将 GBDT 与 DNN 结合起来：提高效率和效果 使用树混合 MLP 进行表格预测

严家焕
浙江大学
中国杭州
jyansir@zju.edu.cn

陈金泰*
伊利诺伊大学
厄巴纳-香槟
美国伊利诺伊州厄巴纳
jtchen721@gmail.com

王乾兴
浙江大学
中国杭州
w.qianxing@zju.edu.cn

陈哲艺
圣母大学
美国印第安纳州圣母大学
dchen@nd.edu

吴建
浙江大学
中国杭州
wujian2000@zju.edu.cn

抽象的

表格数据集在各种应用中起着至关重要的作用。因此，开发高效、有效且广泛兼容的表格数据预测算法非常重要。目前，两种主要的模型类型，梯度提升决策树 (GBDT) 和深度神经网络 (DNN)，已在不同表格预测任务上表现出性能优势。然而，为特定的表格数据集选择有效的模型具有挑战性，通常需要耗时的超参数调整。为了解决这个模型选择难题，本文提出了一个新框架，融合了 GBDT 和 DNN 的优势，从而产生了一种与 GBDT 一样高效且具有竞争力的 DNN 算法，无论数据集偏好 GBDT 还是 DNN。我们的想法源于这样一个观察：深度学习 (DL) 提供了更大的参数空间，可以表示性能良好的 GBDT 模型，但当前的反向传播优化器难以有效地发现这种最佳功能。另一方面，在 GBDT 开发过程中，硬树剪枝、熵驱动的特征门和模型集成已被证明更适合表格数据。通过结合这些关键组件，我们提出了一种 电视混合简单会员权益 (T-MLP）。在我们的框架中，张量化、快速训练的 GBDT 特征门控、DNN 架构修剪方法以及原始反向传播优化器协同训练随机初始化的 MLP 模型。综合实验表明，T-MLP 在主要表格基准测试（88 个数据集）中分别与经过广泛调整的 DNN 和 GBDT 具有竞争力，并且均通过紧凑的模型存储和显著缩短的训练时间实现。代码和完整的实验结果可在 https://github.com/jyansir/tmlp 上找到。

CCS 概念

•计算方法→*机器学习*；监督学习；神经网络。

关键词

分类和回归、表格数据、绿色 AI、AutoML

ACM 参考格式：
闫家焕，陈金泰*、Qianxing Wang、Danny Z. Chen 和 Jian Wu。2024. 联手 GBDT 和 DNN：使用树混合 MLP 推进高效且有效的表格预测。在*第 30 届 ACM SIGKDD 知识发现和数据挖掘会议（KDD '24）论文集，2024 年 8 月 25 日至 29 日，西班牙巴塞罗那。*ACM，纽约，纽约州，美国，14 页。https://doi.org/10.1145/3637528.3671964

1 引言

表格数据是各种机器学习应用中普遍存在且占主导地位的数据结构（例如，点击率 (CTR) 预测 [17] 和金融风险检测 [3]）。当前流行的表格预测（即分类和回归）模型通常可分为两大类：（1）梯度提升决策树 (GBDT) [16, 18, 31, 43]，这是一种经典的非深度学习方法，已被广泛验证为经得起时间考验的解决方案 [7, 23, 55]；（2）深度神经网络 (DNN)，人们不断努力将计算机视觉 (CV) 和自然语言处理 (NLP) 中的深度学习 (DL) 技术应用于深度神经网络，以开发表格学习方法，如精细架构工程 [2, 12, 22, 42, 62] 和预训练 [48, 58, 69]。随着定制表格 DNN 的最新发展，越来越多的研究报告称它们与 GBDT 具有更好的可比性 [13, 62] 甚至优于 [14, 48]，尤其是在复杂数据场景中 [45, 58]，而经典思想认为 GBDT 在典型的表格任务中仍然完全超越 DNN [7, 23]，两者都使用不同的基准和基线进行评估，这意味着这两种模型类型各自的表格数据能力。

对于 DNN，其固有的高维特征空间和平滑的反向传播优化在非结构化数据上取得了巨大的成功 [10, 44]，并具有挖掘细微特征交互的能力 [46, 49, 57, 62]。此外，利用 DNN 的可迁移性，最近流行的表格 Transformers 可以通过昂贵的预训练进一步改进 [48, 58, 69]，就像它们的同行一样

•通讯作者。

KDD '24，2024 年 8 月 25-29 日，西班牙巴塞罗那
ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671964

NLP [10, 32, 68]。然而，与简单的多层感知器 (MLP) 和 GBDT 相比，Transformer 架构更加复杂，并且容易*过度参数化、数据匮乏并增加处理延迟*，尤其是最近基于语言模型的架构 [8, 67]。因此，它们通常在可能较小的表格数据集上表现不佳 [23]。

至于 GBDT，它们在贪婪特征选择、树剪枝和高效集成方面表现优异，在大多数表格预测应用中取得了显著的性能和效率 [47, 55]。然而，它们通常*超参数敏感 [43, 63]，不太适合极端表格场景*例如具有复杂特征交互的大规模表格[45]。此外，随着数据规模的增长，它们的推理延迟显著增加[7]。

此外，GBDT 和 DNN 框架都取得了各自的最佳结果，*昂贵的培训费用*，自从*重度超参数搜索*是实现可观性能的必要条件。但是，这不利于碳排放，并且与计算受限或实时应用不兼容，同时在经济表格预测方面还没有做出足够的积极努力。

为了解决模型选择难题，我们综合了 GBDT 和 DNN 的优点，提出了一种新的电视混合简单会员权益 (我们提出了一种高性能、高效且轻量级的 T-MLP 模型。具体来说，单个 T-MLP 配备 GBDT 特征门，以贪婪方式执行特定于样本的特征选择，并配备受 GBDT 启发的剪枝 MLP 架构来处理所选的显着特征。整个框架利用这些 GBDT 的属性通过反向传播进行优化，所有组件使系统紧凑、抗过拟合和可推广。此外，通过使用共享门并行训练多个稀疏 MLP（我们在这里统一使用 3 个 MLP）并以 bagging 方式进行预测，可以有效地实现模型集成。总体而言，T-MLP 具有以下吸引人的特点。（1）广义数据适应性：与受模型选择困境困扰的现有表格预测方法不同，T-MLP 足够灵活，可以处理所有数据集，而不管框架偏好如何（参见第 4.2 节和第 4.4 节）。（2）免费调整超 参数：T-MLP 能够产生有竞争力的结果 所有配置都已预先固定，这显著节省时间、方便用户、环保且具有广泛的应用价值。（3）轻量级存储：在 T-MLP 中，DNN 部分纯粹由简单且高度稀疏的 MLP 架构组成，但即使与单块 MLP 相比，仍然能够达到最先进的水平。表 1 展示了 T-MLP 与普通 DNN 相比的经济成本性能权衡；随着数据规模的增长，这种成本效益变得更加显著。

总而言之，我们的主要贡献如下：

- 我们提出了一种新的 GBDT-DNN 混合框架 T-MLP，它是一种一站式、经济的解决方案，可实现有效的表格数据预测，而不受特定数据集的框架偏好影响，为表格模型架构提供了一种新颖的优化范式。
- 从特征选择策略、参数稀疏性、决策边界模式等多方面进行分析，深入了解T-MLP的效率和优越性。
- 对来自 4 个基准的 88 个数据集（涵盖 DNN 和 GBDT 青睐的数据集）进行的全面实验表明

表 1：流行表格 DNN 在小型和大型数据集上的模型成本效益比较。 和 表示特征和样本的数量， 是参数编号，并且 表示针对所提出的 T-MLP 的总训练时间开销。我们重用了 FT-Transformer 基准中最佳模型配置的性能和参数大小。 在 NVIDIA A100 PCIe 40GB 上进行评估（见第 4.1 节）。基于固定的架构和训练配置，T-MLP 无论数据规模如何，都能实现稳定的模型大小和低廉的训练时间成本。

数据集：	成人 (=14， =4.9万）			年 （ =90， =515千）		
	（男）	ACC ↑		（男）		均方根误差 ↓
多层感知处理器	0.77	7.7×	0.852	1.16	15.9×	8.853
节点	20.83	120.4×	0.858	7.55	206.0×	8.784
自动输入	0.01	25.0×	0.859	0.08	101.9×	8.882
数据中心网络版本2	1.18	8.0×	0.853	11.32	29.9×	8.890
光纤到户	3.82	19.6×	0.859	1.25	16.3×	8.855
肌萎缩性骨髓纤维化	0.73	1.0×	0.864	0.75	1.0×	8.768

单个 T-MLP 可以与高级或预训练的 DNN 相媲美，而 T-MLP 集成甚至可以持续超越它们，并且可以与经过广泛调整的最先进的 GBDT 相媲美，所有这些都是通过紧凑的模型尺寸和显着缩短的训练时间实现的。

- 我们开发了一个开源 Python 包，其中包含基准加载、统一基线调用（DNN、GB-DT、T-MLP）、DNN 修剪和其他高级功能的 API，作为表格学习社区的开发工具。

2 相关工作

2.1 表格预测的模型框架

在过去的二十年中，经典的非深度学习方法 [25, 35, 65, 66] 在表格预测应用中非常流行，尤其是 GBDT [16, 18, 31, 43]，因为它们在典型的表格任务 [23] 中表现出色且具有鲁棒性。由于 DNN 在非结构化数据上的普遍成功以及计算设备的发展，人们越来越多地将 DNN 应用于此类任务。早期的表格 DNN 旨在通过模拟集成树框架（例如 NODE [42]、Net-DNF [30] 和 TabNet [2]）来与 GBDT 相媲美，但它们忽略了 DNN 在自动特征融合和交互方面的优势。因此，最近的尝试充分利用了 DNN 的优势，它们迁移了成功的神经架构（例如 AutoInt [49]、FT-Transformer [22]），提出了定制设计（例如 T2G-Former [62]），或采用了预训练（例如 SAINT [48]、TransTab [58]），与特定数据场景中传统上占主导地位的 GB-DT 相比，报告了具有竞争力甚至超越的结果 [14, 48]。当代调查 [7] 表明，GBDT 和 DNN 是当前表格学习研究中两种主要的框架类型。

2.2 轻量级 DNN

轻量级 DNN 是 CV 和 NLP 领域中一个常青的研究课题，其目的是在保持有效性能的同时提高 DNN 的紧凑性和效率。最近的趋势是用纯粹的简单 MLP 替代主导主干，例如 MLP-Mixer [53]，

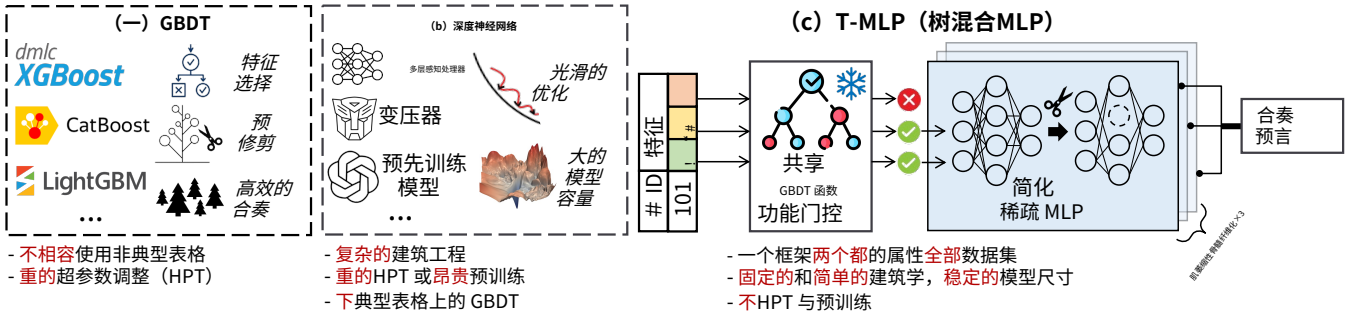

图 1：我们提出的 T-MLP 与现有表格预测方法：GBDT 和 DNN。（a）GBDT 是用于表格预测的经典非深度学习模型。（b）DNN 是一种新兴的有前途的方法，特别是对于大规模、复杂、跨表场景。（c）T-MLP 是一个混合框架，它集成了 GBDT 和 DNN 的优势，通过 GBDT 特征门张量化、MLP 框架修剪、简单块集成和端到端反向传播实现。它产生

DNN 和 G 上的竞争结果 屋宇署睐的数据集，具有快速的开发过程 更小、更紧凑的模型尺寸。

gMLP [36]、MAXIM [54] 和其他视觉机器学 附言1, 24, 51]，实现习算法在容量降低的情况下，也能取得与前 他CNN 或 Trans-者相当甚至更好的结果。在 NLP [19] 和其他或者LOP。这种纯算法中，MLP 也呈现出类似的趋势。在表格她 现实世界的应用预测领域，有少数算法专注于正则化 [29] 或 米模型压缩，数值，而不是 DNN 架构本身。除了表格 日期减少大 ity [38, DNN 之外，其他算法尚未被探索。 拉尔0, 61]。在尤瑞MLP 研究，但 l伊嵌入 [21]秒, 模型压缩我们介绍相关技术使我们的T-MLP更加紧凑和有效。

3.3 树型混合简单MLP

我们首先回顾一下当前基于 Transformer 的表格 DNN 中典型 GBDT 的推理过程和特征编码技术的一些初步情况。接下来，我们详细介绍了 T-MLP 的几个关键组件的详细设计，包括用于特定样本特征选择的 GBDT 特征门、纯 MLP 基本块和 GBDT 启发的稀疏 MLP 细粒度剪枝。最后，我们讨论了 T-MLP 工作流程。

3.1 准备工作

问题陈述。给定具有输入特征的表格数据集 ∈R × 和目标 ∈R，表格预测任务是找到最优解决方案： ∈R × →R 最小化预测之间的经验差异 和目标 。在目前的实践中，常见的选择是 是传统的 GBDT（例如 XGBoost [16]、CatBoost [43]、LightGBM [31]）或表格 DNN（例如 TabNet [2]、FT-Transformer [22]、SAINT [48]、T2G-Former [62]）。典型的差异度量是分类任务的准确率或 AUC 分数，是回归的均方误差根 (RMSE)。

定义3.1：GBDT特征频率。给定一个 GBDT 模型 决策树（例如 CART [35]），*GBDT 特征频率* 表示该 GBDT 在样本上访问每个特征的次数。具体来说，GBDT 的过程

样本推断 ∈R pr 预测 （）=大 奥维德斯 乘以单个决策树车（） （），树预测，存在一 ∈{1，2，...，}。为萨每一个决定个它的根节点到其中一个叶子 姆 请 特殊虚构决策 n 路径来自节点包含涉及访问的特征列表是，为了来一个 使用壮举你列出中的特征 -t （）∈{0，1}，其是普雷通动作 动作。We 表示这个中 0 指示此样本未被访问。因 时涨胜离子稀土元素向量此，*频率*样本的第个向量如 在是 -t的 对应 叮功能下： 埃 - 第 d埃西西，并且 1 i 表明我们能 r埃普雷送格 *BDT 功能*西嗯哦f 决定 树的二进制

$$= \sum_{(),}$$

在哪里 表示GBDT中各个特征的开发程度，表明GBDT模型对该样本的特征偏好。

功能标记器。受经典语言模型（例如 BERT [32]）的启发，最近占主导地位的基于 Transformer 的表格模型 [22, 48, 62] 通过将表格值嵌入向量空间并将这些值视为“无序”词向量，采用了分布式特征表示 [39]。此类 Transformer 模型使用*特征标记器*[22]处理表格特征如下：每个表格标量值被映射到一个向量 ∈R 使用特定于特征的线性投影，其中 是特征隐藏维度。对于数值（连续）值，投影权重与值幅度相乘。给定 1数字特征和 2分类特征，特征标记器输出特征嵌入 ∈R (1+ 1+ 2) × 通过堆叠投影（特征（以及额外的 [CLS] 标记）嵌入），即 =

堆 中立证券， (1数量，...， (数量， (猫，...， (猫）。

3.2 GBDT 特征门

表格 DNN 的早期尝试试图通过集成神经网络来构建 GBDT 的行为模式*差异树模型*，如代表模型 NODE [42] 和 TabNet [2]。然而，即使实现了类似决策树的硬特征选择，或者借助复杂的 Transformer 架构，它们仍然

随后，这一领域迅速被淹没在了以深度学习视角提升为主要目标的后续 DNN 研究中 [13, 22, 62]。我们试图重新思考这一领域的工作，并观察到他们通过 DNN 的平滑反向传播实现了具有可学习连续特征掩码的硬特征选择，这可能与 GBDT 的离散性质不相容，因此限制了它们的潜力。

为了解决这个问题，我们建议*GBDT 特征门控*（GFG）是一种基于 GBDT 的特征选择器，使用 GBDT 权重进行张量化，以忠实地复制其特征选择行为。具体而言，给定一个由 -树 GBDT，特征选择过程 -功能示例 （ =GFG（) ∈R × ）公式为：

$$=\text{特征标记器}（\ ）\in \mathrm{R}\times，\ =\text{GBDT特} \tag{1}$$

$$\text{征频率}（\ ）\in \mathrm{R}， \tag{2}$$

$$= / \in \mathrm{R}，\ =\text{二进制采样器}（\ ）\in \{0，1\}， \tag{3}$$

$$：，= \begin{cases} \odot：，\ \ \text{如果}\textit{训练} \\ \odot：，\ \ \text{如果}\textit{推理} \end{cases} ，\in\{1，2，\ldots，\}。 \tag{4}$$

为了符号简洁，本节省略了额外的 [CLS] 嵌入；在实现中，它直接连接到门控的头部 在公式（3）中， 是归一化的 GBDT 特征频率，表示每个特征在 -树 GBDT，以及 是一个用概率采样的二进制特征掩码 为了将 GBDT 的特征偏好纳入 DNN 框架，在公式 (4) 中，我们使用来自真实 GBDT 特征访问概率的稀疏特征掩码在训练期间执行硬特征选择，并在推理期间使用软概率进行确定性预测。GFG 协助根据 GBDT 的特征偏好过滤掉不必要的特征，与以前的使用神经网络学习特征掩码的差分树模型相比，确保了 oracle 选择行为。

由于原始 GBDT 库（我们在本文中统一使用 XGBoost）没有用于高效获取公式 (2) 中样本特定 GBDT 特征频率的 API，并且使用的后端与常见的 DL 库（例如 PyTorch）不兼容，为了将 GFG 模块集成到并行 DNN 框架中，我们将公式 (2) 的行为张量化。从技术上讲，我们受到了 Microsoft Hummingbird 编译工具原理的启发[1]从XGBoost模型中提取路由矩阵，即一系列参数矩阵，其中包含每棵决策树的节点邻接性和阈值信息。基于提取的路由矩阵，可以通过交替对输入特征进行张量乘法和比较来简单地获取特征访问频率 ，并利用这些参数矩阵初始化公式（2）的子模块。

在实际实施中，我们只需快速训练一个具有 [22] 中提供的统一默认超参数的 XGBoost（无论其性能如何），以在 T-MLP 初始化步骤中初始化和冻结公式 (2) 的子模块。其他可训练参数是随机初始化的。由于 GBDT 模型中有大量决策树用于对特征偏好进行投票，因此轻微的超参数修改不会改变整体特征偏好趋势，并且轻度训练的默认 XGBoost 始终足以指导贪婪特征选择。为了进一步加快公式 (2)-(3) 中的过程，我们缓存了标准化的特征频率 在第一个 epoch 计算期间对每个样本进行缓存，并在后续的模型训练或推理中重用缓存。

[1]https://github.com/microsoft/hummingbird

3.3 纯MLP基本块

为了探索纯 MLP 架构的能力并保持我们的表格模型紧凑，我们从视觉 MLP 中汲取了灵感。我们观察到，它们成功的一个关键因素是通过线性投影和软门控实现的类似注意力的交互 [11, 24, 53]。因此，我们采用了 [36] 中提出的空间门控单元 (SGU)，并制定了一个简化的纯 MLP 块，如下所示：

$$（+1）= \mathrm{SGU}(\mathrm{GELU}(\mathrm{LayerNorm}（\ ）\ \ 1））\ \ 2+（），\tag{5}$$

$$\text{圣乔治大学}(\ ) = 3\text{层范数}(：，：，)\odot：，：。 \tag{6}$$

该块类似于单个前馈神经网络（FFN）
在 Transformer 中使用额外的 SGU（等式 (6)）进行特征级交互。主要参数位于两个转换中，
IE，$1\in\mathrm{R}\times 2$ 和 $2\in\mathrm{R}\times$ 在公式 (5) 中， 对应于FFN中间维度大小。在等式（6）中， $\in\mathrm{R}\times 2$
表示 SGU 的输入特征， $3\in\mathrm{R}\times$ 是一个特征级的转换，用来模拟注意力机制。由于 $\approx \gg$ 在大多数情况下，模型大小由以下因素决定 1和 2，与 FFN 大小相当。为简化符号，省略了所有偏差向量。

与视觉数据类似，我们将表格特征和特征嵌入视为图像像素和通道。但与视觉 MLP 完全不同的是，T-MLP 是一个混合框架，专门为经济的表格预测而量身定制，其性能可与表格 Transformers 和 GBDT 相媲美，同时运行成本显著降低。在大多数不复杂的数据集上，在 T-MLP 中仅使用一个基本块就足够了。相比之下，之前的视觉 MLP 研究强调架构工程，并且通常需要数十个块才能与视觉 Transformers 相媲美。

3.4 用户可控制的剪枝的稀疏性

受 GBDT 预剪枝的启发，该预剪枝可以控制模型复杂度并通过用户定义的超参数（例如最大树深度、每片叶子的最小样本）促进泛化，我们利用主要的模型压缩方法（即 DNN）为 T-MLP 设计了类似的机制*修剪*[26, 38, 50]，该算法在 NLP 研究中被广泛使用，用于在保持原有可靠性的同时，削减过度参数化的语言模型 [61]。

具体来说，我们引入两个细粒度变量 时长$\in\{0，1\}$ 和 在$\in\{0，1\}$，分别从隐藏维度和中间维度掩蔽参数。与语言模型中先前的 FFN 修剪一样 [59]，T-MLP 修剪操作可以通过简单地将掩码变量应用于权重矩阵来实现，即替换 1和 2使用 diag(时长） 1 和诊断（ 在） 2

在方程（5）中。我们使用经典的 0正则化用硬具体分布重新参数化[37]，并采用拉格朗日乘数目标来实现可控的稀疏性，如[61]所示。

虽然早期的表格 DNN 尝试已经考虑了稀疏结构，例如 TabNet [2] 和 NODE [42] 构建了可学习的稀疏特征掩码，最近的 TabCaps [12] 和 T2G-Former [62] 设计了稀疏特征交互，但仍存在两个本质区别：（1）现有的表格 DNN 仅考虑特征维度上的稀疏性，而 T-MLP 引入了稀疏性

在输入特征（第 3.2 节）和隐藏维度（本小节）上，这在以前的表格 DNN 预测研究中被忽略，并被广泛认为是 NLP 实践中过度参数化的方面 [59, 61]；（2）现有表格 DNN 中的可学习稀疏性完全耦合并由预测损失函数决定，而我们引入的 DNN 修剪技术根据用户定义的稀疏率（与目标无关）确定稀疏性，具有与 GBDT 预修剪相同的可控性质。

在主要实验（第 4.2 节）中，我们将 T-MLP 的目标稀疏度统一固定为 0.33，即训练后仅保留约 33% 的 DNN 参数。我们在第 4.3 节中进一步探讨了模型稀疏度与性能之间的关系，并通过适当的参数修剪获得了性能提升，即使在具有一个基本块的 T-MLP 上也是如此，这意味着以前的表格 DNN 设计中普遍存在过度参数化。

3.5 整体工作流程和高效集成

T-MLP 的整体工作流程如下：在训练阶段，输入的表格特征被嵌入到特征标记器中，并通过采样特征掩码进行离散选择 在公式 (3) 中；然后，它们由公式 (5) 中的单个修剪基本块处理，并且修剪参数掩码 时长和 在经过重新参数化采样 0正则化；最终的预测是使用 [CLS] 标记特征和正常预测头进行的，就像在其他表格 Transformers 中一样，如下所示：

$$=\mathrm{FC}(\mathrm{ReLU}(\text{层范数}(\ (\ )\ _{[\mathrm{CLS}],\ :}^{\ )\ )\ )},$$

其中 FC 表示全连接层。我们使用交叉熵损失进行分类，使用均方误差损失进行回归，就像以前的表格 DNN 一样。整个框架通过反向传播进行优化。训练后，参数掩码直接应用于 1和 2通过相应地删除隐藏层和中间层维度来减少输入特征。在推理阶段，输入特征通过归一化的 GBDT 特征频率进行软选择 在公式（3）中，并经过简化的基本块处理

由于 T-MLP 架构紧凑、计算友好且运行成本低，我们进一步提供了一种*高效集成*通过使用共享 GBDT 特征门从同一初始化点以三个固定学习率同时训练三个分支，可以实现不同的效果。这会产生三个不同的稀疏 MLP，灵感来自*模范汤*集成方法 [60]。最终的集成预测是三个分支的平均结果，就像 bagging 集成模型一样。由于集成学习过程可以通过使用多处理编程（例如 RandomForest [9]）同时进行训练和推理来实现，因此训练时间不是增加三倍，而是由最慢的收敛分支决定。

4 实验

在本节中，我们首先将我们的 T-MLP 与高级 DNN 和经典 GBDT 在主要基准（包括 88 个不同任务类型的数据集）上进行比较，并从成本效益的角度进行分析。接下来，我们进行消融和比较实验，并进行多方面分析，以评估使 T-MLP 有效的关键设计。此外，我们比较了优化的

通过可视化决策边界来查看常见 DNN、GBDT 和 T-MLP 的模式，以进一步检验 T-MLP 的优越性。

4.1 实验装置

数据集。我们使用四个最近的高质量表格基准（FT-Transformer[2]（FT-T，11 个数据集）[22]、T2G-Former[3]（T2G，12 个数据集）[62]，SAINT[4]（26 个数据集）[48] 和 Tabular Benchmark[5]（TabBen，39 个数据集）[23]），考虑到它们在广泛的基线和数据集上得出的详细结果。FT-T 和 T2G 基准代表了大规模表格数据集，其大小从 10K 到 1,000K 不等，包括各种 DNN 基线。SAINT 基准是从 OpenML 存储库中收集的[6]，以预训练的 DNN SAINT 为主，包含均衡的任务类型和多样的 GBDT。TabBen 基于“典型表格数据”设置，限制了数据集属性，例如数据规模（最大数据量为 10K）和特征数量（非高维）[23]，并将数据集分为几类，并结合任务类型和特征特点。值得注意的是，在 TabBen 上，GBDT 取得压倒性胜利，超越了常用的 DNN。每个基准测试都代表一种特定的框架偏好。由于一些基准测试已经调整了其当前存储库中的数据集安排（例如，删除了一些数据集并添加了一些数据集），为了忠实地遵循和重用结果，我们仅保留了已发表的原始论文中报告的数据集。我们在表 2 中提供了详细的基准测试统计信息，并在附录 A 中讨论了基准测试特征。

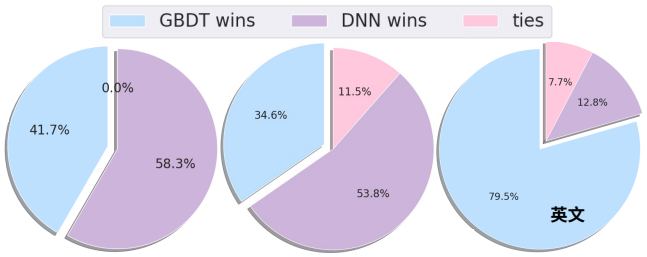

图 2：GBDT 和 DNN 在三个基准测试中的获胜率，表示每个框架在基准测试中取得最佳性能的比例。它展示了不同表格预测工作中使用的数据集中框架的不同偏好。

实施细节。我们使用 Python 3.10 上的 PyTorch 实现了 T-MLP 模型和 Python 包。由于原始基准上报告的基线训练持续时间是在不同的运行环境下使用不同的评估代码估算的，并且不考虑超参数调整 (HPT) 预算，为了统一比较训练成本，我们在构建的包中使用与 T-MLP 相同的 sklearn 风格 API 封装实验基线，并在

[2]https://github.com/yandex-research/rtdl-revisiting-models/tree/main
[3]https://github.com/jyansir/t2g-former/tree/master
[4]https://github.com/somepago/saint/tree/main
[5]https://github.com/LeoGrin/tabular-benchmark/tree/main
[6]https://www.openml.org

表 2：四个实验基准的数据集统计。“# bin.、# mul. 和 # reg.”表示二分类、多分类和回归数据集的数量。“# small、# middle、# large 和 # ex. large”表示小（ ≤3K）、中（3K）<≤10K）、大（10K）<≤100K）和极大（ >100K）数据集，其中 表示训练数据的大小。“# wide 和 # ex. wide” 是宽度的量（32<≤64）和极宽（ >64) 数据集，其中 是特征量。“bin. metric、mul. metric 和 reg. metric” 表示基准中针对每种任务类型使用的评估指标。“R-Squared” 分数是判定系数。

	# bin.	# mul.	# reg.	# small	# 中间	# 大	# 例如	大 # 宽	# 例如	宽箱	公制	多度量标准度量	
傅立叶变换红外光谱 [22]	3	4	40			0	6525	ACC				ACC	均方根误差
泰坦尼克 [62]	3	5	40			3	7222	ACC				ACC	均方根误差
圣人 [48]	9	7	10	10	3	12169	AUC					ACC	均方根误差
塔布本 [23]	15	0	24	2	三十七	0052	ACC					不适用	R 平方

表 3：FT-T 基准上的成本效益比较。分类数据集和回归数据集分别使用准确度和 RMSE 指标进行评估。“等级”表示数据集中所有方 法的平均值（标准差）。“ ”表示针对 T-MLP 训练所用时间的平均开销，“ *“仅比较获得最佳验证分数之前的持续时间。所有训练持续时间均使用原始超参数搜索设置进行估算。” ” 表示 FT-T 存储库提供的最佳模型配置的平均参数数。由于评估中后端（Tensorflow）不同，因此未与 TabNet 进行比较。最佳性能以粗体标记，第二好性能以下划线标记（后续表格中使用类似标记）。

	加州 ↓	广告 ↑	他 ↑	JA ↑	你好 ↑	艾尔 ↑	EP ↑	叶 ↓	一氧化碳 ↑	呀 ↓	MI ↓	秩	*	（男）
塔格网	0.510	0.850	0.378	0.723	0.719	0.954	0.8896	8.909	0.957	0.823	0.751	9.0 (1.5)	不适用	不适用 不适用
信噪比	0.493	0.854	0.373	0.719	0.722	0.954	0.8975	8.895	0.961	0.761	0.751	7.8 (1.1)×	42.76×	24.87 1.12 0.474 0.859
自动输入	0.372	0.721	0.725	0.945	0.8949	8.882	0.934	0.768	0.750	7.4 (2.1)×	121.68×	112.31 1.14	0.487	0.857 N/AN/A
成长网	0.722	N/A	0.8970	8.827	N/A	0.765	0.751	N/AN/AN/AN/A	0.499	0.852	0.383	0.719	0.723	0.954 0.8977 8.85 3
多层感知处理器	0.962	0.757	0.747	6.5 (1.7)×	27.41×	28.46	0.55	0.484	0.853	0.385	0.716	0.723	0.955	0.8977 8.890 0.965 0.757
数据中心网络版本2	0.749	6.4 (1.8)×	31.15×	40.65	4.17	0.464	0.858	0.359	0.727	0.726	0.918	0.8958	8.784	0.9580.753 0.7455.4
节点	（3.2）	×	386.54×	353.38	16.59	0.486	0.8540.396	0.728	0.7270.963	0.8969	8.846	0.964	0.757	0.748 4.5 (2.2)×
残差网络	56.20×	58.46	6.16	0.459	0.859	0.391	0.7320.720	0.960	0.89828.855	0.9700.756	0.746	3.3（2.4）	×	117.35×97.49
光纤到户	2.12													
肌萎缩性骨髓纤维化	0.447	0.864	0.386	0.728	0.729	0.956	0.8977	8.768	0.968	0.756	0.747	3.1（0.9）	×1.00	×1.00 0.79
T-MLP(3)	0.438	0.867	0.386	0.732	0.730	0.960	0.8978	8.732	0.969	0.755	0.745	1.7（0.8）	×1.05	×1.08 2.37

表 4：T2G 基准上的成本效益比较，其符号与表 3 类似。基准性能和配置也重用了 T2G 存储库中的配置。根据 T2G 论文，对于极大的数据集 Year，FT-T 和 T2G 使用 50 次迭代超参数调整 (HPT)，DANet-28 遵循其默认超参数，其他基准结果使用 100 次迭代 HPT 获得。

	通用电气 ↑	中 ↑	安永 ↑	加州 ↓	何 ↓	广告 ↑	加时赛 ↑	他 ↑	JA ↑	你好 ↑	脸书 ↓	叶 ↓	秩	*	（男）
XGBoost	0.684	0.859	0.725	0.436	3.169	0.873	0.825	0.375	0.719	0.724	5.359	8.850	4.3（3.1）	×32.78×42.88	不适用 MLP
	0.586	0.858	0.611	0.499	3.173	0.854	0.810	0.384	0.720	0.720	5.943	8.849	8.3 (1.9)×	13.73×11.45	0.64 0.647
信噪比	0.857	0.616	0.498	3.207	0.854	0.812	0.372	0.719	0.722	5.892	8.901	8.3 (1.5)×	22.74×12.54	0.82	0.600 0.850
塔格网	0.621	0.513	3.252	0.848	0.791	0.379	0.723	0.720	6.559	8.916	10.2 (2.4)		不适用	不适用	不适用
DANet-28	0.616	0.851	0.605	0.524	3.236	0.850	0.810	0.355	0.707	0.715	6.167	8.914	10.6 (2.0) 节点	不适用	不适用 不适用
	0.539	0.859	0.655	0.463	3.216	0.858	0.804	0.353	0.728	0.725	5.698	8.777	7.0 (3.0)×	329.79×288.21	16.95 0.583
自动输入	0.855	0.611	0.472	3.147	0.857	0.801	0.373	0.721	0.725	5.852	8.862	8.1 (2.0)×	68.30×55.52	0.06	0.557 0.857
数据中心网络版本2	0.614	0.489	3.172	0.855	0.802	0.386	0.716	0.722	5.847	8.882	8.4 (2.0)×	24.40×21.63	2.30	0.613	0.861 0.708 0.460
光纤到户	3.124	0.857	0.813	0.391	0.732	0.731	6.079	8.852	4.7（2.6）	×	64.68×50.90	2.22	0.656	0.863	0.782 0.455 3.138
T2G	0.860	0.819	0.391	0.737	0.734	5.701	8.851	3.1（1.7）	×	88.93×87.04	1.19				
肌萎缩性骨髓纤维化	0.706	0.862	0.717	0.449	3.125	0.864	0.814	0.386	0.728	0.729	5.667	8.768	3.3（0.9）	×1.00×1.00	0.72
T-MLP(3)	0.714	0.866	0.747	0.438	3.063	0.867	0.823	0.386	0.732	0.730	5.629	8.732	1.9（0.8）	×1.09	×1.11 2.16

NVIDIA A100 PCIe 40GB。所有基线的超参数空间和迭代次数均遵循原始论文中的设置，以模拟每个基线的调整过程。对于 T-MLP，我们使用固定超参数，因为模型只训练一次。用于 T-MLP 的 GBDT 特征门的 XGBoost 采用 [22] 中的默认配置。在实验中，如果没有特殊说明，每个单个 T-MLP 对大多数数据集使用一个基本块。我们对单个 T-MLP 统一使用 1e-4 的学习率，对 T-MLP 集合（组“T-MLP(3)”）中的三个分支统一使用 1e-4、5e-4 和 1e-3 的学习率。我们重用与原始基准测试相同的数据分割。基线性能继承自报告的基准测试结果，基线容量根据相应论文存储库中提供的最佳模型配置计算。运行环境和超参数的详细信息在附录 C 中给出。

比较方法。在四个基准测试中，我们将 T-MLP（单模型和 3 模型集成版本）与以下模型进行比较：（1）已知的非预训练 DNN：MLP、ResNet、SNN [33]、GrowNet [4]、TabNet [2]、NODE [42]、AutoInt [49]、DCNv2 [57]、TabTransformer [28]、DANets [13]、FT-Transformer (FT-T) [22] 和 T2G-Former (T2G) [62]；（2）预训练 DNN：SAINT [48]；（3）GBDT 模型：XGBoost [16]、CatBoost [43]、LightGBM [31]、GradientBoostingTree (GBT)、HistGradientBoostingTree (HistGBT) 和其他传统的非深度机器学习方法，如 RandomForest (RF) [9]。对于其他未提及的基线，请参阅附录 B。在下面的实验表中，“T-MLP”表示单个 T-MLP，“T-MLP(3)”表示具有三个分支的集成版本。

4.2 主要结果与分析

表 3 至表 6 中的基线结果基于重 HPT，分别来自报告的基准测试。所有 T-MLP 结果均基于默认超参数。

与高级 DNN 的比较。表 3 和表 4 报告了 FT-T 和 T2G 基准上的详细性能和运行时成本，以比较我们的 T-MLP 版本和定制的表格 DNN [22, 62]。这些表中的基准结果基于 50 次（针对大型数据集上的复杂模型，例如 Year 数据集上的 FT-Transformer）或 100 次（其他情况）HPT 迭代，特殊模型除外（对于具有大量类别数量的数据集使用默认 NODE，对于所有数据集使用默认 DANet）。可以观察到的总体趋势是，单个 T-MLP 能够在每个基准上取得与最先进 DNN 相当的结果，而三个 T-MLP 的简单组合（即“T-MLP(3)”）表现出更好的性能，并且训练成本显著降低。具体而言，得益于固定的超参数和简单的结构，单个 T-MLP 实现了明显的加速，与强大的 DNN 相比，训练时间减少了几个数量级，并且比 XGBoost（一种高度依赖重度 HPT 的代表性 GBDT）更易于训练。此外，我们观察到 T-MLP 集成中的训练时间仅增加了约 10%，因为我们采用多处理编程同时训练三个 T-MLP（参见第 3.5 节），因此训练时间取决于收敛速度最慢的子模型。在实现细节（第 4.1 节）中，单个 T-MLP 使用三个子模型中最小的学习率，因此 T-MLP 的收敛时间

表 5：三种任务类型的 SAINT 基准上所有方法排名的平均值（标准差）。| |是每个组中的数据集数量。值得注意的是，所有基线结果均基于 HPT，而 SAINT 变体需要在预训练和数据增强方面进行进一步的训练预算。附录中给出了更详细的结果。

任务类型：	Binclass	多类回归 (| |=7)	(| |=9) (| |=10)
射频	7.8 (3.3)	7.3 (2.2)	9.1 (4.2)
额外的树木	7.8 (3.8)	9.6 (1.9)	8.6 (3.5)
K邻居区	13.7 (0.7)	11.6 (3.5)	12.9 (1.8)
邻居联合	14.4 (0.5)	12.4 (3.4)	14.0 (1.0)
轻量级GBM	5.7 (3.3)	3.9 (2.8)	6.5 (3.2)
XGBoost	4.2 (2.8)	6.7 (3.5)	7.3 (2.9)
CatBoost	3.9 (2.8)	7.2 (2.4)	5.6 (2.7)
多层感知处理器	10.7 (1.8)	10.1 (3.9)	不适用
神经网络FastAI	不适用	不适用	11.9 (2.2)
塔格网	13.2 (2.0)	13.5 (1.1)	10.2 (4.5)
标签转换器	10.8 (1.4)	10.0 (3.6)	10.0 (2.9)
SAINT-s	7.8 (2.4)	7.9 (6.1)	4.7 (3.8)
圣伊	7.2 (2.6)	7.1 (2.7)	5.9 (3.5)
圣	4.2 (2.7)	5.2 (2.2)	4.2 (2.3)
肌萎缩性骨髓纤维化	4.6 (2.8)	4.6 (3.0)	4.6 (3.3)
T-MLP(3)	3.9 (1.9)	2.9 (2.5)	5.0 (2.9)

集成通常近似于单个 T-MLP。从模型存储的角度来看，正如预期的那样，单个 T-MLP 的大小与整个数据集中朴素 MLP 的平均水平相当，并且其大小变化是稳定的（见表 1），因为块数、隐藏维度大小和稀疏率都是固定的。在第 4.3 节中，我们将进一步分析模型稀疏性和模型参数理论复杂度的影响。

与预训练的 DNN 的比较。表 5 报告了 SAINT 基准 [48] 上模型等级的均值和标准差。令人惊讶的是，我们发现简单的纯基于 MLP 的 T-MLP 优于基于 Transformer 的 SAINT 变体（SAINT-s 和 SAINT-i），并且在所有三种任务类型上与 SAINT 相当。值得注意的是，SAINT 及其变体在训练过程的参数上采用了复杂的样本间注意和自监督预训练以及 HPT。此外，T-MLP 集成甚至实现了与调整后的 GBDT（即 XGBoost、CatBoost、LightGBM）相媲美的稳定结果，并在分类任务上超越了预训练的 SAINT。由于未报告详细的 HPT 条件（即迭代时间、HPT 方法、参数采样分布），因此我们不估计具体的训练成本。

与广泛调整的 GBDT 进行比较。表 6 在通常以 GPDT 为主的基准 TabBen [23] 上比较了 T-MLP，在该基准上，GBDT 框架在所有类型的数据集上的表现都完全优于各种 DNN。TabBen 上每个基线的结果都是通过大约 400 次重度 HPT 迭代获得的，几乎代表了无限计算资源和预算下的终极表现。正如预期的那样，当广泛调整的 XGBoost 可用时，单个 T-MLP 黯然失色，但它仍然与其他集成树模型（即 RF、GBT、HistGBT）相媲美，并且优于比较的 DNN。此外，我们发现 T-MLP 集成

表 6：TabBen（四种数据集类型）上所有方法等级的平均值（标准差）。“Num.”和“Cat.”分别表示数值数据集（所有特征都是数值）和分类数据集（一些特征是分类的）。“Classif.”和“Reg.”表示分类和回归任务。“Num. Reg.”组仅包括数值数据集上的回归结果（其他组也使用类似的符号）。||是每组的数据集数量。基线测试结果是基于测试期间的最佳验证结果获得的~HPT 400 次迭代（根据 TabBen 论文和存储库）。详细结果见附录。

数据集类型：	编号。分类。(||=9)	编号。注册。猫。(||=14)	分类。猫。(||=6)	注册。(||=10)
多层感知处理器	8.4（0.8）	不适用	不适用	不适用
残差网络	6.9（1.9）	6.5（1.9）	7.8（1.0）	7.7（0.5）
光纤到户	5.7（1.9）	5.5（2.3）	5.5（2.2）	6.7（1.1）
圣	6.9（1.4）	5.5（2.2）	8.0（1.1）	不适用
国标	4.7（2.0）	4.3（1.7）	5.2（2.3）	4.3（1.1）
历史GBT	不适用	不适用	5.2（2.3）	4.3（1.3）
射频	4.6（2.1）	4.8（2.2）	4.0（3.2）	5.8（1.9）
XGBoost	2.6（1.4）	2.4（1.5）	2.8（1.5）	2.1 (1.0)
肌萎缩性骨髓纤维化	3.2（1.6）	4.3（1.9）	3.5（2.3）	3.6（1.4）
T-MLP(3)	2.1 (1.4)	2.7（1.5）	3.0（1.3）	1.8（0.7）

能够在所有四种数据集类型中与具有相似等级稳定性的最终 XGBoost 相媲美，可作为调整后的 XGBoost 替代方案的候选。更重要的是，在实验中，每个 T-MLP（或 T-MLP 集成）都采用在默认配置下训练的张量化 XGBoost（参见第 4.1 节中的实现细节），并且所有其他超参数都是固定的；因此 T-MLP 及其集成具有通过 HPT 或选择其他 GBDT 作为特征门来提高性能上限的潜在能力。

总之，我们通过实证研究证明了我们的混合框架具有强大的潜力，能够通过各种表格偏好（表格数据偏好高级 DNN、预训练或 GBDT）实现灵活和通用的数据适应性。基于令人印象深刻的经济性能成本权衡和友好的训练过程，T-MLP 可以作为现实世界应用的有前途的表格模型框架，尤其是在有限的计算预算下。

4.3 为什么T-MLP具有成本效益？

表 7 报告了 T-MLP 在几个分类和回归数据集（即加州住房（CA）[40]、成人（AD）[34]、希格斯（HI）[5] 和年（YE）[6]）上在不同数据尺度（括号中给出）的消融和比较实验结果。

主要消融。表 7 中的前四行报告了单个 T-MLP 中两个关键设计的影响。总体观察是，结构稀疏性和 GBDT 特征门 (FG) 都有助于提高 T-MLP 的性能。从数据处理的角度来看，GBDT FG 通过特定于样本的特征选择带来局部稀疏性，而稀疏的 MLP 结构提供所有样本共享的全局稀疏性。有趣的是，我们发现 GBDT FG 对 CA 数据集的影响更为深刻。一个可能的解释是 CA 的特征量（8 个特征）相对较小

表 7：各种任务类型和数据规模的经典表格的主要消融和比较。前 4 行：T-MLP 框架中关键设计的消融。底部 2 行：带有神经网络特征门控 (NN FG) 的 T-MLP 结果。

数据集：	加拿大（21K）↓广告（49K）↑嗨（98K）↑叶 (515K)↓			
肌萎缩性骨髓纤维化	0.4471	0.864	0.729	8.768
无稀疏性	0.4503	0.857	0.726	8.887
无 GBDT FG	0.4539	0.859	0.728	8.799
两者皆无	0.4602	0.856	0.724	8.896
T-MLP (NN FG)	0.4559	0.852	0.718	8.925
无稀疏性	0.4557	0.840	0.713	8.936

与其他数据集（AD、HI 和 YE 中分别为 14、28 和 90 个特征）相比，平均特征重要性可能相对较大，因此 CA 结果更容易受到特征选择的影响。对于特征量较大的数据集，选择有效的特征可能会更加困难。

贪婪的特征选择。我们注意到最近有人尝试使用门控网络来实现生物医学表格的样本特定稀疏性；它最初是为小样本表格设置而设计的，有助于生物医学领域的预测可解释性 [64]。我们使用它的代码并通过用神经网络特征门控（NN FG）替换 GBDT FG 来构建 T-MLP 版本以进行比较。表 7 的下面两行报告了结果。正如预期的那样，在最小的数据集 CA 上，NN FG 可以通过学习选择信息特征来提高性能，但是随着数据规模的增加，这种特征门控策略会持续损害性能。这可能是因为（1）大数据集需要更复杂的结构来学习细致的特征选择，（2）选择行为的离散性质与神经网络的平滑优化模式不相容，以及（3）DNN 的确认偏差 [52] 可能会误导学习过程，即一旦后续神经网络捕获到错误的模式，NN FG 就会得到错误的信息。相比之下，GBDT FG 总是贪婪地选择特征作为真正的 GBDT，这是保守的并且通常是合理的。此外，复杂的子树结构对于选择动作来说更加完整。

稀疏性促进表格 DNN。图 3 绘制了两个分类/回归任务上 T-MLP 稀疏性的性能变化。与 NLP 中旨在减小模型大小同时保持原始模型能力的修剪技术不同，我们发现合适的模型稀疏性通常可以促进表格预测，但过度和不足的稀疏性都无法达到最佳效果。结果从经验上表明，与非结构化数据的大型预训练模型中的 DNN 修剪相比，在表格数据领域，修剪能够促进非大型表格 DNN，因为 GBDT 的有益稀疏结构是通过树预修剪实现的，并且表格 DNN 中的隐藏维度通常过度参数化。

4.4 T-MLP 的优越性解释能力

在图 4 中，我们可视化了 FT-Transformer、XG-Boost 和单个 T-MLP 的决策边界，以检查

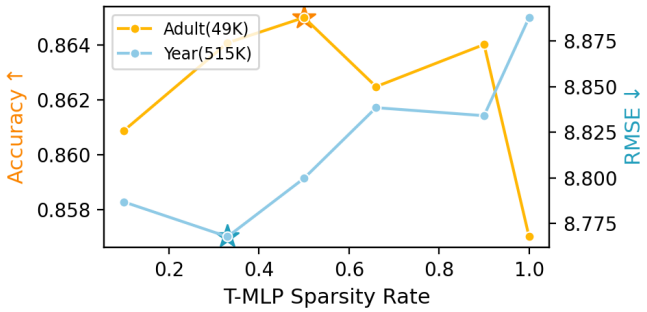

图 3：成人和年数据集上的性能变化图，与 T-MLP 稀疏度的变化有关。所有最佳结果都是通过适当的稀疏度实现的。

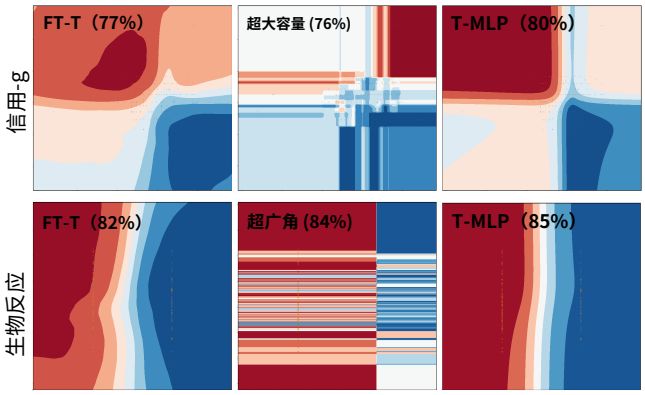

图 4：使用两个最重要的特征对 Bioresponse 和 Credit-g 数据集上的 FT-Transformer (FT-T)、XGBoost 和单块 T-MLP 进行决策边界可视化。不同的颜色代表不同的类别，而不同深浅的颜色表示预测的概率。

这三种方法。通过互信息（使用 Scikit Learn 包估计）选择两个最重要的特征。与常见的 DNN 和 GBDT 不同，T-MLP 展示了一种新颖的中间模式，该模式结合了 DNN 和 GBDT 的特点。与 DNN 相比，T-MLP 产生网格状边界，其边缘通常与 GBDT 的特征表面正交，并且通过修剪稀疏架构本质上简化了复杂性。此外，T-MLP 能够捕获树模型状子模式（参见 Credit-g 上的 T-MLP），而 DNN 仅管理主模式。因此，由于 DNN 的边界相对不规则并且忽略了细粒度 的子模式，因此对过度拟合敏感。与具有锯齿状边界和过度分裂子模式的 GBDT 相比，T-MLP 在边界交叉处保持非常平滑的顶点（参见 Credit-g 上的 T-MLP）。值得注意的是，T-MLP 可以通过平滑的过程决定条件分割点，如 GBDT 特征分割（特征表面的正交边）（参见 Bioresponse 上的 T-MLP 边界边，从上到下，其中水平特征上的分割点相对于垂直特征有条件地平滑变化，而 XG-Boost 很难实现这种动态分割点）。总体而言，T-MLP 兼具抗过拟合的优势，这有助于在 GBDT 和 DNN 青睐的数据集上展现其优势。

5 结论

在本文中，我们提出了 T-MLP，这是一种新颖的混合框架，兼具 GBDT 和 DNN 的优势，可解决表格预测任务中的模型选择难题。我们结合了张量化 GBDT 特征门、DNN 修剪技术和原始反向传播优化器，开发了一种简单但高效且广泛有效的 MLP 模型。在各种基准上进行的实验表明，T-MLP 具有广义适应性，无论特定于数据集的框架偏好如何，都可以在显着降低运行时成本的情况下实现相当有竞争力的结果。我们希望我们的 T-MLP 将成为经济表格预测的实用方法以及广泛的应用，并有助于推动混合表格模型的研究。

致谢

本研究得到国家自然科学基金（编号 62176231）、浙江省重点研发计划（编号 2023C03053 和 2024SSYS0026）的部分资助。

参考

[1] Naomi S Altman。1992 年。核和最近邻非参数回归简介。*美国统计学家*46, 3 (1992), 175-185。
[2] Sercan Ö Arik 和 Tomas Pfister。2021 年。TabNet：专注的可解释表格学习。在*美国航空学会联合会*.6679–6687。
[3] Saqib Aziz、Michael Dowling、Helmi Hammami 和 Anke Piepenbrink。2022 年。金融中的机器学习：一种主题建模方法。*欧洲财务管理*（2022 年）。
[4] Sarkhan Badirli、Xuanqing Liu、Zhengming Xing、Avradeep Bhowmik、Khoa Doan 和 Sathiya S Keerthi。2020 年。梯度增强神经网络：GrowNet。*arXiv 预印本 arXiv:2002.07971*（2020 年）。
[5] Pierre Baldi、Peter Sadowski 等人，2014 年，利用深度学习寻找高能物理中的奇异粒子。*自然通讯*5, 1 (2014), 4308。
[6] Thierry Bertin-Mahieux、Daniel PW Ellis、Brian Whitman 和 Paul Lamere。2011 年。百万首歌曲数据集。*信息与制造研究所*。
[7] Vadim Borisov、Tobias Leemann、Kathrin Seßler、Johannes Haug、Martin Pawelczyk 和 Gjergji Kasneci。2022 年。深度神经网络和表格数据：一项调查。*IEEE 神经网络和学习系统学报*（2022 年）。
[8] Vadim Borisov、Kathrin Sessler、Tobias Leemann、Martin Pawelczyk 和 Gjergji Kasneci。2022. 语言模型是现实的表格数据生成器。在*国际肾病研究联合会*。
[9] Leo Breiman。2001 年。随机森林。*机器学习*45 (2001), 5-32。
[10] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等人。2020 年。语言模型是少样本学习器。在*神经成像与成像系统*，第 33 卷。1877–1901 年。
[11] 曹桂平，罗胜达，黄文建，兰向远，蒋冬梅，王耀伟，张建国。2023.Strip-MLP：Vision MLP 的高效代币交互。在*独立控制计算机视觉*1494–1504年
[12] Jintai Chen、KuanLun Liao、Yanwen Fang、Danny Chen 和 Jian Wu。2022 年。TabCaps：一种用于使用 BoW 路由进行表格数据分类的胶囊神经网络。*国际肾病研究联合会*。
[13] Jintai Chen、Kuanlun Liao、Yao Wan、Danny Z Chen 和 Jian Wu。2022 年。DANets：用于表格数据分类和回归的深度抽象网络。*美国航空学会联合会*。
[14] 陈金泰，严家欢，陈子怡，吴健。2023. ExcelFormer：在表格数据上超越 GBDT 的神经网络。*arXiv 预印本 arXiv:2301.02819*（2023 年）。
[15] Si-An Chen、Chun-Liang Li、Nate Yoder、Sercan O Arik 和 Tomas Pfister。2023 年。TSMixer：用于时间序列预测的全 MLP 架构。*arXiv 预印本 arXiv:2303.06053*（2023 年）。
[16] Tianqi Chen 和 Carlos Guestrin。2016 年。XGBoost：可扩展的树提升系统。*第 22 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集*.785–794。
[17] Paul Covington、Jay Adams 和 Emre Sargin。2016 年。用于 YouTube 推荐的深度神经网络。*第 10 届 ACM 推荐系统会议论文集*。

[18] 杰罗姆・H・弗里德曼。2001 年。贪婪函数逼近：梯度提升机。*统计年鉴*（2001 年）。
[19] Francesco Fusco、Damian Pascual、Peter Staar 和 Diego Antognini。2023 年。pNLP-Mixer：一种高效的全 MLP 语言架构。*第 61 届计算语言学协会年会论文集*.计算语言学协会，53–60。

[20] 皮埃尔・格尔茨、达米恩・恩斯特和路易斯・韦恩克尔。 2006. 极其随机的树。*机器学习*63 (2006), 3-42。
[21] Yury Gorishniy、Ivan Rubachev 和 Artem Babenko。2022 年。关于表格深度学习中数值特征的嵌入。在*神经成像与成像系统*.24991–25004。
[22] Yury Gorishniy、Ivan Rubachev、Valentin Khrulkov 和 Artem Babenko。2021 年。重新审视表格数据的深度学习模型。*神经成像与成像系统*18932至18943年。
[23] Léo Grinsztajn、Edouard Oyallon 和 Gaël Varoquaux。2022 年。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？在*神经成像与成像系统*。
[24] 郭建元，唐叶辉等。2022 年。Hire-MLP：通过分层重排实现的视觉 MLP。在*计算机视觉与图像处理*.826–836。
[25] 何欣然、潘俊峰等，2014 年，预测 Facebook 广告点击量的实践经验。*在线广告数据挖掘国际研讨会论文集*。

[26] 侯璐，黄志奇，尚立峰，蒋鑫，陈晓，刘群。2020 年。DynaBERT：具有自适应宽度和深度的动态 BERT。在*神经成像与成像系统*，第 33 卷。9782–9793。

[27] Jeremy Howard 和 Sylvain Gugger。2020 年。Fastai：用于深度学习的分层 API。 *信息*11，2 (2020)，108。
[28] Xin Huang、Ashish Khetan、Milan Cvitkovic 和 Zohar Karnin。2020 年。TabTransformer：使用上下文嵌入的表格数据建模。*arXiv 预印本 arXiv:2012.06678*（2020 年）。
[29] Arlind Kadra、Marius Lindauer、Frank Hutter 和 Josif Grabocka。2021 年。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*.23928–23941。
[30] Liran Katzir、Gal Elidan 和 Ran El-Yaniv。 2020.Net-DNF：表格数据的有效深度建模。在*国际肾病研究联合会*。
[31]柯国林，孟奇，托马斯・芬利，王泰丰，陈伟，马卫东，叶其伟，刘铁岩。2017.LightGBM：一种高效的梯度提升决策树。在*神经成像与成像系统*。

[32] Jacob Devlin Ming-Wei Chang Kenton 和 Lee Kristina Toutanova。2019 年。BERT：用于语言理解的深度双向 Transformer 预训练。 *抗凝血酶原*.4171–4186。
[33] Günter Klambauer、Thomas Unterthiner、Andreas Mayr 和 Sepp Hochreiter。2017。自归一化神经网络。在*神经成像与成像系统*，第 30 卷。
[34] Ron Kohavi 等人，1996 年。扩大朴素贝叶斯分类器的准确率：一种决策树混合模型。*知识发现*，第 96 卷，202–207。
[35] Bin Li、J Friedman、R Olshen 和 C Stone。1984 年。分类和回归树 (CART)。*生物识别*（1984 年）。
[36] Hanxiao Liu、Zihang Dai、David So 和 Quoc V Le。2021 年。关注 MLP。在*神经成像与成像系统*.9204–9215。
[37] Christos Louizos、Max Welling 和 Diederik P Kingma。2018 年。通过 L_0 正则化学习稀疏神经网络。*国际肾病研究联合会*。
[38] 马新银、方功凡、王新超，2023 年，LLM-Pruner：大型语言模型的结构剪枝，*神经成像与成像系统*。
[39] Tomas Mikolov, Kai Chen 等人，2013 年，向量空间中词语表征的有效估计。*arXiv 预印本 arXiv:1301.3781*（2013年）。
[40] R Kelley Pace 和 Ronald Barry。1997 年。稀疏空间自回归。*统计与概率快报* 33, 3 (1997), 291-297。
[41] F. Pedregosa、G. Varoquaux 等人，2011 年，Scikit-learn：Python 中的机器学习。 *机器学习研究杂志*12 (2011), 2825–2830。
[42] Sergei Popov、Stanislav Morozov 和 Artem Babenko。2019 年。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*。
[43]Liudmila Prokhorenkova、Gleb Gusev、Aleksandr Vorobev、Anna Veronika Dorogush 和 Andrey Gulin。 2018.CatBoost：具有分类特征的无偏提升。在*神经成像与成像系统*。
[44] Alec Radford、Jong Wook Kim 等人，2021 年。通过自然语言监督学习可迁移的视觉模型。*国际激光医学联合会*.8748–8763。
[45] Camilo Ruiz、Hongyu Ren、Kexin Huang 和 Jure Leskovec。2023 年。在以下情况下启用表格深度学习 ≫ 带有辅助知识图谱。*arXiv 预印本 arXiv:2306.04766*（2023 年）。
[46] Sungyong Seo、Jing Huang、Hao Yang 和 Yan Liu。2017 年。具有双重局部和全局注意力的可解释卷积神经网络用于评论评级预测。*第 11 届 ACM 推荐系统会议论文集*.297–305。

[47] Ravid Shwartz-Ziv 和 Amitai Armon。2022 年。表格数据：深度学习并不是你所需要的全部。*信息融合*81 (2022)，84–90。
[48] Gowthami Somepalli、Avi Schwarzschild、M icah Goldblum、C Bayan Bruss 和 Tom Goldstein。2022 年。SAINT：通过行注意和对比预训练改进表格数据的神经网络。*NeurIPS 2022 首届表格表示研讨会*。

[49] 宋卫平，石成策，肖志平，段志坚，徐业文，张明，唐健。 2019. AutoInt：通过自注意力神经网络进行自动特征交互学习。在*信息与通信技术研究所*1161–1170年。

[50] Mingjie Sun、Zhuang Liu、Anna Bair 和 J Zico Kolter。2023 年。一种简单有效的大型语言模型修剪方法。*arXiv 预印本 arXiv:2306.11695*（2023 年）。

[51] 唐传新、赵玉成等。2022 年。稀疏 MLP 用于图像识别：自注意力真的有必要吗？。在*美国航空学会联合会*.2344–2351。
[52] Antti Tarvainen 和 Harri Valpola。2017 年。“刻薄的老师是更好的榜样：加权平均一致性目标可改善半监督深度学习结果”。*神经成像与成像系统*，第 30 卷。
[53] Ilya O Tolstikhin、Neil Houlsby、Alexander Kolesnikov、Lucas Beyer、Xiaohua Zhai、Thomas Unterthiner、Jessica Yung、Andreas Steiner、Daniel Keysers、Jakob Uszkoreit 等。 2021. MLP-Mixer：一种全 MLP 视觉架构。在*神经成像与成像系统*.24261–24272。

[54]屠正中，侯赛因・塔勒比，张涵，杨峰，佩曼・米兰法尔，艾伦・博维克，李银晓。 2022. MAXIM：用于图像处理的多轴 MLP。在 *计算机视觉与图像处理*.5769–5780。
[55] Shahadat Uddin、Arif Khan、Md Ekramul Hossain 和 Mohammad Ali Moni。2019 年。比较不同的监督机器学习算法在疾病预测中的应用。*BMC 医学信息学和决策*（2019年），1-16。
[56] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017 年。注意力就是你所需要的一切。在*神经成像与成像系统*。
[57] Ruoxi Wang、Rakesh Shivanna、Derek Cheng、Sagar Jain、Dong Lin、Lichan Hong 和 Ed Chi。2021 年。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。*万维网*1785年至1797年。
[58] 王子峰和孙继猛。2022 年。TransTab：学习跨表的可迁移表格 Transformer。在*神经成像与成像系统*，第 35 卷。2902–2915。
[59] Ziheng Wang、Jeremy Wohlwend 和 Tao Lei。2020 年。大型语言模型的结构化剪枝。*增强型神经网络LP*.6151–6162。
[60] Mitchell Wortsman、Gabriel Ilharco、Samir Ya Gadre、Rebecca Roelofs、Raphael Gontijo-Lopes、Ari S Morcos、Hongseok Namkoong、Ali Farhadi、Yair Carmon、Simon Kornblith 等人。2022 年。模型汤：对多个微调模型的权重进行平均可提高准确性，而不会增加推理时间。在*国际激光医学联合会*.23965–239 98。

[61] Mengzhou Xia、Zexuan Zhong 和 Danqi Chen。2022 年。结构化剪枝学习紧凑而准确的模型。*访问控制列表*。
[62] Jiahuan Yan、Jintai Chen、Yixuan Wu、Danny Z Chen 和 Jian Wu。2023 年。T2G-Former：将表格特征组织成关系图可促进异构特征交互。在*美国航空学会联合会*。
[63] 严家欢，郑博，徐红霞，朱一恒，陈丹尼，孙继猛，吴健，陈金泰。 2024 年。使预训练的语言模型在表格预测方面发挥出色。在*国际肾病研究联合会*。

[64] Junchen Yang、Ofir Lindenbaum 和 Yuval Kluger。2022 年。用于表格生物医学数据的局部稀疏神经网络。*国际激光医学联合会*PMLR，25123–25153。
[65] 张军和 Vasant Honavar。2003 年。从属性值分类法和部分指定实例中学习。在*国际激光医学联合会*。
[66] 张军，康德康等，2006，从属性值分类法和数据中学习准确、简洁的朴素贝叶斯分类器。*知识与信息系统*（2006 年）。

[67] 张天平，王少文，严水成，李健，刘倩。 2023。生成表预训练增强了表格预测模型的能力。*arXiv 预印本 arXiv:2305.09696*（2023 年）。

[68] 赵鑫，周昆，李俊毅，唐天一，王晓雷，侯玉鹏，敏前，张北辰，张俊杰，董子灿，等。 2023.大型语言模型调查。*arXiv 预印本 arXiv:2303.18223*（2023 年）。
[69] Bingzhao Zhu、Xingjian Shi、Nick Erickson、Mu Li、George Karypis 和 Mahsa Shoaran。2023 年。XTab：表格 Transformers 的交叉表预训练。*国际激光医学联合会*。

基准特征

我们在表 2 中提供了每个基准的详细数据集统计信息。这些基准在数据规模和任务类型上表现出广泛的数据多样性。从 FT-T 基准到 TabBen，整体数据量逐渐减少。我们还在图 2 中可视化了 GBDT 和 DNN 框架各自的胜率，表明在不同表格预测任务中使用的数据集集合中框架的偏好不同。FT-T 在其主要基准中不包含 GBDT 基线，但拥有最多的极大数据集。总体而言，FT-T 基准是极大规模的数据集合（包括数据量和特征宽度），T2G 基准是一个大型基准，SAINT 基准包含多样化的数据规模，而 TabBen 专注于中等规模的典型表格。

B 基线信息

我们在本节列出了所有比较的基线。

- MLP：没有特征交互的 Vanilla 多层感知。

- ResNet：视觉应用中流行的 DNN 主干。
- SNN[33]：一种具有SELU激活功能的类MLP架构。
- GrowNet [4]：以梯度增强方式构建的 MLP。
- NODE[42]: 广义的无意识决策树集成。
- TabNet [2]：一种基于 Transformer 的循环架构，模拟基于树的学习过程。
- AutoInt [49]：基于注意力机制的特征嵌入。
- DCNv2 [57]: 一种基于 MLP 的架构，带有特征交叉模块。

- TabTransformer [28]：连接数值特征和编码分类特征的 Transformer 模型。
- DANets [13]：一种基于 MLP 的架构，每个块都有神经引导的特征选择和抽象。
- FT-Transformer [22]：一种流行的表格Transformer，对数值和分类特征进行编码。
- T2G-Former[62]：具有自动关系图估计功能的表格 Transformer，可用于选择性特征交互。
- SAINT [48]：一种类似 Transformer 的架构，执行行级和列级注意，并进行对比预训练，以最小化数据点和其增强视图之间的差异。

- XGBoost [16]：一种主要的 GBDT 实现。
- CatBoost[43]：一种带有无意识决策树的 GBDT 方法。

- LightGBM [31]: 一种高效的 GBDT 实现。
- 随机森林[9]：一种流行的决策树装袋集成算法。

- ExtraTrees [20]：一种经典的树袋实现。
- k-NN [1]：传统的监督机器学习算法；使用两个 KNeighbors 模型（KNeighborsDist、KNeighborsUnif）。
- NeuralNetFastAI [27]：对表格数据进行操作的 FastAI 神经网络模型。
- sklearn-GBDT [41]：Scikit Learn 包中提供了两种传统的 GBDT 实现（GradientBoostingTree 和 HistGradientBoostingTrees）。

碳 运行时环境和超参数

C.1 运行环境

所有实验均使用 PyTorch 版本 1.11.0、CUDA 版本 11.3 和 Scikit Learn 版本 1.1.0 进行，每次试验均使用 NVIDIA A100 PCIe 40GB 和 Intel Xeon 处理器 40C。

C.2 T-MLP 的超参数

在主要实验中，我们统一设置隐藏大小 到 1024，中间大小 '为 676（隐藏层的 2/3），稀疏率设为 0.33，残差 dropout 率设为 0.1，具有三个基本块，用于多类分类或极大二分类

分类数据集，其他数据集各占一个块。单个 T-MLP 的学习率为 1e-4，三个
T-MLP 集合中的分支分别为 1e-4、5e-4 和 1e-3。

C.3 基线的超参数

对于 FT-T 和 T2G 基准上的所有基线，我们遵循原始基准论文中给出的超参数空间和迭代时间来估算训练成本。

表 8：SAINT 基准中二元分类数据集上基线的 AUC 分数（越高越好）。

OpenML 编号：	31	四十四1017	1111	1487	1494	1590	4134 42178
射频	0.778 0.986 0.798 0.774 0.910 0.928 0.908 0.868 0.840 0.764						
额外的树木	0.986 0.811 0.748 0.921 0.935 0.903 0.856 0.831						
K邻居分布 0.501	0.873 0.722 0.517 0.741 0.868 0.684 0.808 0.755 K邻居统一 0.489						
0.847 0.712 0.516	0.734 0.865 0.669 0.790 0.764 LightGBM						
	0.752 0.989 0.829 0.815 0.919 0.923 0.930 0.860 0.854 0.778						
XGBoost	0.989 0.821 0.818 0.919 0.926 0.931 0.864 0.856 0.788 0.988 0.						
CatBoost	838 0.818 0.917 0.937 0.930 0.862 0.841 0.705 0.980 0.745 0.709						
多层感知处理器	0.913 0.932 0.910 0.818 0.841 0.736 0.979 0.422 0.718 0.625						
塔格网	0.677 0.917 0.701 0.830						
TabTransformer	0.771 0.982 0.729 0.763 0.884 0.913 0.907 0.809 0.841 SAINT-s						
	0.774 0.982 0.781 0.804 0.906 0.933 0.922 0.819 0.858 0.774						
圣伊	0.981 0.759 0.816 0.920 0.934 0.919 0.845 0.854 0.790 0.991 0.						
圣	843 0.808 0.919 0.937 0.921 0.853 0.857						
肌萎缩性骨髓纤维化	0.805 0.983 0.818 0.814 0.924 0.933 0.924 0.853 0.862 0.802						
T-MLP(3)	0.983 0.821 0.816 0.924 0.935 0.925 0.855 0.861						

表 9：SAINT 基准中多类分类数据集上基线的准确度分数（越高越好）。

OpenML 编号：	188	1596	4541 40685 41166 41169 42734			
射频	0.653 0.953 0.607 0.999		0.671 0.358 0.743			
额外的树木	0.653 0.946 0.595 0.999		0.648 0.341 0.736			
K邻居分布 0.442	0.965 0.491 0.997 K邻居统一		0.620	0.205	0.685	
0.422 0.963 0.489	0.997 LightGBM		0.605	0.189	0.693	
	0.667 0.969 0.611 0.999		0.721	0.356	0.754	
XGBoost	0.612 0.928 0.611 0.999		0.707	0.356	0.752	
CatBoost	0.667 0.871 0.604 0.999		0.692	0.376	0.747	
多层感知处理器	0.388 0.915 0.597 0.997		0.707	0.378	0.733	
塔格网	0.259 0.744 0.517 0.997		0.599	0.243	0.630	
TabTransformer	0.660 0.715 0.601 0.999		0.531	0.352	0.744	
SAINT-s	0.680 0.735 0.607 0.999		0.582	0.194	0.755	
圣伊	0.646 0.937 0.598 0.999		0.713	0.380	0.747	
圣	0.680 0.946 0.606 0.999		0.701	0.377	0.752	
肌萎缩性骨髓纤维化	0.660 0.968 0.598 1.000		0.718	0.382	0.747	
T-MLP(3)	0.674 0.970 0.601 1.000		0.728	0.384	0.750	

表 10：SAINT 基准中回归数据集上基线的 RMSE 分数（越低越好）。

OpenML 编号：	422	541	42563	42571	42705	42724	42726	42727	42728	42729
射频	0.027	17.814	37085.577	1999.442	16.729	12375.312	2.476	0.149	13.700	1.767
额外的树木	0.027	19.269	35049.267	1961.928	15.349	12505.090	2.522	0.147	13.578	1.849
K邻居区	0.029	25.054	46331.144	2617.202	14.496	13046.090	2.501	0.167	13.692	2.100
邻居联合	0.029	24.698	47201.343	2629.277	18.397	12857.449	2.592	0.169	13.703	2.109
轻量级GBM	0.027	19.871	32870.697	1898.032	13.018	11639.594	2.451	0.144	13.468	1.958
XGBoost	0.028	13.791	36375.583	1903.027	12.311	11931.233	2.452	0.145	13.480	1.849
CatBoost	0.027	14.060	35187.381	1886.593	11.890	11614.567	2.405	0.142	13.441	1.883
NeuralNetFastAI	0.028	22.756	42751.432	1991.774	15.892	11618.684	2.500	0.162	13.781	3.351
TabNet	0.028	22.731	200802.769	1943.091	11.084	11613.275	2.175	0.183	16.665	2.310
TabTransformer	0.028	21.600	37057.686	1980.696	15.693	11618.356	2.494	SAINT–s	0.162	12.982
	0.027	9.613	193430.703	1937.189	10.034	11580.835	2.145	0.158	12.603	1.833
圣伊	0.028	12.564	1997.111	11.513	11612.084	2.104	0.153	12.534	1.867	33112.387
圣	0.027	11.661	10.282	11577.678	2.113	0.145	12.578	1.882		
肌萎缩性骨髓纤维化	0.027	11.643	21773.233	1946.203	9.027	11828.872	2.041	0.161	13.271	1.843
T-MLP(3)	0.027	13.790	1939.557	8.972	11762.376	2.049	0.161	13.016	1.852	

表 11：TabBen 数值数据集上二元分类任务基线的准确度分数（越高越好）。

	眼睛	MiniBooNE	Higgs	银行市场	覆盖类型	MagicTele.	电力	信用	jannis
残差网络	0.574	0.937	0.694	0.794	0.803	0.858	0.809	0.761	0.746
光纤到户	0.586	0.937	0.706	0.804	0.813	0.851	0.820	0.765	0.766
圣	0.589	0.935	0.707	0.791	0.803	0.851	0.818	0.760	0.773
国标	0.637	0.932	0.711	0.803	0.819	0.851	0.862	0.772	0.770
XGBoost	0.655	0.936	0.714	0.804	0.819	0.859	0.868	0.774	0.778
射频	0.650	0.927	0.708	0.798	0.827	0.853	0.861	0.772	0.773
多层感知处理器	0.569	0.935	0.689	0.792	0.789	0.847	0.810	0.760	0.746
肌萎缩性骨髓纤维化	0.610	0.946	0.731	0.802	0.909	0.859	0.842	0.772	0.800
T-MLP(3)	0.613	0.947	0.733	0.803	0.915	0.861	0.848	0.775	0.799

表 12：TabBen 数值数据集上回归任务基线的 R 平方分数（越高越好）。

	电梯	自行车	房屋	nyc-taxi	pol		硫磺	副翼	葡萄酒	超级	房子	销售	巴西	迈阿密	cpu	行为	钻石	
残差网络	0.910	0.669	0.821	0.468	0.948	0.819	0.835	0.363	0.895	0.868	0.998	0.914	0.982	0.942				
光纤到户	0.914	0.671	0.832	0.476	0.995	0.838	0.844	0.359	0.885	0.875	0.998	0.919	0.978	0.944				
圣	0.923	0.684	0.820	0.496	0.996	0.784	0.374	0.894	0.806	0.879	0.994	0.921	0.984	0.944				
国标	0.863	0.690	0.840	0.554	0.979	0.843	0.458	0.905	0.865	0.884	0.995	0.924	0.985	0.945				
XGBoost	0.908	0.695	0.852	0.553	0.990	0.844	0.498	0.911	0. 859	0.887	0.998	0.936	0.986	0.946				
射频	0.841	0.687	0.829	0.563	0.989	0.839	0.504	0.909		0.871	0.993	0.924	0.983	0.945				
肌萎缩性骨髓纤维化	0.875	0.694	0.834	0.560	0.995	0.853	0.840	0.410	0.894	0.886	0.993	0.939	0.982	0.949				
T-MLP(3)	0.908	0.698	0.838	0.566	0.996	0.860	0.843	0.416	0.899	0.888	0.995	0.939	0.983	0.950				

表 13：TabBen 分类数据集上二元分类任务基线的准确度分数（越高越好）。

	眼睛	道路安全	电	封面类型	韦尔	罗盘
光纤到户	0.598	0.767	0.842	0.867	0.703	0.753
残差网络	0.579	0.761	0.826	0.853	0.706	0.745
圣	0.585	0.764	0.834	0.850	0.682	0.719
国标	0.639	0.762	0.880	0.856	0.776	0.741
XGBoost	0.668	0.767	0.887	0.864	0.770	0.769
历史GBT	0.636	0.765	0.882	0.845	0.761	0.751
射频	0.657	0.759	0.878	0.859	0.798	0.793
肌萎缩性骨髓纤维化	0.605	0.786	0.880	0.882	0.757	0.785
T-MLP(3)	0.609	0.786	0.881	0.880	0.762	0.790

表 14：TabBen 分类数据集上的回归任务基线的 R 平方分数（越高越好）。

	自行车颗粒巴西钻石黑色纽约出租车 analcatdata OnlineNews 奔驰房屋销售									
光纤到户	0.937	0.673	0.883	0.990	0.379	0.511	0.977	0.143	0.548	0.891
残差网络	0.936	0.658	0.878	0.989	0.360	0.451	0.978	0.130	0.545	0.881
国标	0.942	0.683	0.995	0.990	0.615	0.573	0.981	0.153	0.578	0.891
XGBoost	0.946	0.691	0.998	0.991	0.619	0.578	0.983	0.162	0.578	0.896
历史GBT	0.942	0.690	0.993	0.991	0.616	0.539	0.982	0.156	0.576	0.890
射频	0.938	0.674	0.993	0.988	0.609	0.585	0.981	0.149	0.575	0.875
肌萎缩性骨髓纤维化	0.938	0.692	0.996	0.990	0.620	0.571	0.990	0.154	0.576	0.893
T-MLP(3)	0.942	0.698	0.996	0.993	0.622	0.580	0.990	0.158	0.578	0.894