:00:07	0:06:40	0:06:22	0:06:45
多层感知处理器	0:00:02	0:00:18	0:00:09	0:00:17	0:00:15	0:00:31	0:00:24	0:01:38	0:00:29	0:04:01	0:02:09
多层线性规划	0:00:03	0:00:43	0:00:14	0:00:24	0:00:25	0:02:09	0:00:17	0:00:52	0:20:01	0:03:32	0:30:30
检索增强模型											
TabR-S (CF-4)	0:00:08	0:00:25	0:00:30	0:00:34	0:00:43	0:00:57	0:01:02	0:03:08	0:09:08	0:23:13	-
TabR-S	0:00:20	0:01:20	0:01:23	0:03:04	0:01:44	0:01:17	0:02:09	0:11:22	0:12:11	0:49:59	0:55:04
泰伯	0:00:16	0:00:40	0:00:55	0:01:30	0:01:24	0:01:47	0:06:22	0:04:14	1:03:18	0:37:03	1:46:07
丹麦航空	-	0:06:15	0:03:55	-	0:21:59	-	-	1:04:10	-	-	-
心钠素	-	0:37:40	0:42:16	-	2:14:38	-	-	1:32:27	6:00:11	-	-
圣	0:00:23	0:06:04	0:01:44	0:00:58	0:01:55	0:05:37	0:03:47	0:06:22	2:55:51	6:17:20	5:39:37
《不扩散核武器条约》	0:08:44	0:06:58	0:12:21	0:11:22	0:54:55	10:45:42	3:26:47	0:55:04	5:28:56	12:05:28	8:07:36

A.4.2 目录比较推理效率：TabR对比. 陣容东南欧

在本节中，我们比较了TabR和XGBoost的推理效率。重要的是，我们当前对 TabR 的实现还很幼稚，甚至缺乏基本的优化。

表 11表明推理速度大致相当。更细微的观察如下：

- 在“非简单”任务（CA、OT、WE、CO）上，TabR 速度更快（非正式地，“非简单”意味着 XGBoost 需要许多树和/或 XGBoost 需要高深度和/或数据集具有更多特征）。
- 在“简单”任务上，XGBoost 速度更快（非正式地说，“简单”意味着 XGBoost 很浅且/或数据集特征很少）。

随着训练规模的增长（例如参见 MI 数据集表 11），TabR 可能会因为检索而变得更慢，但是仍有很大的优化空间：

- 缓存候选密钥表示，而不是在每次前向传递时重新计算它们。
- 在 float16 而不是 float32 中执行搜索。
- 使用近似搜索技术代替当前的强力搜索技术。
- 仅使用训练数据的子集作为候选。
- ETC。

表 11：经过调整的 TabR-S 和 XGBoost 的推理吞吐量（批大小为 4096）表 4 在 NVIDIA 2080 Ti 上。最后一行报告了 XGBoost 吞吐量与 TabR-S 吞吐量之间的比率。

	中	加州	何	广告	直接投资	加时赛	你好	基本法	我们	一氧化碳	MI
# 对象	6400	13209	14581	26048	34521	39601	62751	106764	296554	371847	723412
# 特征	11	8	16	14	9	93	二十八	9	119	54	136
XGBoost #树	121	3997	1328	988	802	524	1040	1751	3999	1258	3814
XGBoost 最大树深度	9	7	10	十三	十三	11	8	十三	12	12	12
XGBoost 吞吐量 (对象/秒)	2197k	3.3万	17.9万	131千	417千	19千	72千	8.4万	15千	10千	14千
TabR-S 吞吐量 (对象/秒)	3.5万	3.5万	5.5万	3.3万	43千	4万	3.7万	27千	3.4万	23千	11千
开销	62.3	0.9	3.3	3.9	9.6	0.5	1.9	3.1	0.5	0.4	1.2

A.5 一个增强超大容量硬盘带有检索组件的 OOST

在 TabR 成功结果于第 4.2 节，我们尝试使用一个简单的检索组件来增强 XGBoost，以确保我们不会错过这个改进基线的机会。也就是说，对于给定的输入对象，我们发现米=96（在每个原始特征空间中，选取“平均邻居”的最小向量（即等于 TabR 上下文大小）作为最近的训练对象，对它们的特征和标签取平均值（回归问题中取原有的标签，分类问题中取独热编码表示），将目标对象的特征与“平均邻居”的特征和标签连接起来，得到的向量作为 XGBoost 的输入。结果表 12 表明该策略不会为 XGBoost 带来任何明显的收益。我们尝试改变邻居的数量，但没有取得任何显著的改进。

表 12：调整模型集合的结果。“XGBoost + 检索”代表使用“平均邻居”特征和标签增强的 XGBoost，如中所述 A.5 小节。

	中	加州	何	广告	直接投资	加时赛	你好	基本法	我们	一氧化碳	MI	平均排名
XGBoost	0.	861 0.	432 3.	1640.	8720.	1360.	8320.	726 0.	680 1.	769 0.	9710.	7411. 9 ± 0. 7
XGBoost + 检索	0.	855 0.	436 3.	134 0.	871 0.	133 0.	815 0.	724 0.	687 1.	788 0.	962 0.	743 2. 5 ± 0. 5 泰伯
	0.	865 0.	391 3.	025 0.	872 0.	131 0.	831 0.	733 0.	674 1.	661 0.	9770.	748 1. 2 ± 0. 6

A.6 高度 OW 的表现电视 ABR 取决于上下文大小米？

回想一下，在整篇论文中，我们使用了固定的米=96 作为 TabR 的上下文大小（邻居数量）。我们评估米在表 13。至关重要的是，选择米必须根据表现进行验证集合（不在测试测试）。结果表明 米=96 是一个合理的默认值。

A.7 一个附加结果 “**背景冻结**” 技术

我们报告扩展结果第 5.1 节在图 7，表 14 和表 15. 对于正式的定义 Δ -上下文度量，参见 D.2 小节。

表 13: 数据集的平均排名表 1 具有不同值的默认 TabR-S 米。

	m=1	m=2	m=4	m=8	m=16	m=32	m=64	m=96	m=128	m=256
瓦尔德数据集										
平均排名	5	4.5	4	3.25	2.75	2.12	2.12	1.88	1.88	1.88
等级标准	2.4	5	1.87	1.73	1.71	1.56	1.45	1.45	0.93	1.05
电视数据集										
平均排名	4.1	2	3.62	3.5	3.12	2.38	2.12	2	1.62	1.75
等级标准	2.2		1.8	1.5	1.05	1.41	1.27	1.41	0.99	0.97

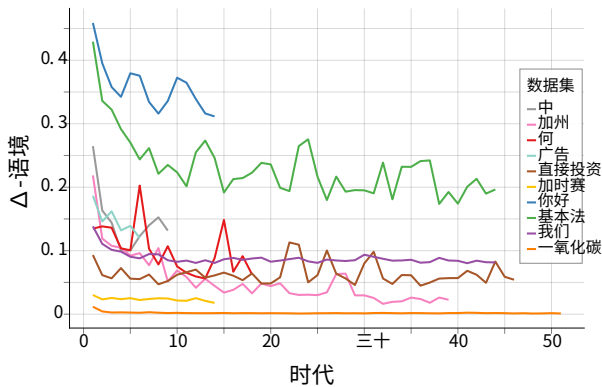


图 7 : 扩展版本 我们图 5 和 更多数据集。

桌子 14: 扩展的 veized 的表 6. 免费提供教 0 个纪元后 意味着冻结 随机地
最初模型。特定的 育 n 表 15

	中	加州	何	广告	直接投资	加时赛	你好	基本法	西	埃	一氧化碳	我们 (全部)	平均排名
米大泥	0.854	0.499	0.311	0.085	0.140	0.816	0.719	0.697	1.905	0.963	-	-	2.9 ± 1.5
塔bRS (CF-0)	0.8570	0.443	0.075	0.8570	0.137	0.816	0.7180	0.700	1.787	0.969	1.387	-	2.3 ± 1.4
塔bRS (CF-1)	0.8560	0.443	0.065	0.8560	0.137	0.816	0.7180	0.692	1.077	0.973	1.325	-	1.8 ± 1.0
塔bRS (CF-2)	0.8560	0.41	0.074	0.8560	0.137	0.8160	0.7180	0.6911	0.7670	0.973	-	-	1.7 ± 0.8
塔bRS (CF-4)	0.8580	0.40	0.087	0.8570	0.136	0.8160	0.7170	0.6911	0.7630	0.973	-	-	1.3 ± 0.5
塔bRS (CF-8)	0.8580	0.40	0.073	0.1180	0.8570	0.135	0.8170	0.7190	0.6911	0.7610	0.973	-	1.3 ± 0.5
塔b遥感	0.8590	0.40	0.063	0.093	0.8580	0.133	0.8160	0.7190	0.6911	0.7550	0.973	1.315	1.0 ± 0.0

桌子 15: 冻结时间的分 埃训练 i 所花的时 n 表 14, 相对: 训练时间 没有
孔特 撒 (最后一个 瓦间; 格式为小时 s 分: 秒) 。

	中	加州	何	广告	直接投资	加时赛	你好	乙大号	我们	合作	(满的)
TabR-S (CF-0)	0.96	0.78	0.79	0.83	0.75	0.87	1.03	0.64	0.53	0.52	0.13
TabR-S (CF-1)	0.88	0.72	0.89	0.89	0.80	0.80	0.61	0.57	0.49	0.13	0.13
TabR-S (CF-2)	0.94	0.65	0.78	0.83	0.86	0.82	0.63	0.57	0.60	-	-
TabR-S (CF-4)	1.01	0.71	0.73	0.51	0.97	0.73	0.62	0.56	0.59	0.76	0.71
TabR-S (CF-8)	0.82	0.61	0.90	0.78	0.67	0.59	0.59	-	-	-	-
TabR-S	0:00:08	0:00:36	0:00:42	0:00:46	0:01:25	0:00:58	0:01:24	0:05:03	0:16:19	0:39:13	18:08:39

A.8 A附加技术说明电视ABR

我们重点介绍 TabR 的以下技术方面：

1. 由于第 3 步中引入的变更第 3.2 节，价值表述五 $(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}})$ 无法为训练模型预先计算候选对象，因为它们取决于目标对象。这意味着在将模型部署到生产环境时，内存使用量大约减少一半（因为只需为训练对象部署关键表示和标签），但五 $(\tilde{x}, \tilde{x}_{\text{我}}, y_{\text{我}})$ 必须在运行时计算。
2. 尽管检索模块具有与注意力机制类似的特性 R 与之前的工作相反，TabR 不会受到候选对象数量的二次复杂度的影响，因为它只计算目标对象的注意力，而不计算上下文对象的注意力。
3. 在等式 4，电视用途无偏差线性在其定义中。严格来说，从表达能力的角度来看，添加一个偏差（即使用线性）将是多余的，因为 $\tilde{x}_{\text{我}}$ 在等式 4。为了以防万一，我们避免了这种冗余（我们没有使用简单的线性 (Linear) 而不是线性无偏差）。

乙 大号仿制品 & P 实际考虑

以下限制和实际考虑适用于一般的检索增强模型。TabR 本身并没有为这个列表添加任何新内容。

第一的，对于给定的应用程序，应该从各个角度（商业逻辑、法律考虑、道德方面等）仔细评估使用真实训练对象进行预测是否合理。

第二，根据应用程序的不同，对于给定的目标对象，人们可能只想从可用数据的子集中检索数据，其中该子集是根据特定于应用程序的过滤器为目标对象动态形成的。在第 3.1 节，这意味着 $\text{我坎德} = \text{我坎德} (+) \subset \text{我火车}$ 。

第三，理想情况下，训练期间的检索应该模拟部署期间的检索，否则，基于检索的模型可能会导致（高度）次优的性能。示例：

- 对于时间序列，在训练期间，必须允许 TabR 仅检索过去数据。此外，也许还应该限制这个“过去”数据，以防止检索太旧的数据和太新的数据。应根据领域专业知识和业务逻辑做出决定。
- 让我们考虑这样一个任务，在所有训练对象中，有一些“相关对象”。例如，当以逐点回归的形式解决排名问题时，可以获得与同一查询但不同文档相对应的查询-文档对作为此类“相关对象”。在某些情况下，在训练期间，对于给定的目标对象，从“相关对象”中检索可能不公平，因为对于在可用数据中没有“相关对象”的新对象，在生产中不可能获得相同的结果。同样，此设计决策应基于领域专业知识和业务逻辑做出。

最后，虽然 TabR 比之前基于检索的表格深度学习模型效率更高，但检索模块 R 与纯参数模型相比，仍然会产生开销，因此 TabR 可能无法按原样扩展到真正大型的数据集。我们展示了一个简单的技巧，可以将 TabR 扩展到更大的数据集第 5.1 节。我们将在 A.4 小节。

碳 乙基准

C.1 电视违约基准

在表 16，我们提供了更多来自以下数据集的信息表 1. 数据集包括：

- 客户流失建模¹
- 加州住房（房地产数据，（凯莉·佩斯和巴里，1997））
- 16H 号房屋²

¹<https://www.kaggle.com/shrutimechlearn/churn-modelling>

²<https://www.openml.org/d/574>

表 16：主要基准数据集的详细信息。“# Num”、“# Bin”和“# Cat”分别表示数值、二进制和分类特征的数量。“批次大小”是用于训练基于 DL 的模型的默认批次大小。

缩写	姓名	# 火车	# 验证	# 测试	#Num	#Bin	#Cat	任务类型	批次大小
中	客户流失模型	6400	1600	2000	10	3	1	宾类	128
加州	加州住房	13209	3303	4128	8	0	0	回归	256
何	16H 号住宅	14581	3646	4557	16	0	0	回归	256
广告	成人	26048	6513	16281	6	1	8	宾类	256
直接投资	钻石	34521	8631	10788	6	0	3	回归	512
加时赛	奥托集团产品	39601	9901	12376	93	0	0	多类别	512
你好	希格斯·斯莫尔	62751	15688	19610	二十八	0	0	宾类	512
基本法	黑色星期五	106764	26692	33365	4	1	4	回归	512
我们	Shifts Weather (子集)	296554	47373	53172	118	1	0	回归	1024
一氧化碳	封面类型	371847	92962	116203	54	四十四	0	多类别	1024
WE (完整版)	Shifts Weather (完整版)	2965542	47373	531720	118	1	0	回归	1024

- 成人（收入估算，（科哈维，1996））
- 钻石³
- 奥托集团产品分类⁴
- 希格斯（模拟物理粒子，（Baldi 等人，2014）；我们使用 OpenML 存储库中提供的包含 98K 个样本的版本（Vanschoren 等人，2014））
- 黑色星期五⁵
- 天气（温度，（Malinin 等人，2021））。由于数据集较大，我们选取 10% 进行实验。
- 天气（完整）（温度，（Malinin 等人，2021））。原文与论文截取部分。
- 覆盖类型（森林特征，（布莱克德和迪恩，2000））
- Microsoft（搜索查询，（秦、刘，2013））。我们遵循逐点方法学习排名，并将该排名问题视为回归问题。

C.2 其他基准 [Grinsztajn 等人（2022）](#)

在本节中，我们描述了如何使用 [Grinsztajn 等人（2022）](#)。

- 我们使用相同的训练-验证-测试分割。
- 当一个数据集有多个分割时（即，当在 [Grinsztajn 等人（2022）](#)），我们首先将它们视为单独的数据集，同时调整和评估算法，如中所述 [附录 D](#)，然后，我们对分割的指标进行平均，以获得数据集的最终数字。例如，如果给定数据集有五个分割，那么我们会对给定算法进行五次调整和评估，在相应分割上使用 15 个随机种子对五个调整后的配置进行评估，报告的指标值是 $5 \times 15 = 75$ 运行。
- 当有多种的版本一数据集（例如，原始回归任务和相同的数据集，但转换为二分类任务或相同的数据集，但删除了分类特征等），我们只保留一个 *原来的数据集*。
- 我们删除了“眼球运动”数据集，因为该数据集存在泄漏。
- 我们使用如下所述的调整和评估协议 [附录 D](#)，在之前的表格深度学习研究中也使用过（[Gorishniy 等人，2021；2022](#)）。至关重要的是，我们对 GBDT 模型的超参数进行了比大多数（如果不是全部）先前工作更广泛的调整，包括预算（20 次随机采样的热身迭代，然后是 180 次树结构 Parzen 估计器算法的迭代）和超参数空间（参见 [附录 D](#)）。

³<https://www.openml.org/d/42225>

⁴<https://www.kaggle.com/c/otto-group-product-classification-challenge/data>

⁵<https://www.openml.org/d/41540>

德 我实施细节

D.1 赫硬件

我们在随源代码一起发布的结果中报告了所使用的硬件。简而言之，绝大多数 GPU 实验是在一块 NVidia A100 GPU 上进行的，其余一小部分 GPU 实验是在一块 Nvidia 2080 Ti GPU 上进行的，还有一小部分运行是在 CPU 上进行的（例如，所有 LightGBM 上的实验）。

D.2 我实施细节小节5.1

在第5.1节，我们使用了 TabR-S 和默认超参数（参见D.8 小节）计算 Δ -上下文，我们收集训练对象的上下文分布*训练周期之间*。也就是说，在我第 1 个训练阶段，我们暂停训练，收集所有训练对象的上下文分布，然后开始下一个阶段（我+1）-第 个训练时期。

Δ -语境。直观地讲，这个启发式指标用一个数字描述了对于给定的输入对象，上下文注意力质量相对于*以前的时代*。也就是说，它是两个项的总和：

1. 的小说注意力质量，即来自当前时期出现的、但在前一个时期未出现的上下文对象的注意力质量
2. 增加注意力质量，即我们取当前上下文对象和前一个上下文对象的交集，并计算它们的总注意力质量的增加量。如果实际减少了，我们将其设置为 0.0。

现在，我们正式定义这个指标。对于给定的输入对象，让一个 $\in \mathbb{R}^{|I_{\text{我}}|}$ 和 $b \in \mathbb{R}^{|I_{\text{我}}|}$ 分别表示前一个时期和当前时期候选集的两个分布。将非零项集表示为一个 $\{ \text{我} : \text{一个我} > 0 \}$ 和 $\{ \text{我} : \text{一个我} > 0 \}$ 。注意 $|I_{\text{我}}|/|I_{\text{我}}| = 96$ 。换句话说，一个和乙是两个时代的背景。然后：

$$\Delta\text{-上下文} = \text{新颖} + \text{增加} \quad (8)$$

$$\text{小说} = \sum_{\text{我} \in \text{乙}} b_{\text{我}} \quad (9)$$

$$\text{增加} = \text{最大限度} \left(\sum_{\text{我} \in \text{乙} \cap \text{一个}} b_{\text{我}} - \sum_{\text{我} \in \text{乙} \cap \text{一个}} \text{一个我}, 0.0 \right) \quad (10)$$

D.3 我实施细节小节5.2

在第5.2节，我们使用了 TabR-S 和默认超参数（参见D.8 小节）。

D.4 我实施细节子节A.1.2

在A.1.2 小节，我们在与组装“步骤 1”和“步骤 2”行时使用的完全相同的模型检查点上执行了分析表 2。

重申一下，这就是表 7 计算如下：

1. 首先，我们获得所有测试对象的候选分布。一个这样的分布恰好包含（米+1）非零条目。
2. 然后，我们对所有个体分布求平均，并得到平均分布。
3. 表 7 报告平均分布的熵。

请注意，在获取候选人的分布时，前米操作被考虑在内。如果没有这个，如果无论输入对象如何，分布总是均匀的，那么平均分布也将是均匀的，并且具有最高的熵，这在故事的背景下会产生误导A.1.2 小节。

最后，回想一下，在步骤 1 和步骤 2 模型中，输入对象被添加到其自己的上下文中。然后，当所有输入对象 100% 只关注自己时，边缘情况将导致最高的

可能的熵，这会对故事产生误导。A.1.2 小节换句话说，对于 A.1.2 小节，我们应该对所有对象都采取类似的“关注自我”行为。为了实现这一点，在上述方法的第一步中，我们将注意力从“自我”重新分配给一个新的虚拟上下文对象，即相同对于所有输入对象。

D.5 我实施细节 A.2

构建表 8，我们使用了 TabR-S 和默认超参数（参见 D.8 小节）。

D.6 E 实验设置

大多数情况下，我们只是遵循 Gorishniy 等人 (2022)，但为了完整性，我们提供了所有细节。请注意，一些先前的工作可能与我们下面描述的通用协议不同，但我们在本节中进一步提供了特定于算法的实现细节。

数据预处理。对于每个数据集，对于所有基于深度学习的解决方案，都使用相同的预处理来进行公平比较。对于数值特征，默认情况下，我们使用 Scikit-learn 包中的分位数归一化 (Pedregosa 等人, 2011)，但极少数情况下，当它被证明是有害的时（对于这样的数据集，我们使用标准规范化或不进行规范化）。对于分类特征，我们使用独热编码。二进制特征（即仅采用两个不同值的特征）被映射到 $\{0, 1\}$ 无需任何进一步的预处理。

训练神经网络。对于基于深度学习的算法，我们最小化分类问题的交叉熵和回归问题的均方误差。我们使用 AdamW 优化器 (洛什奇洛夫和胡特, 2019)。我们不应用学习率计划。我们不使用数据增强。对于每个数据集，我们使用预定义的特定于数据集的批处理大小。我们继续训练，直到有耐心 +1 连续几个时期没有对验证集进行改进；我们设定 耐心 = 16 适用于 DL 模型。

我们如何比较算法。对于给定的数据集，我们首先将“初步最佳”算法定义为具有最佳平均得分的算法。然后，我们定义一组最佳算法（即它们的结果在表格中以粗体显示），如下所示：如果给定算法的平均得分与初步最佳算法的平均得分之差不超过初步最佳算法的标准差，则该算法被纳入最佳算法中。

D.7 E 数值特征的向量嵌入

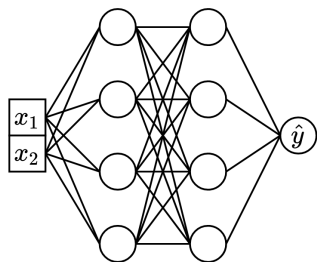


图 8: (复制自 Gorishniy 等人 (2022)) 原始 MLP。该模型以两个数值特征作为输入。

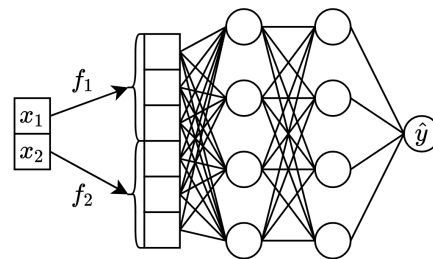


图 9: (复制自 Gorishniy 等人 (2022)) 与图 8，但现在有了数值特征的嵌入。

在这项工作中，我们积极地利用了来自 (Gorishniy 等人, 2022) (看图 8 和图 9)，据报道该技术可以普遍改善 DL 模型。简而言之，对于给定的标量数值特征，嵌入模块是一个可训练的模块，它将这个标量特征映射到一个向量。然后，所有数值特征的嵌入被连接成一个平面向量，并传递给后续层。根据原始论文，当我们对数值特征使用嵌入时，所有数值特征都使用相同的嵌入架构。

在这项工作中，我们使用了左 (线性层和 ReLU 的组合) 以及公共利益清单 (原始论文中的周期嵌入、线性层和 ReLU 嵌入的组合)。此外，我们

介绍PLR（精简版）嵌入，是公共关系嵌入线性层在所有特征之间共享。我们观察到它显著地更加轻量，且没有严重的性能损失。

超参数调整。对于左心室嵌入，我们调整嵌入维度 制服[16, 96]。对于公共关系和PLR（精简版）嵌入，我们调整频率的数量制服[16, 96]（在制服[8, 96]对于 TabR 的数据集Grinsztajn 等人 (2022)），频率初始化尺度对数均匀[0.01, 100.0]以及嵌入维数 制服[16, 64]（在制服[4, 64]对于 TabR 的数据集Grinsztajn 等人 (2022)）。

D.8 TabR

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

数值特征的嵌入。（看D.7 小节）对于 TabR 的非简单配置，在数据集 CH、CA、HO、AD、DI、OT、HI、BL 以及来自的所有数据集上Grinsztajn 等人 (2022)，我们使用了PLR（精简版）嵌入定义在D.7 小节。对于其他数据集，我们使用左心室嵌入。

其他详情。我们观察到初始化西是正确使用模块对于获得良好性能可能很重要。请参阅源代码。

默认 TabR-S。TabR-S 的默认超参数是在项目的某个阶段通过对多个数据集上的调整后的超参数进行平均而获得的。用于平均的特定数据集包括来自表 16加上两个曾经是默认基准的一部分但后来被排除的数据集。因此，总共有 13 个数据集为默认超参数做出了贡献。

正式地，在对默认超参数有贡献的数据集上评估获得的默认 TabR-S 并不是 100% 公平的，就像表4。但是，我们也测试了公平留一法（即对于给定的数据集，对除此数据集之外的所有数据集的调整超参数取平均值），并没有观察到任何有意义的变化，因此我们决定保持简单，并为所有数据集设置一组通用的默认超参数。此外，获得的默认 TabR-S 在以下方面表现出色表 5这也说明获得的默认配置并没有强烈地“过度拟合”来自表 16TabR-S具体默认超参数值如下：

- $d=265$
- 注意力流失率 = 0.38920071545944357
- 辍学率FFN = 0.38852797479169876
- 学习率 = 0.0003121273641315169
- 体重衰减 = 0.0000012260352006404615

超参数。第一线性层的输出大小纤维蛋白原和电视是 $2d$ 我们使用来自的树形结构 Parzen 估计器算法进行调整Akiba 等人 (2019) 库。在调整模型时使用了相同的协议和超参数空间表 2和表 9。

表 17: TabR 的超参数调整空间。这里 (A) = {CH, CA, HO, AD, DI, OT, HI, BL}, (B) = {WE, CO, MI}。对于来自Grinsztajn 等人 (2022), 调整空间与 (A) 相同, 唯一的区别是 d 正在收听 UniformInt[16, 384]。

范围	(数据集) 分布	评论
宽度 d	(A、B) UniformInt[96, 384]	(A、B)
注意力流失率	制服[0.0, 0.6] (A、B) 制服[0.0, 0.6] (A、B)	
辍学率纤维蛋白原	对数均匀[3埃5, 1埃3]	
学习率	(A、B) {0, 对数均匀[1埃6, 1埃3]}	
权重衰减	(A、B) UniformInt[0, 1] (A、B) 统一	
否埃	整数[1, 2]	常量[0]对于 TabR-S
否磷		常量[1]对于 TabR-S
# 调整迭代	(A) 100 (B) 50	

D.9 MLP

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里, 我们总结了一些细节以方便参考。

我们使用了Gorishniy 等人 (2022)。

超参数。我们在整个网络中使用相同的隐藏维度。我们使用来自Akiba 等人 (2019) 图书馆。

表 18: MLP 的超参数调整空间

范围	分配
# 层	统一整数[1, 6]
宽度 (隐藏尺寸)	UniformInt[64, 1024]
辍学率	{0.0, 制服[0.0, 0.5]} 对数
学习率	均匀[3埃5, 1埃3] {0, 对数均
权重衰减	匀[1埃6, 1埃3]}
# 调整迭代	100

D.10 FT-T 变压器

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里, 我们总结了一些细节以方便参考。

我们采用了“rtdl”Python 包 (版本 0.0.13)。

超参数。我们使用rtdl.FTTransformer.make_baseline方法来创建 FT-Transformer, 因此大多数超参数都继承自该方法的签名, 其余的则按照相应表所示进行调整。

表 19: FT-Transformer 的超参数调整空间

范围	分配
# 区块	统一整数[1, 4]
$d_{令牌}$	UniformInt[16, 384]
注意力流失率	制服[0.0, 0.5] 制服[$\frac{2}{3}$, $\frac{8}{3}$]
FFN隐藏维度扩展率 FFN dropout率]制服[0.0, 0.5] {0.0, 制服[0.0, 0.2]}对数均匀[1埃-5, 1埃3] {0, 对数均匀[1埃-6, 1埃4]}
剩余辍学率	
学习率	
权重衰减	
# 调整迭代	100

D.11神经网络

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

特征的预处理方式与 DL 模型相同。唯一的超参数是我们调整的邻居数量统一整数[1, 128]。

D.12 DNNR

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

我们使用了官方实现⁶但为了在超参数变化更大的大型数据集上评估 DNNR，我们重写了部分源代码，使其更高效：尽可能启用 GPU 使用、批量数据处理、多处理。至关重要的是，我们保持底层方法不变。我们在源代码中提供了效率改进的 DNNR。不支持分类问题，因此我们仅在回归问题上评估 DNNR。

超参数。我们对所有数据集上的主要 DNNR 超参数进行了网格搜索，由于 WE 和 MI 上的扩展问题，恢复为默认值（作者建议）。

表 20: 用于 DNNR 的超参数网格。这里 (A) = {CA, HO}; (B) = {DI, BL, WE, MI}。符号: \neg 数据集的特征数量。

范围	(数据集) 参数网格	评论
# 邻居 κ	(A, B) [1, 2, 3, ..., 128]	
学会扩展	(A, B) [无缩放, 训练有素的缩放] (A, B) [8 ·	
# 用于缩放的邻居	$\neg r$, 2, 3, 4, 8, 16, 三十二, 64, 128] 10	8 · \neg 在 WE, MI
# 用于缩放 Cat. 特征编码的时期	[独热算法、留一法]	
# 导数的邻居 κ'	(一个) 林空间[2 · $\neg r$, 18 · $\neg r$, 20] (二) 林空间[2 · $\neg r$, 12 · $\neg r$, 14]	

⁶<https://github.com/younader/dnnr>

D.13 DKL

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

我们使用了 GPyTorch 的 DKL 实现 (加德纳等人, 2018)。由于扩展问题，我们没有在 WE 和 MI 数据集上评估 DKL (例如，仅调整就需要 1 天 17 小时，而 TabR 在中等 DI 数据集上只需 3 小时)。不支持分类问题，因此我们仅在回归问题上评估 DKL。

超参数。与 MLP 一样，我们在整个网络中使用相同的隐藏维度。并使用 Akiba et al. [1] 库中的树结构 Parzen Estimator 算法进行调整。

表 21: DKL 的超参数调整空间

范围	分配
核心	<i>{径向基函数, 山猫}</i>
# 层	统一整数[1, 4]
宽度 (隐藏尺寸)	UniformInt[64, 768]
辍学率	<i>{0.0, 0.1, 0.2, 0.3, 0.4, 0.5}</i> 对数
学习率	均匀[1 埃 5, 1 埃 2] <i>{0.0, 0.001, 0.01, 0.1}</i> 对数均
权重衰减	匀[1 埃 6, 1 埃 3]
# 调整迭代	100

D.14 心钠素

虽然介绍 ANP 的原始论文并没有关注表格数据，但从概念上讲，它与基于检索的表格 DL 的先前工作非常相关，因此我们将其视为基线之一。

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

我们使用了非官方存储库中的 Pytorch 实现⁷并根据官方实现进行了修改Kim 等人 (2019)。具体来说，我们重新实现了解码器 就像在Kim 等人 (2019) 并将二元交叉熵损失改为高斯负对数似然损失潜在模型类，因为它与官方实现相匹配。

由于扩展问题，我们没有在 MI 数据集上评估 ANP。仅在较小的 WE 数据集上进行调整就需要四天以上的时间进行 20 次迭代 (而其他算法则需要 50-100 次)。此外，不支持分类问题，因此我们仅在回归问题上评估 ANP。

我们对 CA 和 HO 进行了 100 次调整迭代，对 DI 进行了 50 次调整迭代，对 BL 和 WE 进行了 20 次调整迭代。

⁷<https://github.com/soobinseo/Attentive-Neural-Process>

表 22: ANP 的超参数调整空间

范围	分配
# 解码层	统一整数[1, 3]
# 交叉注意力层	统一整数[1, 2]
# 自注意力层宽度 (隐藏大小)	统一整数[1, 2] UniformInt[64, 384]
学习率	对数均匀[3埃5, 1埃3] {0, 对
权重衰减	数均匀[1埃6, 1埃4]}

D.15 不扩散核武器条约

我们使用官方的NPT (Kossen 等人, 2021) 执行⁸。我们保持模型和训练代码不变, 仅根据我们的协议调整数据集及其预处理。

我们评估NPT 基座配置模型并遵循NPT 基座架构和优化超参数。我们训练 NPT2000CH、CA、AD、HO 上的纪元, 10000OT、WE、MI 上的时代, 15000DI、BL、HI 上的时期和30000对于所有不适合 A100 80GB GPU 的数据集, 我们使用批量大小4096 (如 NPT 论文中所建议的那样)。我们还将隐藏暗淡降低到三十二在 WE 和 MI 上以避免 OOM 错误。

请注意, NPT 在概念上等同于其他基于 Transformer 的非参数表格 DL 解决方案: (Somepalli 等人, 2021; Schäfl 等人, 2022) 这三种方法都使用基于点积的自注意力模块, 在对象特征之间的自注意力和对象之间的自注意力之间交替 (针对整个训练数据集或其随机子集)。

D.16 圣人

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里, 我们总结了一些细节以方便参考。

我们使用 SAINT 的官方实现⁹并有一个重要修复。回想一下, 在 SAINT 中, 目标对象通过样本间注意与其上下文对象进行交互。在 SAINT 的官方实现中, *上下文对象与目标对象取自同一数据集部分*: 对于训练对象, 上下文对象取自训练集; 对于验证对象, 上下文对象取自验证集; 对于测试对象, 上下文对象取自测试集。这与本文中描述的方法不同, 本文中*上下文对象总是取自训练集*。像 SAINT 的官方实现那样从不同的数据集部分获取上下文对象可能是不受欢迎的, 原因如下:

1. 模型的验证和测试性能可能不是最理想的, 因为它被训练为在上下文对象取自训练集时进行操作, 但是当上下文对象取自其他数据集部分时进行评估。
2. 对于给定的验证/测试对象, *预测取决于其他验证/测试对象*。这与其他基于检索的模型不一致, 可能会导致不一致的比较。此外, 在许多现实场景中, 在部署/测试时, 输入对象应该单独处理, 而 SAINT 的官方实现并非如此。

出于上述原因, 我们对 SAINT 进行了轻微修改, 使得每个样本在训练和评估期间都只关注自身和来自训练集的上下文样本。有关详细信息, 请参阅源代码。

在小数据集 (CH、CA、HO、AD、DI、OT、HI、BL) 上, 我们将注意力头的数量固定在8并使用来自的树结构 Parzen 估计器算法执行超参数调整 Akiba 等人 (2019) 图书馆。

⁸<https://github.com/OATML/non-parametric-transformers>

⁹<https://github.com/somepago/saint>

表 23: SAINT 的超参数调整空间

范围	分配
深度	统一整数[1, 4]
宽度	UniformInt[4, 三十二, 4] 制
前馈乘数注意力机制	服[2/3, 8/3] 制服[0, 0.5] 制服
前馈丢失	[0, 0.5] 对数均匀[3埃5, 1埃
学习率	-3] {0, 对数均匀[1埃6, 1埃4]
权重衰减	}

在较大的数据集（WE，CO，MI）上，我们使用论文中稍微修改过的（为了优化内存消耗）默认配置，并具有以下固定的超参数：

- 深度 =4
- n_heads =8
- 昏暗 =三十二
- ffn_mult =4
- attn_head_dim=四十八
- attn_dropout=0.1
- ff_dropout=0.8
- 学习率=0.0001
- 体重衰减=0.01

D.17 XGB东南欧

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

在本研究中，我们尽最大努力对 GBDT 模型进行尽可能好的调整，以确保比较公平，结论可靠。与以前的研究相比（[Gorishniy 等人，2021；2022](#)），其中 GBDT 已经进行了广泛的调整，我们增加了一倍的调整迭代次数，增加了一倍的树的数量，增加了最大深度，并且将早期停止轮次的数量增加了 4 倍。

以下超参数是固定的，未经调整：

- 助推器= “gbtree”
- n_估计器=4000
- 树方法 = “gpu_hist”
- 提前停止回合 =200

我们使用来自的树形结构 Parzen 估计器算法进行调整[Akiba 等人（2019）](#) 图书馆。

表 24: XGBoost 的超参数调整空间

范围	分配
colsample_bytree	制服[0.5, 1.0]
伽马	{0.0, 对数均匀[0.001, 100.0]}
拉姆达	{0.0, 对数均匀[0.1, 10.0]}
学习率	对数均匀[0.001, 1.0]
最大深度	统一整数[3, 14]
最小儿童体重	对数均匀[0.0001, 100.0]
子样本	统一[0.5, 1.0]
# 调整迭代	200

D.18 大号右胶质母细胞瘤

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

在本研究中，我们尽最大努力对 GBDT 模型进行尽可能好的调整，以确保比较公平，结论可靠。与以前的研究相比（[Gorishniy 等人, 2021](#)；[2022](#)），其中 GBDT 已经进行了广泛的调整，我们增加了一倍的调整迭代次数，增加了一倍的树的数量，增加了最大深度，并且将早期停止轮次的数量增加了 4 倍。

以下超参数是固定的，未经调整：

- n_估计器=4000
- 提前停止回合 =200

我们使用来自的树形结构 Parzen 估计器算法进行调整[Akiba 等人（2019）](#) 图书馆。

表 25: LightGBM 的超参数调整空间

范围	分配
特征分数	制服[0.5, 1.0]
lambda_l2	{0.0, 对数均匀[0.1, 10.0]}
学习率	对数均匀[0.001, 1.0]
叶子数量	统一整数[4, 768]
min_sum_hessian_in_leaf	对数均匀[0.0001, 100.0]
bagging_fraction	统一[0.5, 1.0]
# 调整迭代	200

D.19 C在乙东南欧

源代码中提供了实现、调整超参数、评估超参数、指标、执行时间、硬件和其他详细信息。这里，我们总结了一些细节以方便参考。

在本研究中，我们尽最大努力对 GBDT 模型进行尽可能好的调整，以确保比较公平，结论可靠。与以前的研究相比（[Gorishniy 等人, 2021](#)；[2022](#)），其中 GBDT 已经进行了广泛的调整，我们将调整迭代次数增加了一倍，

树木的数量，增加了最大深度，并将早期停止轮次的数量增加了 4 倍。

以下超参数是固定的，未经调整：

- `n_估计器`=4000
- 提前停止回合 =200
- `od_pval` =0.001

我们使用来自的树形结构 Parzen 估计器算法进行调整Akiba 等人（2019）图书馆。

表 26：CatBoost 的超参数调整空间

范围	分配
装袋温度	制服[0。0, 1。0]
深度	统一整数[3, 14]
<code>l2_leaf_reg</code>	制服[0。1, 10。0]
叶估计迭代学习率	制服[1, 10]
	对数均匀[0。001, 1。0]
# 调整迭代	200

电子工程扩展结果与标准偏差

在本节中，我们提供了正文中报告的主要结果的扩展结果和标准差。默认基准的结果位于表 27。基准测试结果来自Grinsztajn 等人（2022）位于[0477]表 28。

表 27：默认基准的扩展结果。结果按数据集分组，并跨越以下多页。符号：↓ 对应于 RMSE，↑ 对应于准确性。

中↑		
方法	单一模型	合奏
调整超参数		
kNN	0. 837 ±0. 000	—
地下水自然保护区	—	—
丹麦航空	—	—
心钠素	—	—
《不扩散核武器条约》	0. 858 ±0. 003 0	—
圣	。 860 ±0. 003 0	—
多层感知处理器	。 854 ±0. 003	—
多层线性规划	0. 860 ±0. 002 0. 860 ±0. 001 0	
TabR-S	。 860 ±0. 002 0. 862 ±0. 002 0	
泰伯	。 862 ±0. 002 0. 865 ±0. 001 0	
CatBoost	。 858 ±0. 002 0. 859 ±0. 001 0	
XGBoost	。 861 ±0. 002 0. 861 ±0. 001	
轻量级GBM0.	860 ±0. 001 0. 860 ±0. 000	
默认超参数		
CatBoost	0. 860 ±0. 002 0. 861 ±0. 001 0	
XGBoost	。 855 ±0. 000 0. 856 ±0. 000	
轻量级GBM0.	856 ±0. 000 0. 856 ±0. 000	
TabR-S	0. 859 ±0. 003 0. 864 ±0. 001	
调整超参数（表 2）		
步骤-0	0. 855 ±0. 003 0. 857 ±0. 002 0	
步骤1	。 855 ±0. 003 0. 858 ±0. 002 0	
步骤2	。 860 ±0. 002 0. 862 ±0. 003 0	
步骤3	。 859 ±0. 002 0. 862 ±0. 002	

何↓		
方法	单一模型	合奏
调整超参数		
kNN	3. 744 ±0. 000 3	—
地下水自然保护区	。 210 ±0. 000 3	—
丹麦航空	。 423 ±0. 393 3	—
心钠素	。 162 ±0. 028 3	—
《不扩散核武器条约》	。 175 ±0. 032 3	—
圣	。 242 ±0. 059 3	—
多层感知处理器	。 112 ±0. 036	—
多层线性规划	3. 056 ±0. 021 2. 993 ±0. 019 3	
TabR-S	。 067 ±0. 040 2. 996 ±0. 027 3	
泰伯	。 105 ±0. 041 3. 025 ±0. 010 3	
CatBoost	。 117 ±0. 013 3. 106 ±0. 002 3	
XGBoost	。 177 ±0. 010 3. 164 ±0. 007	
轻量级GBM3.	177 ±0. 009 3. 167 ±0. 005	
默认超参数		
CatBoost	3. 122 ±0. 011 3. 108 ±0. 002 3	
XGBoost	。 368 ±0. 000 3. 368 ±0. 000	
轻量级GBM3.	222 ±0. 000 3. 222 ±0. 000	
TabR-S	3. 093 ±0. 060 2. 971 ±0. 017	
调整超参数（表 2）		
步骤-0	3. 234 ±0. 053 3. 144 ±0. 034 3	
步骤1	。 205 ±0. 056 3. 104 ±0. 043 3	
步骤2	。 153 ±0. 031 3. 117 ±0. 012 3	
步骤3	。 158 ±0. 017 3. 117 ±0. 006	

加州↓		
方法	单一模型集成	
调整超参数		
kNN	0. 588 ±0. 000 0	—
地下水自然保护区	。 430 ±0. 000 0	—
丹麦航空	。 521 ±0. 055 0	—
心钠素	。 472 ±0. 007 0	—
《不扩散核武器条约》	。 474 ±0. 003 0	—
圣	。 468 ±0. 005 0	—
多层感知处理器	。 499 ±0. 004	—
多层线性规划	0. 476 ±0. 004 0. 470 ±0. 001 0	
TabR-S	。 403 ±0. 002 0. 396 ±0. 001 0	
泰伯	。 400 ±0. 003 0. 391 ±0. 002 0	
CatBoost	。 429 ±0. 001 0. 426 ±0. 000 0	
XGBoost	。 433 ±0. 002 0. 432 ±0. 001	
轻量级GBM0.	435 ±0. 002 0. 434 ±0. 001	
默认超参数		
CatBoost	0. 433 ±0. 001 0. 432 ±0. 001 0	
XGBoost	。 471 ±0. 000 0. 471 ±0. 000	
轻量级GBM0.	449 ±0. 000 0. 449 ±0. 000	
TabR-S	0. 406 ±0. 003 0. 398 ±0. 001	
调整超参数（表 2）		
步骤-0	0. 484 ±0. 006 0. 470 ±0. 005 0	
步骤1	。 489 ±0. 007 0. 474 ±0. 005 0	
步骤2	。 418 ±0. 002 0. 411 ±0. 000 0	
步骤3	。 408 ±0. 003 0. 399 ±0. 002	

广告↑		
方法	单一模型集成	
调整超参数		
kNN	0. 834 ±0. 000	—
地下水自然保护区	—	—
丹麦航空	—	—
心钠素	—	—
《不扩散核武器条约》	0. 853 ±0. 010 0	—
圣	。 860 ±0. 002 0	—
多层感知处理器	。 853 ±0. 001	—
多层线性规划	0. 870 ±0. 002 0. 873 ±0. 001 0	
TabR-S	。 865 ±0. 002 0. 868 ±0. 002 0	
泰伯	。 870 ±0. 001 0. 872 ±0. 001 0	
CatBoost	。 871 ±0. 001 0. 872 ±0. 001 0	
XGBoost	。 872 ±0. 001 0. 872 ±0. 000	
轻量级GBM0.	871 ±0. 001 0. 872 ±0. 000	
默认超参数		
CatBoost	0. 873 ±0. 001 0. 874 ±0. 001 0	
XGBoost	。 871 ±0. 000 0. 871 ±0. 000	
轻量级GBM0.	869 ±0. 000 0. 869 ±0. 000	
TabR-S	0. 858 ±0. 001 0. 859 ±0. 000	
调整超参数（表 2）		
步骤-0	0. 857 ±0. 002 0. 858 ±0. 000 0	
步骤1	。 857 ±0. 002 0. 860 ±0. 000 0	
步骤2	。 858 ±0. 002 0. 862 ±0. 001 0	
步骤3	。 863 ±0. 002 0. 866 ±0. 001	

直接投资 ↓		
方法	单一模型	合奏
调整超参数		
kNN	0. 256 ±0. 000 0	—
地下水自然保护区	。 145 ±0. 000 0	—
丹麦航空	。 147 ±0. 005 0	—
心钠素	。 140 ±0. 001 0	—
《不扩散核武器条约》	。 138 ±0. 001 0	—
圣	。 137 ±0. 002 0	—
多层感知处理器	。 140 ±0. 001	—
多层线性规划	0. 134 ±0. 001 0. 133 ±0. 000 0	
TabR-S	。 133 ±0. 001 0. 131 ±0. 000 0	
泰伯	。 133 ±0. 001 0. 131 ±0. 000 0	
CatBoost	。 134 ±0. 001 0. 133 ±0. 000 0	
XGBoost	。 137 ±0. 000 0. 136 ±0. 000	
轻量级GBM0.	136 ±0. 000 0. 136 ±0. 000	
默认超参数		
CatBoost	0. 133 ±0. 000 0. 132 ±0. 000 0	
XGBoost	。 143 ±0. 000 0. 143 ±0. 000	
轻量级GBM0.	137 ±0. 000 0. 137 ±0. 000	
TabR-S	0. 133 ±0. 001 0. 131 ±0. 000	
调整超参数（表 2）		
步骤-0	0. 142 ±0. 001 0. 139 ±0. 001 0	
步骤1	。 142 ±0. 002 0. 138 ±0. 000 0	
步骤2	。 140 ±0. 001 0. 139 ±0. 001 0	
步骤3	。 135 ±0. 001 0. 133 ±0. 001	

你好 ↑		
方法	单一模型	合奏
调整超参数		
kNN	0. 665 ±0. 000	—
地下水自然保护区	—	—
丹麦航空	—	—
心钠素	—	—
《不扩散核武器条约》	0. 721 ±0. 003 0	—
圣	。 724 ±0. 002 0	—
多层感知处理器	。 719 ±0. 002	—
多层线性规划	0. 729 ±0. 002 0. 735 ±0. 000 0	
TabR-S	。 722 ±0. 001 0. 726 ±0. 001 0	
泰伯	。 729 ±0. 001 0. 733 ±0. 001 0	
CatBoost	。 726 ±0. 001 0. 727 ±0. 001 0	
XGBoost	。 725 ±0. 002 0. 726 ±0. 001	
轻量级GBM0.	726 ±0. 001 0. 726 ±0. 001	
默认超参数		
CatBoost	0. 725 ±0. 001 0. 726 ±0. 001 0	
XGBoost	。 716 ±0. 000 0. 716 ±0. 000	
轻量级GBM0.	720 ±0. 000 0. 720 ±0. 000	
TabR-S	0. 719 ±0. 002 0. 724 ±0. 000	
调整超参数（表 2）		
步骤-0	0. 719 ±0. 002 0. 727 ±0. 000 0	
步骤1	。 719 ±0. 002 0. 724 ±0. 001 0	
步骤2	。 720 ±0. 002 0. 723 ±0. 001 0	
步骤3	。 722 ±0. 002 0. 724 ±0. 000	

加时赛 ↑		
方法	单一模型集成	
调整超参数		
kNN	0. 774 ±0. 000	—
地下水自然保护区	—	—
丹麦航空	—	—
心钠素	—	—
《不扩散核武器条约》	0. 815 ±0. 002 0	—
圣	。 812 ±0. 002 0	—
多层感知处理器	。 816 ±0. 003	—
多层线性规划	0. 819 ±0. 002 0. 822 ±0. 002 0	
TabR-S	。 818 ±0. 002 0. 824 ±0. 001 0	
泰伯	。 825 ±0. 002 0. 831 ±0. 001 0	
CatBoost	。 825 ±0. 001 0. 827 ±0. 000 0	
XGBoost	。 830 ±0. 001 0. 832 ±0. 001	
轻量级GBM0.	830 ±0. 001 0. 832 ±0. 001	
默认超参数		
CatBoost	0. 820 ±0. 001 0. 822 ±0. 001 0	
XGBoost	。 817 ±0. 000 0. 817 ±0. 000	
轻量级GBM0.	826 ±0. 000 0. 826 ±0. 000	
TabR-S	0. 816 ±0. 002 0. 824 ±0. 000	
调整超参数（表 2）		
步骤-0	0. 814 ±0. 002 0. 823 ±0. 002 0	
步骤1	。 814 ±0. 002 0. 824 ±0. 001 0	
步骤2	。 813 ±0. 002 0. 818 ±0. 001 0	
步骤3	。 810 ±0. 002 0. 814 ±0. 001	

基本法 ↓		
方法	单一模型集成	
调整超参数		
kNN	0. 712 ±0. 000 0	—
地下水自然保护区	。 704 ±0. 000 0	—
丹麦航空	。 699 ±0. 001 0	—
心钠素	。 705 ±0. 005 0	—
《不扩散核武器条约》	。 692 ±0. 001 0	—
圣	。 693 ±0. 001 0	—
多层感知处理器	。 697 ±0. 001	—
多层线性规划	0. 687 ±0. 000 0. 684 ±0. 000 0	
TabR-S	。 690 ±0. 000 0. 688 ±0. 000 0	
泰伯	。 676 ±0. 001 0. 674 ±0. 001 0	
CatBoost	。 682 ±0. 000 0. 681 ±0. 000 0	
XGBoost	。 681 ±0. 000 0. 680 ±0. 000	
轻量级GBM0.	680 ±0. 000 0. 679 ±0. 000	
默认超参数		
CatBoost	0. 685 ±0. 000 0. 684 ±0. 000 0	
XGBoost	。 683 ±0. 000 0. 683 ±0. 000	
轻量级GBM0.	681 ±0. 000 0. 681 ±0. 000	
TabR-S	0. 691 ±0. 000 0. 688 ±0. 000	
调整超参数（表 2）		
步骤-0	0. 699 ±0. 001 0. 694 ±0. 001 0	
步骤1	。 698 ±0. 001 0. 693 ±0. 001 0	
步骤2	。 692 ±0. 001 0. 690 ±0. 000 0	
步骤3	。 692 ±0. 001 0. 688 ±0. 000	

我们↓		
方法	单一模型	合奏
调整超参数		
kNN	2. 296 ±0. 000 1	—
地下水自然保护区	。 913 ±0. 000	—
丹麦航空	—	—
心钠素	1. 902 ±0. 009 1	—
《不扩散核武器条约》	。 947 ±0. 006 1	—
圣	。 933 ±0. 028 1	—
多层感知处理器	。 905 ±0. 005	—
多层线性规划	1. 860 ±0. 002 1. 833 ±0. 002 1	—
TabR-S	。 747 ±0. 002 1. 718 ±0. 001 1	—
泰伯	。 690 ±0. 003 1. 661 ±0. 002 1	—
CatBoost	。 807 ±0. 002 1. 773 ±0. 001 1	—
XGBoost	。 784 ±0. 001 1. 769 ±0. 001	—
轻量级GBM	1. 771 ±0. 001 1. 761 ±0. 001	—
默认超参数		
CatBoost	1. 895 ±0. 001 1. 886 ±0. 000 1	—
XGBoost	。 920 ±0. 000 1. 920 ±0. 000	—
轻量级GBM	1. 845 ±0. 003 1. 817 ±0. 001	—
TabR-S	1. 755 ±0. 002 1. 721 ±0. 002	—
调整超参数 (表 2)		
步骤-0	1. 903 ±0. 004 1. 835 ±0. 004 1	—
步骤1	。 906 ±0. 003 1. 845 ±0. 001 1	—
步骤2	。 804 ±0. 003 1. 754 ±0. 001 1	—
步骤3	。 814 ±0. 003 1. 765 ±0. 001	—

MI ↓	
方法	单一模型集成
调整超参数	
kNN	0. 764 ±0. 000 0 —
地下水自然保护区	。 765 ±0. 000 —
丹麦航空	— —
心钠素	— —
《不扩散核武器条约》	0. 753 ±0. 001 0 —
圣	。 763 ±0. 007 0 —
多层感知处理器	。 748 ±0. 000 —
多层线性规划	0. 744 ±0. 000 0. 743 ±0. 000 0
TabR-S	。 750 ±0. 001 0. 749 ±0. 000 0
泰伯	。 750 ±0. 001 0. 748 ±0. 000 0
CatBoost	。 741 ±0. 000 0. 741 ±0. 000 0
XGBoost	。 741 ±0. 000 0. 741 ±0. 000
轻量级GBM	0. 742 ±0. 000 0. 741 ±0. 000
默认超参数	
CatBoost	0. 745 ±0. 000 0. 744 ±0. 000 0
XGBoost	。 750 ±0. 000 0. 750 ±0. 000
轻量级GBM	0. 747 ±0. 000 0. 744 ±0. 000
TabR-S	0. 757 ±0. 001 0. 752 ±0. 001

一氧化碳↑		
方法	单一模型集成	
调整超参数		
kNN	0. 927 ±0. 000	—
地下水自然保护区	—	—
丹麦航空	—	—
心钠素	—	—
《不扩散核武器条约》	0. 966 ±0. 001 0	—
圣	。 964 ±0. 010 0	—
多层感知处理器	。 963 ±0. 001	—
多层线性规划	0. 970 ±0. 001 0. 974 ±0. 000 0	—
TabR-S	。 973 ±0. 000 0. 974 ±0. 000 0	—
泰伯	。 976 ±0. 000 0. 977 ±0. 000 0	—
CatBoost	。 968 ±0. 000 0. 969 ±0. 000 0	—
XGBoost	。 971 ±0. 000 0. 971 ±0. 000	—
轻量级GBM	0. 971 ±0. 000 0. 971 ±0. 000	—
默认超参数		
CatBoost	0. 923 ±0. 000 0. 924 ±0. 000 0	—
XGBoost	。 966 ±0. 000 0. 966 ±0. 000	—
轻量级GBM	0. 884 ±0. 016 0. 899 ±0. 005	—
TabR-S	0. 973 ±0. 001 0. 974 ±0. 000	—
调整超参数 (表 2)		
步骤-0	0. 957 ±0. 002 0. 965 ±0. 001 0	—
步骤1	。 960 ±0. 002 0. 967 ±0. 001 0	—
步骤2	。 972 ±0. 000 0. 973 ±0. 000 0	—
步骤3	。 975 ±0. 001 0. 976 ±0. 000	—

表 28: 扩展结果Grinsztajn 等人 (2022) 基准。结果按数据集分组，并跨越以下多页。符号：↓ 对应于 RMSE，↑ 对应于准确性。

副翼↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	1。624 ±0。035	1。620 ±0。037
多层线性规划	。591 ±0。021	1。582 ±0。019
TabR-S	。620 ±0。030	1。595 ±0。022
泰伯	。615 ±0。035	1。585 ±0。042
CatBoost	。533 ±0。034	1。527 ±0。037
XGBoost	。571 ±0。041	1。565 ±0。040
轻量级GBM	1。581 ±0。038	1。577 ±0。040
默认超参数		
TabR-S	1。615 ±0。029	1。599 ±0。029
CatBoost	。542 ±0。041	1。538 ±0。043
XGBoost	。644 ±0。046	1。644 ±0。048
轻量级GBM	1。594 ±0。051	1。594 ±0。053

巴西房屋↓	
方法	单一模型集成
调整超参数	
多层感知处理器	0。049 ±0。018
多层线性规划	。043 ±0。019
TabR-S	。049 ±0。015
泰伯	。045 ±0。016
CatBoost	。047 ±0。031
XGBoost	。054 ±0。027
轻量级GBM	0。060 ±0。025
默认超参数	
TabR-S	0。052 ±0。016
CatBoost	。043 ±0。027
XGBoost	。052 ±0。025
轻量级GBM	0。071 ±0。021

KDDCup09 追加销售↑	
方法	单一模型集成
调整超参数	
多层感知处理器	0。776 ±0。011
多层线性规划	。797 ±0。009
TabR-S	。784 ±0。014
泰伯	。791 ±0。012
CatBoost	。799 ±0。012
XGBoost	。793 ±0。011
轻量级GBM	0。793 ±0。012
默认超参数	
TabR-S	0。772 ±0。013
CatBoost	。804 ±0。008
XGBoost	。794 ±0。008
轻量级GBM	0。789 ±0。007

共享单车需求↓	
方法	单一模型集成
调整超参数	
多层感知处理器	四十五。702 ±0。756
多层线性规划	42。615 ±0。415
TabR-S	637 ±0。681
泰伯	±0。939
CatBoost	。232
XGBoost	126
轻量级GBM	四十二。503 ±0。190
默认超参数	
TabR-S	43。486 ±0。573
CatBoost	42。848 ±0。256
XGBoost	45。100 ±0。381
轻量级GBM	43。089 ±0。103

希格斯↑	
方法	单一模型集成
调整超参数	
多层感知处理器	0。723 ±0。002
多层线性规划	。728 ±0。001
TabR-S	。725 ±0。001
泰伯	。730 ±0。001
CatBoost	。729 ±0。000
XGBoost	。729 ±0。001
轻量级GBM	0。727 ±0。001
默认超参数	
TabR-S	0。722 ±0。001
CatBoost	。727 ±0。001
XGBoost	。718 ±0。000
轻量级GBM	0。721 ±0。000

魔幻望远镜↑	
方法	单一模型集成
调整超参数	
多层感知处理器	0。853 ±0。006
多层线性规划	。860 ±0。007
TabR-S	。868 ±0。006
泰伯	。864 ±0。005
CatBoost	。859 ±0。007
XGBoost	。855 ±0。009
轻量级GBM	0。855 ±0。008
默认超参数	
TabR-S	0。868 ±0。006
CatBoost	。860 ±0。007
XGBoost	。856 ±0。011
轻量级GBM	0。859 ±0。009

梅赛德斯奔驰绿色制造 ↓ 方法 单一模型集成

调整超参数		
多层感知处理器	8. 383 ±0. 854 8	8. 336 ±0. 888 8
多层线性规划	。 383 ±0. 854 8	。 336 ±0. 888 8
TabR-S	。 351 ±0. 815 8	。 269 ±0. 840 8
泰伯	。 319 ±0. 819 8	。 244 ±0. 844 8
CatBoost	。 163 ±0. 819 8	。 155 ±0. 844 8
XGBoost	。 218 ±0. 817	。 209 ±0. 846 8
轻量级GBM8.	208 ±0. 823	。 162 ±0. 857
默认超参数		
TabR-S	8. 290 ±0. 838 8	8. 223 ±0. 865 8
CatBoost	。 167 ±0. 825 8	。 164 ±0. 848 8
XGBoost	。 371 ±0. 787	。 371 ±0. 810 8
轻量级GBM8.	280 ±0. 845	。 280 ±0. 869

迷你布诺 ↑

方法	单一模型	合奏
调整超参数		
多层感知处理器	0. 947 ±0. 001 0.	948 ±0. 001 0
多层线性规划	。 947 ±0. 001 0.	949 ±0. 000 0
TabR-S	。 949 ±0. 001 0.	950 ±0. 000 0
泰伯	。 948 ±0. 001 0.	949 ±0. 000 0
CatBoost	。 945 ±0. 001 0.	946 ±0. 001 0
XGBoost	。 944 ±0. 001 0.	945 ±0. 000
轻量级GBM0.	942 ±0. 001 0.	943 ±0. 000
默认超参数		
TabR-S	0. 947 ±0. 001 0.	950 ±0. 001 0
CatBoost	。 945 ±0. 001 0.	945 ±0. 000 0
XGBoost	。 942 ±0. 000 0.	942 ±0. 000
轻量级GBM0.	944 ±0. 000 0.	944 ±0. 000

SGEMM GPU 内核性能 ↓

方法	单一模型	集成
调整超参数		
多层感知处理器	0. 016 ±0. 000 0.	016 ±0. 000 0
多层线性规划	。 015 ±0. 000 0.	015 ±0. 000 0
TabR-S	。 017 ±0. 001 0.	016 ±0. 000 0
泰伯	。 015 ±0. 000 0.	015 ±0. 000 0
CatBoost	。 017 ±0. 000 0.	017 ±0. 000 0
XGBoost	。 017 ±0. 000 0.	017 ±0. 000
轻量级GBM0.	017 ±0. 000 0.	017 ±0. 000
默认超参数		
TabR-S	0. 017 ±0. 001 0.	016 ±0. 000 0
CatBoost	。 017 ±0. 000 0.	017 ±0. 000 0
XGBoost	。 017 ±0. 000 0.	017 ±0. 000
轻量级GBM0.	017 ±0. 000 0.	017 ±0. 000

迈阿密住房2016 ↓

方法	单一模型集成	
调整超参数		
多层感知处理器	0. 161 ±0. 003	0. 157 ±0. 003
多层线性规划	。 150 ±0. 002	。 147 ±0. 002
TabR-S	。 142 ±0. 002	。 139 ±0. 002
泰伯	。 139 ±0. 002	。 136 ±0. 002
CatBoost	。 142 ±0. 002	。 141 ±0. 003
XGBoost	。 144 ±0. 003	。 143 ±0. 003
轻量级GBM0.	146 ±0. 002	145 ±0. 003
默认超参数		
TabR-S	0. 141 ±0. 002	0. 139 ±0. 002
CatBoost	。 142 ±0. 003	。 141 ±0. 003
XGBoost	。 160 ±0. 003	。 160 ±0. 003
轻量级GBM0.	152 ±0. 004	152 ±0. 004

网上新闻人气 ↓

方法	单一模型集成		
调整超参数			
多层感知处理器	0. 862 ±0. 001	0. 860 ±0. 000	0. 862 ±0. 000
多层线性规划	。 862 ±0. 001	0. 860 ±0. 000	0. 862 ±0. 000
TabR-S	。 868 ±0. 001	0. 863 ±0. 001	0. 862 ±0. 000
泰伯	。 862 ±0. 001	0. 859 ±0. 000	0. 862 ±0. 000
CatBoost	。 853 ±0. 000	0. 853 ±0. 000	0. 853 ±0. 000
XGBoost	。 854 ±0. 000	0. 854 ±0. 000	0. 854 ±0. 000
轻量级GBM	0. 855 ±0. 000	0. 854 ±0. 000	0. 855 ±0. 000
默认超参数			
TabR-S	0. 870 ±0. 001	0. 864 ±0. 000	0. 870 ±0. 000
CatBoost	。 855 ±0. 000	0. 854 ±0. 000	0. 855 ±0. 000
XGBoost	。 874 ±0. 000	0. 874 ±0. 000	0. 874 ±0. 000
轻量级GBM	0. 862 ±0. 000	0. 862 ±0. 000	0. 862 ±0. 000

肛门猫数据至尊 ↓

方法	单一模型集成		
调整超参数			
多层感知处理器	0. 078 \pm 0. 009	0. 077 \pm 0. 010	0
多层线性规划	。 079 \pm 0. 008	0. 077 \pm 0. 008	0
TabR-S	。 080 \pm 0. 007	0. 076 \pm 0. 005	0
泰伯	。 081 \pm 0. 009	0. 075 \pm 0. 005	0
CatBoost	。 078 \pm 0. 007	0. 073 \pm 0. 002	0
XGBoost	。 080 \pm 0. 013	0. 077 \pm 0. 011	
轻量级GBM0.	078 \pm 0. 012	0. 077 \pm 0. 011	
默认超参数			
TabR-S	0. 077 \pm 0. 007	0. 074 \pm 0. 007	0
CatBoost	。 071 \pm 0. 004	0. 071 \pm 0. 004	0
XGBoost	。 076 \pm 0. 006	0. 076 \pm 0. 006	
轻量级GBM0.	073 \pm 0. 006	0. 073 \pm 0. 006	

银行营销 ↑	
方法	单一模型集成
调整超参数	
多层感知处理器	0.786 ±0.006 0.790 ±0.004 0
多层线性规划	。795 ±0.005 0.798 ±0.004 0
TabR-S	。800 ±0.005 0.802 ±0.004 0
泰伯	。802 ±0.009 0.804 ±0.010 0
CatBoost	。803 ±0.007 0.806 ±0.008 0
XGBoost	。801 ±0.008 0.803 ±0.008
轻量级GBM0.	801 ±0.008 0.801 ±0.007
默认超参数	
TabR-S	0.800 ±0.006 0.801 ±0.005 0
CatBoost	。803 ±0.009 0.803 ±0.009 0
XGBoost	。800 ±0.009 0.800 ±0.009
轻量级GBM0.	803 ±0.004 0.803 ±0.004

加利福尼亚州 ↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.149 ±0.002 0.146 ±0.001 0	
多层线性规划	。138 ±0.001 0.135 ±0.000 0	
TabR-S	。124 ±0.001 0.121 ±0.000 0	
泰伯	。122 ±0.001 0.120 ±0.000 0	
CatBoost	。129 ±0.000 0.128 ±0.000 0	
XGBoost	。131 ±0.001 0.130 ±0.000	
轻量级GBM0.	131 ±0.001 0.130 ±0.000	
默认超参数		
TabR-S	0.124 ±0.001 0.122 ±0.000 0	
CatBoost	。129 ±0.000 0.129 ±0.000 0	
XGBoost	。141 ±0.000 0.141 ±0.000	
轻量级GBM0.	135 ±0.000 0.135 ±0.000	

封面类型 ↑		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.929 ±0.001 0.934 ±0.001 0	
多层线性规划	。944 ±0.002 0.950 ±0.001 0	
TabR-S	。953 ±0.000 0.954 ±0.000 0	
泰伯	。957 ±0.000 0.958 ±0.000 0	
CatBoost	。938 ±0.000 0.939 ±0.000 0	
XGBoost	。940 ±0.000 0.940 ±0.000	
轻量级GBM0.	939 ±0.000 0.939 ±0.000	
默认超参数		
TabR-S	0.952 ±0.000 0.953 ±0.000 0	
CatBoost	。912 ±0.000 0.913 ±0.000 0	
XGBoost	。927 ±0.000 0.927 ±0.000	
轻量级GBM0.	936 ±0.000 0.936 ±0.000	

黑色星期五 ↓	
方法	单一模型集成
调整超参数	
多层感知处理器	0.369 ±0.000 0.367 ±0.000 0
多层线性规划	。363 ±0.000 0.363 ±0.000 0
TabR-S	。364 ±0.000 0.363 ±0.000 0
泰伯	。362 ±0.002 0.359 ±0.001 0
CatBoost	。361 ±0.000 0.360 ±0.000 0
XGBoost	。360 ±0.000 0.360 ±0.000
轻量级GBM0.	360 ±0.000 0.360 ±0.000
默认超参数	
TabR-S	0.364 ±0.000 0.363 ±0.000 0
CatBoost	。361 ±0.000 0.361 ±0.000 0
XGBoost	。362 ±0.000 0.362 ±0.000
轻量级GBM0.	361 ±0.000 0.361 ±0.000

罗盘 ↑		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.768 ±0.005 0.776 ±0.006 0	
多层线性规划	。783 ±0.007 0.796 ±0.006 0	
TabR-S	。863 ±0.003 0.870 ±0.003 0	
泰伯	。871 ±0.003 0.879 ±0.001 0	
CatBoost	。771 ±0.004 0.775 ±0.003 0	
XGBoost	。819 ±0.005 0.822 ±0.003	
轻量级GBM0.	771 ±0.003 0.773 ±0.003	
默认超参数		
TabR-S	0.865 ±0.004 0.870 ±0.001 0	
CatBoost	。758 ±0.002 0.760 ±0.001 0	
XGBoost	。751 ±0.000 0.751 ±0.000	
轻量级GBM0.	762 ±0.004 0.762 ±0.004	

CPU 行为 ↓	
方法	单一模型集成
调整超参数	
多层感知处理器	2.712 ±0.207 2.544 ±0.052 2
多层线性规划	。270 ±0.048 2.214 ±0.059 2
TabR-S	。298 ±0.053 2.223 ±0.050 2
泰伯	。128 ±0.078 2.063 ±0.050 2
CatBoost	。124 ±0.049 2.109 ±0.050 2
XGBoost	。524 ±0.353 2.472 ±0.379
轻量级GBM2.	222 ±0.089 2.207 ±0.092
默认超参数	
TabR-S	2.285 ±0.045 2.214 ±0.032 2
CatBoost	。185 ±0.088 2.162 ±0.091 2
XGBoost	。910 ±0.463 2.910 ±0.486
轻量级GBM2.	274 ±0.128 2.274 ±0.135

信用↑		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.772±0.004	0.774±0.003
多层线性规划	。774±0.004	。775±0.006
TabR-S	。773±0.004	。774±0.004
泰伯	。772±0.004	。775±0.003
CatBoost	。773±0.003	。775±0.004
XGBoost	。770±0.003	。771±0.003
轻量级GBM0.	769±0.003	773±0.003
默认超参数		
TabR-S	0.772±0.005	0.774±0.005
CatBoost	。771±0.005	。773±0.002
XGBoost	。772±0.002	。772±0.002
轻量级GBM0.	771±0.003	771±0.003

电↑		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.832±0.004	0.841±0.002
多层线性规划	。841±0.004	。849±0.000
TabR-S	。924±0.003	。929±0.001
泰伯	。937±0.002	。942±0.000
CatBoost	。880±0.002	。882±0.001
XGBoost	。890±0.001	。891±0.001
轻量级GBM0.	887±0.001	887±0.001
默认超参数		
TabR-S	0.887±0.004	0.893±0.002
CatBoost	。875±0.001	。877±0.000
XGBoost	。882±0.000	。882±0.000
轻量级GBM0.	890±0.000	890±0.000

国际足联↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.803±0.013	0.801±0.015
多层线性规划	。794±0.011	。792±0.012
TabR-S	。790±0.012	。786±0.012
泰伯	。791±0.014	。787±0.016
CatBoost	。783±0.012	。782±0.011
XGBoost	。780±0.011	。780±0.011
轻量级GBM0.	781±0.012	779±0.012
默认超参数		
TabR-S	0.790±0.013	0.786±0.012
CatBoost	。782±0.012	。781±0.013
XGBoost	。790±0.012	。790±0.013
轻量级GBM0.	780±0.011	780±0.011

钻石↓	
方法	单一模型集成
调整超参数	
多层感知处理器	0.091±0.002
多层线性规划	。087±0.001
TabR-S	。083±0.001
泰伯	。083±0.001
CatBoost	。084±0.000
XGBoost	。085±0.000
轻量级GBM0.	085±0.000
默认超参数	
TabR-S	0.084±0.001
CatBoost	。084±0.000
XGBoost	。088±0.000
轻量级GBM0.	086±0.000

电梯↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.005±0.000	0.005±0.000
多层线性规划	0.002±0.000	0.002±0.000
TabR-S	0.005±0.000	0.005±0.000
泰伯	0.002±0.000	0.002±0.000
CatBoost	0.002±0.000	0.002±0.000
XGBoost	0.002±0.000	0.002±0.000
轻量级GBM	0.002±0.000	0.002±0.000
默认超参数		
TabR-S	0.005±0.000	0.005±0.000
CatBoost	0.002±0.000	0.002±0.000
XGBoost	0.002±0.000	0.002±0.000
轻量级GBM	0.002±0.000	0.002±0.000

16H 号房屋↓	
方法	单一模型集成
调整超参数	
多层感知处理器	0.598±0.012
多层线性规划	。594±0.003
TabR-S	。608±0.016
泰伯	。629±0.024
CatBoost	。599±0.005
XGBoost	。591±0.007
轻量级GBM0.	575±0.002
默认超参数	
TabR-S	0.603±0.015
CatBoost	。591±0.002
XGBoost	。589±0.000
轻量级GBM0.	593±0.000

房屋销售↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.181±0.001	0.178±0.000
多层线性规划	0.169±0.001	0.168±0.000
TabR-S	0.169±0.001	0.166±0.000
泰伯	0.164±0.001	0.161±0.000
CatBoost	0.167±0.000	0.167±0.000
XGBoost	0.169±0.000	0.169±0.000
轻量级GBM0.	0.169±0.000	0.169±0.000
默认超参数		
TabR-S	0.169±0.001	0.167±0.000
CatBoost	0.167±0.000	0.167±0.000
XGBoost	0.179±0.000	0.179±0.000
轻量级GBM0.	0.173±0.000	0.173±0.000

异位体↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	2.223±0.189	2.037±0.106
多层线性规划	2.224±0.156	2.030±0.103
TabR-S	2.976±0.174	2.763±0.152
泰伯	2.992±0.181	2.748±0.143
CatBoost	2.867±0.014	2.848±0.002
XGBoost	2.757±0.047	2.729±0.037
轻量级GBM2.	2.701±0.030	2.690±0.029
默认超参数		
TabR-S	1.995±0.156	1.754±0.106
CatBoost	2.895±0.020	2.863±0.013
XGBoost	2.368±0.010	2.368±0.011
轻量级GBM2.	2.953±0.056	2.953±0.058

kdd ipums la 97-small↑	
方法	单一模型集成
调整超参数	
多层感知处理器	0.880±0.007
多层线性规划	0.883±0.005
TabR-S	0.880±0.008
泰伯	0.883±0.005
CatBoost	0.879±0.009
XGBoost	0.883±0.009
轻量级GBM0.	0.879±0.007
默认超参数	
TabR-S	0.877±0.006
CatBoost	0.879±0.007
XGBoost	0.883±0.010
轻量级GBM0.	0.884±0.005

房屋↓	
方法	单一模型集成
调整超参数	
多层感知处理器	0.233±0.002
多层线性规划	0.228±0.002
TabR-S	0.199±0.001
泰伯	0.201±0.002
CatBoost	0.216±0.001
XGBoost	0.219±0.001
轻量级GBM0.	0.219±0.001
默认超参数	
TabR-S	0.200±0.001
CatBoost	0.216±0.000
XGBoost	0.234±0.000
轻量级GBM0.	0.226±0.000

詹尼斯↑	
方法	单一模型集成
调整超参数	
多层感知处理器	0.785±0.003
多层线性规划	0.799±0.003
TabR-S	0.798±0.002
泰伯	0.805±0.002
CatBoost	0.798±0.002
XGBoost	0.797±0.002
轻量级GBM0.	0.796±0.002
默认超参数	
TabR-S	0.795±0.002
CatBoost	0.795±0.001
XGBoost	0.783±0.000
轻量级GBM0.	0.794±0.000

医疗费用↓	
方法	单一模型集成
调整超参数	
多层感知处理器	0.082±0.000
多层线性规划	0.081±0.000
TabR-S	0.081±0.000
泰伯	0.081±0.000
CatBoost	0.082±0.000
XGBoost	0.082±0.000
轻量级GBM0.	0.082±0.000
默认超参数	
TabR-S	0.082±0.000
CatBoost	0.082±0.000
XGBoost	0.084±0.000
轻量级GBM0.	0.083±0.000

nyc-taxi-green-dec-2016 ↓		
方法	单一模型	集成
调整超参数		
多层感知处理器	0.397 ±0.001	0.391 ±0.001
多层线性规划	0.368 ±0.002	0.364 ±0.000
TabR-S	0.358 ±0.022	0.338 ±0.003
泰伯	0.372 ±0.009	0.350 ±0.003
CatBoost	0.365 ±0.001	0.363 ±0.000
XGBoost	0.379 ±0.000	0.379 ±0.000
轻量级GBM0.	0.369 ±0.000	0.368 ±0.000
默认超参数		
TabR-S	0.389 ±0.001	0.385 ±0.000
CatBoost	0.366 ±0.000	0.366 ±0.000
XGBoost	0.386 ±0.000	0.386 ±0.000
轻量级GBM0.	0.372 ±0.000	0.372 ±0.000

音素 ↑		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.851 ±0.014	0.861 ±0.013
多层线性规划	0.866 ±0.012	0.875 ±0.012
TabR-S	0.878 ±0.010	0.884 ±0.005
泰伯	0.877 ±0.009	0.885 ±0.007
CatBoost	0.883 ±0.012	0.890 ±0.005
XGBoost	0.868 ±0.017	0.877 ±0.016
轻量级GBM0.	0.870 ±0.013	0.873 ±0.013
默认超参数		
TabR-S	0.877 ±0.007	0.880 ±0.003
CatBoost	0.879 ±0.011	0.881 ±0.012
XGBoost	0.870 ±0.016	0.870 ±0.016
轻量级GBM0.	0.874 ±0.007	0.874 ±0.007

韋爾 ↑		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.671 ±0.013	0.677 ±0.013
多层线性规划	0.744 ±0.019	0.767 ±0.027
TabR-S	0.874 ±0.008	0.880 ±0.006
泰伯	0.884 ±0.016	0.891 ±0.013
CatBoost	0.790 ±0.007	0.793 ±0.005
XGBoost	0.797 ±0.012	0.799 ±0.012
轻量级GBM0.	0.781 ±0.010	0.787 ±0.007
默认超参数		
TabR-S	0.871 ±0.008	0.876 ±0.007
CatBoost	0.785 ±0.010	0.790 ±0.004
XGBoost	0.775 ±0.003	0.775 ±0.003
轻量级GBM0.	0.778 ±0.003	0.778 ±0.003

颗粒物-ukair-2017 ↓		
方法	单一模型	集成
调整超参数		
多层感知处理器	0.377 ±0.001	0.374 ±0.000
多层线性规划	0.367 ±0.001	0.366 ±0.000
TabR-S	0.361 ±0.000	0.359 ±0.000
泰伯	0.360 ±0.000	0.358 ±0.000
CatBoost	0.365 ±0.000	0.364 ±0.000
XGBoost	0.364 ±0.000	0.364 ±0.000
轻量级GBM0.	0.364 ±0.000	0.363 ±0.000
默认超参数		
TabR-S	0.361 ±0.001	0.359 ±0.000
CatBoost	0.366 ±0.000	0.366 ±0.000
XGBoost	0.368 ±0.000	0.368 ±0.000
轻量级GBM0.	0.366 ±0.000	0.366 ±0.000

波尔 ↓		
方法	单一模型	集成
调整超参数		
多层感知处理器	5.659 ±0.543	5.143 ±0.579
多层线性规划	0.615 ±0.137	2.445 ±0.073
TabR-S	0.071 ±0.537	5.558 ±0.404
泰伯	0.577 ±0.169	2.326 ±0.058
CatBoost	0.632 ±0.101	3.551 ±0.090
XGBoost	0.296 ±0.064	4.255 ±0.049
轻量级GBM4.	0.232 ±0.337	4.188 ±0.311
默认超参数		
TabR-S	6.200 ±0.396	5.804 ±0.248
CatBoost	0.479 ±0.051	4.400 ±0.039
XGBoost	0.249 ±0.183	5.249 ±0.197
轻量级GBM4.	0.382 ±0.195	4.382 ±0.210

道路安全 ↑		
方法	单一模型	集成
调整超参数		
多层感知处理器	0.786 ±0.001	0.789 ±0.000
多层线性规划	0.785 ±0.002	0.789 ±0.001
TabR-S	0.840 ±0.001	0.844 ±0.000
泰伯	0.837 ±0.001	0.843 ±0.000
CatBoost	0.801 ±0.001	0.802 ±0.000
XGBoost	0.810 ±0.002	0.813 ±0.000
轻量级GBM0.	0.798 ±0.001	0.800 ±0.000
默认超参数		
TabR-S	0.791 ±0.003	0.796 ±0.003
CatBoost	0.792 ±0.001	0.793 ±0.000
XGBoost	0.796 ±0.000	0.796 ±0.000
轻量级GBM0.	0.803 ±0.000	0.803 ±0.000

硫↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.022±0.002	0.021±0.002
多层线性规划	0.020±0.002	0.019±0.003
TabR-S	0.022±0.002	0.021±0.002
泰伯	0.022±0.003	0.020±0.003
CatBoost	0.019±0.002	0.019±0.002
XGBoost	0.020±0.002	0.020±0.002
轻量级GBM	0.020±0.002	0.020±0.002
默认超参数		
TabR-S	0.021±0.003	0.021±0.002
CatBoost	0.019±0.002	0.019±0.003
XGBoost	0.022±0.002	0.022±0.002
轻量级GBM	0.021±0.001	0.021±0.001

土壤可视化↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.138±0.012	0.132±0.010
多层线性规划	0.158±0.067	0.144±0.060
TabR-S	0.398±0.352	0.387±0.375
泰伯	0.227±0.264	0.202±0.147
CatBoost	0.055±0.006	0.047±0.006
XGBoost	0.176±0.071	0.154±0.054
轻量级GBM	0.062±0.016	0.062±0.017
默认超参数		
TabR-S	0.327±0.254	0.310±0.257
CatBoost	0.064±0.005	0.058±0.005
XGBoost	0.066±0.009	0.066±0.010
轻量级GBM	0.061±0.013	0.061±0.014

葡萄酒品质↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	0.672±0.015	0.659±0.016
多层线性规划	0.654±0.018	0.634±0.018
TabR-S	0.632±0.010	0.620±0.010
泰伯	0.641±0.011	0.620±0.007
CatBoost	0.609±0.013	0.606±0.014
XGBoost	0.604±0.013	0.602±0.014
轻量级GBM	0.613±0.014	0.612±0.014
默认超参数		
TabR-S	0.628±0.015	0.614±0.015
CatBoost	0.628±0.012	0.626±0.012
XGBoost	0.648±0.008	0.648±0.008
轻量级GBM	0.641±0.011	0.641±0.012

超导↓		
方法	单一模型	合奏
调整超参数		
多层感知处理器	10.724±0.062	10.455±0.005
多层线性规划	10.566±0.058	10.334±0.028
TabR-S	10.884±0.107	10.480±0.028
泰伯	10.384±0.056	10.137±0.023
CatBoost	10.242±0.022	10.212±0.006
XGBoost	10.161±0.020	10.141±0.002
轻量级GBM	10.163±0.012	10.155±0.005
默认超参数		
TabR-S	10.812±0.110	10.423±0.046
CatBoost	10.263±0.028	10.222±0.006
XGBoost	10.736±0.000	10.736±0.000
轻量级GBM	10.471±0.000	10.471±0.000

葡萄酒↑		
方法	单一模型集成	
调整超参数		
多层感知处理器	0.769±0.015	0.784±0.010
多层线性规划	0.771±0.016	0.783±0.014
TabR-S	0.794±0.011	0.805±0.006
泰伯	0.780±0.015	0.795±0.012
CatBoost	0.799±0.013	0.806±0.010
XGBoost	0.795±0.018	0.801±0.019
轻量级GBM	0.789±0.016	0.793±0.011
默认超参数		
TabR-S	0.791±0.012	0.800±0.008
CatBoost	0.796±0.010	0.799±0.010
XGBoost	0.796±0.010	0.796±0.010
轻量级GBM	0.798±0.004	0.798±0.004

年↓		
方法	单一模型集成	
调整超参数		
多层感知处理器	8.964±0.018	8.901±0.003
多层线性规划	0.927±0.013	8.901±0.006
TabR-S	0.007±0.015	8.913±0.009
泰伯	0.972±0.010	8.917±0.003
CatBoost	0.037±0.007	9.005±0.003
XGBoost	0.031±0.003	9.024±0.001
轻量级GBM	9.020±0.002	9.013±0.001
默认超参数		
TabR-S	9.067±0.022	8.893±0.008
CatBoost	0.073±0.008	9.046±0.001
XGBoost	0.376±0.000	9.376±0.000
轻量级GBM	9.214±0.000	9.214±0.000

yprop 4 1 ↓	
方法	单一模型集成
调整超参数	
多层感知处理器	0.027 ± 0.001 0.027 ± 0.001 0
多层线性规划	。027 ± 0.001 0.027 ± 0.001 0
TabR-S	。027 ± 0.000 0.027 ± 0.001 0
泰伯	。027 ± 0.000 0.027 ± 0.000 0
CatBoost	。027 ± 0.000 0.027 ± 0.001 0
XGBoost	。027 ± 0.001 0.027 ± 0.001
轻量级GBM	0.027 ± 0.000 0.027 ± 0.000
默认超参数	
TabR-S	0.027 ± 0.001 0.027 ± 0.001 0
CatBoost	。027 ± 0.000 0.027 ± 0.000 0
XGBoost	。027 ± 0.001 0.027 ± 0.001
轻量级GBM	0.027 ± 0.000 0.027 ± 0.000