将分级作为借口任务：改进表格领域的自监督学习

李敬恩[1]艺瑟心[1]赵惠承[1]文正宇[1]尹秀熙[1]尹尚佑[1]
林宇亨[1]

抽象的

深度网络学习优质表征的能力取决于利用适当的归纳偏差，同时考虑到数据集的固有属性。在表格领域，以统一的方式有效处理异构特征（分类和数值）并掌握分段常数函数等不规则函数至关重要。为了应对自监督学习框架中的挑战，我们提出了一种基于经典的借口任务的新方法*分箱*方法。这个想法很简单：*重建 bin 索引*（顺序或类别）而不是原始值。此借口任务为编码器提供了归纳偏差以捕获不规则依赖关系，从连续输入映射到离散化箱，并通过将所有特征设置为具有类别类型目标来减轻特征异质性。我们的实证研究确定了分箱的几个优点：捕获不规则函数、与编码器架构和其他修改的兼容性、将所有特征标准化为相等的集合、将特征内的相似值分组以及提供排序信息。对各种表格数据集的全面评估证实，我们的方法可以持续提高各种下游任务的表格表示学习性能。代码可在https://github.com/kyungeun-lee/tabularbinning。

1. 简介

表格数据集在各种应用中无处不在，从金融市场和医疗诊断到电子商务个性化和制造流程自动化。这些数据集的结构为行代表单个样本，列代表异构特征（分类特征和数值特征的组合），它们是无数分析的基础。尽管表格数据适用范围广泛，但利用深度网络来利用此类数据集固有属性的研究仍处于起步阶段。相比之下，基于树的机器学习算法（如 XGBoost（Chen & Guestrin，2016 年）和 CatBoost（Prokhorenkova 等人，2018 年））在辨别表格域细微差别方面始终表现出色，甚至比具有更大模型容量和专门模块的深度网络表现更好（Arik & Pfister，2021 年；Gorishniy 等人，2021 年；Grinsztajn 等人，2022 年；Rubachev 等人，2022 年）。树模型所拥有的一致优势推动了人们探索如何将其有利偏差应用于深度网络。

最近，提高深度网络在表格数据上的性能的追求势头强劲。一个根本的挑战是表格数据集固有的异质性，包括分类和数值特征（Popov 等人，2019 年；Borisov 等人，2022 年；Yan 等人，2023 年）。为了缓解深度网络中的特征差异，先前的研究提出使用附加模块，如特征标记器（Gorishniy 等人，2021 年）和抽象层（Chen 等人，2022 年）。同时，一些研究探索了将基于树的模型的已证实优势注入深度网络的方法。例如，Grinsztajn 等人（2022 年）观察到，与基于树的模型相比，深度网络倾向于过于平滑的解决方案，并且难以对分段常数函数等不规则性进行建模。为了应对这一挑战，Gorishniy 等人。 （2022）引入了一种新方法，该方法结合了预处理过程中的分段线性编码和周期性激活函数。尽管这些进步已经提高了几个表格数据问题的性能，但它们主要在监督学习中进行探索。

[1]LG AI Research，韩国首尔。联系人：Woohyung Lim <w.lim@lgresearch.ai>。

会议纪要41英石国际机器学习会议，奥地利维也纳。PMLR 235，2024 年。

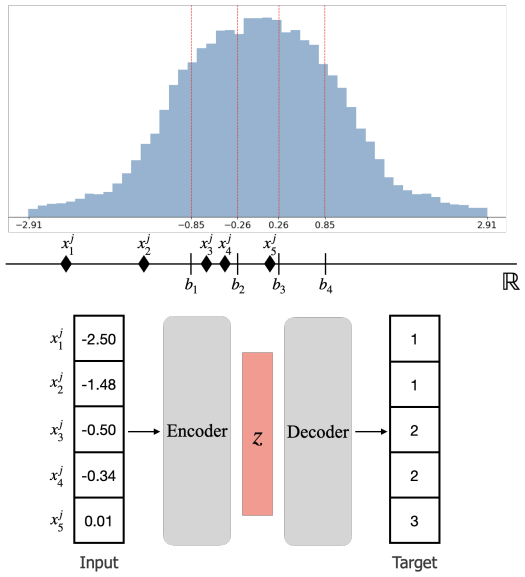


图 1：分箱作为借口任务。根据每个特征的训练数据集分布确定分箱。输入被传递到编码器网络，然后解码器网络预测分箱索引，当借口任务是回归时，分箱索引可以是序数，当借口任务是分类时，分箱索引可以是名义上的。

学习框架，但它们仍然无法超越简单的基于树的方法（Gorishniy 等人，2021 年；Grinsztajn 等人，2022 年；McElfresh 等人，2023 年）。

在本研究中，我们解决了*无监督*表格深度学习，其中基于树的方法根本不适用。为此，我们提出了一种基于经典的借口任务*分箱*基于自动编码的自监督学习（SSL）方法。我们的方法很简单：*重建 bin 索引*而不是重建原始值，如图 1 所示。一旦基于训练数据集的分位数将数值特征离散化为箱体，我们就会优化编码器和解码器网络，以根据原始输入准确预测箱体索引。尽管简单，但将箱体作为借口任务为表格深度学习提供了几个优势。通过将离散化的箱体设置为借口任务的目标，我们可以采用捕获不规则函数和减轻特征之间差异的归纳偏差。箱体过程允许根据训练数据集的分布对附近的样本进行分组，因此学习到的表示应该对可能产生虚假模式的小错误具有鲁棒性。它还有助于将所有特征标准化为相等的集合，从而防止任何无信息特征在 SSL 期间占主导地位。此外，我们的方法与任何其他修改兼容，包括深度架构和输入转换函数的选择。

基于对 25 个公开数据集的大量实验，

我们发现，即使我们只是将 SSL 期间的目标从连续箱改为离散箱，分箱任务也能持续提高 SSL 在各种下游任务上的性能。最后，我们发现，分箱任务不仅可以成为完全无监督学习的有效目标函数，而且可以作为实现最佳性能的预训练策略，在广泛的表格数据问题中超越基于树的和其他监督的深度学习方法。

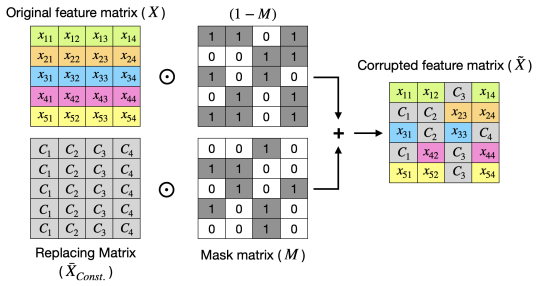
我们的主要贡献可以总结如下。首先，我们建议将分箱作为表格域中 SSL 的新前置任务，并与任何修改兼容。其次，我们对 25 个公共表格数据集进行了广泛的实验，重点关注各种输入转换方法和 SSL 目标。最后，我们在无监督和监督学习框架中始终如一地实现了最佳性能。代码可在https://github.com/kyungeun-lee/tabularbinning。

2.相关工作

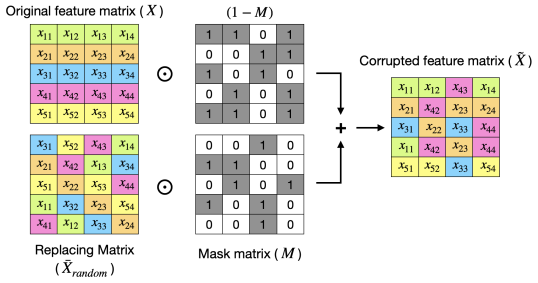
表格深度学习： 近年来，
大量关于表格领域的深度学习研究：开发新的深度架构 (Popov et al., 2019; Badirli et al., 2020; Huang et al., 2020; Wang et al., 2021; Arik & Pfister, 2021; Gorishniy et al., 2021; Chen et al., 2022; Hollmann et al., 2022; Zhu et al., 2023; Kotelnikov et al., 2023; Chen et al., 2023a)；或将表格特征的异构性质表示为图 (Yan et al., 2023; Chen et al., 2023b)；或采用新的激活函数 (Gorishniy et al., 2022)。尽管取得了这些进步，决策树集合（例如 GBDT（梯度提升决策树））仍然充当着竞争基线（Arik & Pfister，2021；Gorishniy 等人，2021；Grinsztajn 等人，2022；Rubachev 等人，2022；McElfresh 等人，2023；Beyazit 等人，2023）。在本文中，我们的目标是为表格领域的自监督学习提出一种新的借口任务，因此除了最先进的表格深度学习模型（例如 T2G-Former，2023）外，我们还专注于直接受经典深度模型启发的架构，特别是 MLP 和 FT-Transformers（Gorishniy 等人，2021）。

表格域中的自监督学习： 监督学习 (SSL) 旨在学习理想的-
的表示而不使用注释信息。最近，对比学习和自动编码已成为表格领域的两种主要选择。对比学习旨在对来自同一样本的两个或多个增强视图之间的相似性（对应于正样本）和其他样本之间的不相似性（对应于负样本）进行建模。Bahri 等人

（2021）；Ucar 等人（2021）在基于数据增强函数（例如在特征维度上进行遮罩或裁剪）定义正样本和负样本后优化了对比损失。自动编码旨在根据损坏的观察结果重建原始样本（Vincent 等人，2008）。与对比学习相比，自动编码器可以处理多种数据类型，这对于涉及异构数据集（如表格数据）的任务非常有用。Yoon 等人（2020）；Huang 等人（2020）；Majmundar 等人（2022）采用了自动编码方法优化重建损失，无论是否有额外的损失，例如损坏检测。在本研究中，我们提出了一种基于自动编码方法的新型 SSL 借口任务。



（a）替换值 = 常数



（b）替换值 = 随机

图 2：生成掩蔽特征的替换向量的两种方法的说明。

3.背景

在本节中，我们将深入研究表格领域中基于自动编码的自监督学习框架，重点关注两个因素：表格输入的转换方法和基于自动编码的 SSL 框架中的目标函数。

输入变换：　　确保编码器网络
不仅仅是学习一个恒等函数，我们在输入上使用转换函数来保留与标签相关的信息。对于表格数据集，只有少数转换函数被提出，如掩码（Yoon 等人，2020 年；Ucar 等人，2021 年；Majmundar 等人，2022 年），如图 2 所示，因为所有单个值都可以在确定语义方面发挥关键作用，而微小的变化可能会影响上下文。给定一个样本 $X_{我} \in \mathrm{R}_{d}$在数据集中*德*在哪里*d*是特征的数量，*我* ∈[1, *否*]，和*否*是批量大小，我们随机生成掩码向量*米我*大小相同$X_{我}$. 掩蔽向量的每个元素*米我*以概率从伯努利分布中独立抽样*页米* ∈ [0，1].要替换屏蔽值，替换向量 $X_{我}$应该定义。在本研究中，我们利用了先前研究中提出的两种方法（Yoon 等人，2020 年；Ucar 等人，2021 年；Majmundar 等人，2022 年）。

- 常数（图2a）：$X_{我知道}$被设置为所有*我*在本研究中，我们使用每个特征的平均值*钾*在训练数据集中。
- 随机（图 2b）：$X_{我知道}$是从给定特征的其他批内样本中采样的。换句话说，为了替换*钾*第个特征*我*批次中的第个样本，我们使用*钾*第个特征*我*同一批（批次）中的第 - 个样品，以及*我*从均匀分布中抽样*乌*1 *否* 。

最后，损坏的样本$X_{我}$公式为$X_{我}$= (1 − *米我*) ⊙$X_{我}$+*米我*⊙$\bar{X}_{我}$在哪里1是大小相同的全 1 向量$X_{我}$。转换过程是随机的，它在训练过程中提供了随机性。当*页米*=0，*米我*成为零矩阵，未损坏的输入$X_{我}$= $X_{我}$用于训练。

SSL 目标：　　按照 SSL 的惯例，
编码器$F_{埃}$首先将损坏的样本$X_{我}$代表*是我*，然后解码器F_{d}将被引入，通过优化无监督损失来学习信息表征*大号*。我们可以通过引入特定的借口任务来决定应该学习哪种表示。作为基准，我们考虑了 Yoon 等人（2020 年）；Huang 等人（2020 年）；Majmundar 等人（2022 年）中使用的两个借口任务。

- 重建原始值：一种常见的方法是从损坏的样本中重建未损坏的样本（Vincent 等人，2008 年）。在此设置中，编码器尝试利用非屏蔽特征中存在的相关性来估算屏蔽特征。学习到的表示将涉及不受损坏影响的语义级信息。为此，解码器网络定义为

作为$g_{侦察}$ ：*是*→ *X*，相应的损失公式为
迟至*大号*价值侦察：=1 $\sum_{我=1}^{否} \|X_{我}-F_{侦察}{}_{d}(是我)\|_{2}{}^{2}$。

- 检测被掩盖的特征：可以促进重建的借口任务的辅助任务是预测
在输入样本的损坏过程中，哪些特征被掩盖了（Yoon 等人，2020 年）。在此设置中，编码器尝试利用特征值之间的不一致性来识别被掩盖的特征，从而产生学习到的表示，以捕获给定输入的异常模式。具体来说，该方法

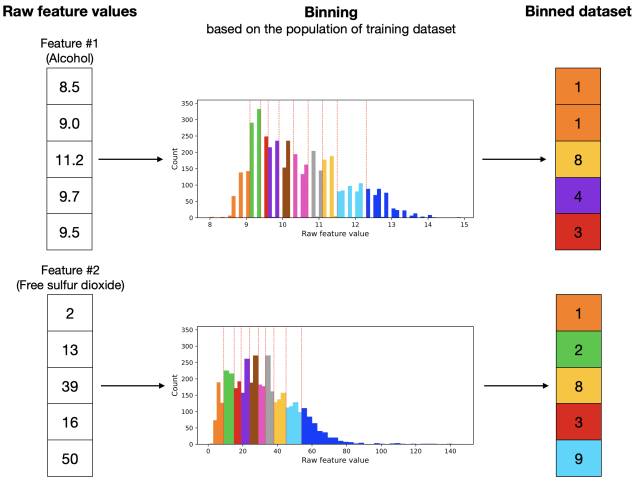


图 3：分箱示例（数据集：葡萄酒质量 (Cortez 等，2009)）。在示例中，我们设置*电视*为 10。对于每个特征，我们根据训练数据集实现分箱以包含相同数量的观察值。最后，我们将分箱索引用作基于自动编码的 SSL 的目标。当我们将分箱索引视为没有顺序信息的类时，分箱索引将转换为独热向量。

采用二元交叉熵 Σ 损失，可以表示为*大号*MaskXent：=−1 $\frac{1}{\text{否}}\sum_{\text{我}=1}^{\text{否}}$*米我*日志$F_{\text{面具}}$（*是我*）+（1）−*米我*）日志 $_d$(1−$F_{\text{面具}}$（*是我*））解码器网络定义为$F_{\text{面具}}$：$\mathbb{R}^d \to M$。

如果我们训练多个解码器，利用这些解码器，我们可以同时优化多个损失函数*是*作为输入。例如，Yoon 等人（2020 年）利用*大号*价值侦察和*大号*MaskXent。

4. 方法：将分箱作为表格 SSL 的前置任务

分箱是一种经典的数据预处理技术，可以量化给定的数值特征$X_{\text{杰}} \in \mathbb{R}$|*D*|进入*电视*离散区间，称为*垃圾箱*$Z_{\text{杰}}$ *吨*= [$b_{\text{杰},t-1}$, $b_{\text{杰},\text{吨}}$）在哪里*吨*∈[1, *吨*]和$b_{\text{杰}\ \text{吨}} \in \mathbb{R}$是分箱边界。分箱可以有效地将连续特征转换为离散特征，减轻数据集中的噪声和异常值等细微错误，并使数据分布更易于管理（Dougherty 等人，1995 年；Han 等人，2022 年）。

在本研究中，我们实施了分箱，以建立基于自动编码的 SSL 的目标。我们预计这些表示将对相同箱中的微小输入变化具有鲁棒性。此外，深度网络可以捕捉类似于基于树的模型的决策过程的不规则性，该模型为每个连续样本分配离散叶子，因为借口任务对应于将连续样本映射到输入到离散化箱中。此外，通过在 SSL 期间将所有特征的目标视为同一类别类型，分箱方法有助于缓解特征异质性。

Müller 等人 (2021) 在贝叶斯推理的背景下提出了一种类似的方法，解决了神经网络在建模连续分布时面临的众所周知的挑战。为了解决这个问题，他们转向利用离散化的连续分布来准确建模后验概率分布。他们的研究表明，将离散化整合到深度网络的目标函数中不仅可以提高训练过程的有效性，而且在理论上也是一种能够建模任何分布的多功能技术。这种方法凸显了分箱在增强神经网络能力方面的巨大潜力。

图 3 描述了分箱过程。我们首先确定箱数*电视*作为设计参数。然后，我们{拆分值}范围到不相交的集合中*电视*间隔，*乙杰* $_{1}$，...，*乙杰电视*，考虑到 ob-训练数据集中的服务*德*$_{\text{火车}}$对于每个*杰*第特征$X_{\text{杰}}$。具体来说，bin 边界$b_{\text{杰}}$ *吨*取决于根据分位数*吨* *电视*。（替代分级策略 gies 也在补充 D.1 中讨论。）当$X_{\text{杰}}$在训练数据集中小于*电视*，每个不同的值被分配有自己的 bin。Fi-

最后，我们将每个数值特征$X_{\text{杰}}$ *我*扔进垃圾桶$Z_{\text{杰}\ \text{吨}}$，我们用相应的值替换原始值分类索引*吨杰* *我*∈[1, *吨*因此，我们使用分组排名（或类别）而不是原始值。我们称分箱数据集为$X_{\text{垃圾桶}}$。

箱索引*我*第个样本和*杰*第个特征，*吨杰* *我*，可表示为序数值或名义类别。当我们利用 bin 索引作为序数值时，我们将借口任务设置为根据连续输入重建 bin 索引，以及相应的*BinRecon 丢失*定义为

$$\text{大号}_{\text{宾雷康}}: = \frac{1}{\text{否}}\sum_{\text{我}=1}^{\text{否}} \left\| \text{吨}_{\text{我}} - F_{d}^{\text{双}_{\text{侦察兵}}}(\text{是}_{\text{我}}) \right\|_2^2$$

在哪里$F_{\text{宾雷康}}$ ：$\mathbb{R}^d \to X_{\text{垃圾桶}}$。

当我们利用箱索引作为名义类别时，我们转换 bin 索引*吨杰* *我*进入独热向量你*杰* *我*= [*你*$_1$，*你*$_2$，...，*你*$_{\text{电视}}$]在哪里*你*$_{\text{五}}$=1什么时候*五*=*吨杰* *我*和*你*$_{\text{五}}$=0否则。然后，我们将借口任务设置为通过优化*BinXent 损失*，定义为每个特征的多类交叉熵损失。

$$\text{大号}_{\text{宾森特}}: = -\frac{1}{\text{否}\,\text{致}}\sum_{\text{我}=1}^{\text{否}}\sum_{\text{杰}=1}^{\text{致}} \text{你}_{\text{我杰}} \log \text{恐}_{\text{宾森特}}(\text{是}_{\text{杰}\ \text{我}})$$

在这种情况下，每个样本的预测应该是 R$d \times$电视。作为一个简单的实现，我们在F宾森特（ • ）：$是 \rightarrow \hat{U}$在哪里 $乌 \in$ R否$\times d \times$电视表示独热编码的分箱数据集。

我们概述了在 SSL 中使用分箱任务的好处。后续章节将提供有关每项如何有利于表格数据问题的经验证据。

- *捕获不规则函数*：我们明确地让深度网络学习在 SSL 期间从连续输入映射到离散目标的函数。它有效地为表格学习提供了有益的归纳偏差，并减轻了异构特征之间的差异。（第 5.2、6.3、6.4 节）
- *与其他修改的兼容性*：分箱任务与编码器架构、输入转换函数和其他目标的变化等修改无关。因此，它可以独立使用，也可以与其他选项结合使用。（第 5.1、5.2 节）
- *将所有特征标准化为相等的集合*[1]：分箱后，所有特征都处于均匀分布中，元素相同。与传统的归一化方案不同，它大大简化了数据集，仅包含*电视*不同的值，这确保所有特征成为相等的集合，从而防止任何不重要的特征在训练过程中占主导地位。（第 6.1 节）
- *将每个特征中的相似值分组*：分箱将每个特征中的邻近值聚类，并消除除箱索引之外的其他信息。深度网络可以将分布中的邻近样本识别为相似的，而不管其大小。（第 6.1 节）
- *BinRecon 中的排序丢失*：BinRecon 损失仅利用分组排序信息，同时消除原始值信息。这确保编码器网络学习排序信息，而不管值的大小如何。（第 6.1 节）

总体而言，我们按如下方式实现 SSL。首先，表格输入经过转换以保留其语义信息。然后，编码器网络F埃接受转换后的输入x并产生表示$是$以及解码器网络F_d模型表示$是$到达目标$\hat{y}$SSL取决于借口任务的选择。在本研究中，我们考虑了四种类型的借口任务，相应的损失分别为 ValueRecon、MaskXent、BinRecon 和 BinXent。一旦 SSL 完成，学习到的表示$是$是基于线性探测来评估的。

[1]详细描述请参阅补充材料 E.1。

5.实验

在本节中，我们将在 25 个公共表格数据集中评估分箱作为前置任务的有效性，这些数据集涵盖了各种数据大小和任务类型。数据集详细信息见补充 A。对于所有数据集，我们对数值特征和标签应用标准化来评估回归任务。

对于编码器网络F埃，我们实验了三种类型的深度网络：（1）MLP，代表最简单的深度架构形式；（2）FT-Transformer（Gorishniy 等人，2021），Transformer 架构针对表格域的简单改编；（3）T2G-Former（Yan 等人，2023），用于表格数据问题的最先进的深度架构。请注意，更大或更复杂的网络并不能保证在表格数据集中获得更好的性能（Gorishniy 等人，2021；Rubachev 等人，2022；Grinsztajn 等人，2022；Gorishniy 等人，2022；McElfresh 等人，2023）。为了确定F埃对于 MLP，我们根据监督设置中的验证性能确定最佳配置，即，只有具有线性头的编码器使用监督损失进行训练，从而确保我们框架的无监督性质。对于 FT-Transformer 和 T2G-Former，我们使用原始论文中的默认设置。对于解码器F_d，我们总是采用与 MLP 型编码器网络相同的 MLP 架构。因此，每个数据集的所有案例都在相同的架构和优化设置上进行了训练。补充 B 中提供了详细描述。对于给定的网络和数据集，我们还研究了掩蔽概率$页米 \in \{0。1，0。2，0。3，0。4，0。5，0。6，0。7，0。8，0。9\}$以及垃圾箱的数量$电视 \in \{2，5，10，20，50，100\}$。然后，我们根据每个下游任务的验证性能找到了最佳配置。在 SSL 之后，我们使用不同的随机种子对基于线性探测的表示进行 10 次评估，并报告平均值。我们根据分类任务的准确度和回归任务的 RMSE 来评估表示质量。带有标准偏差的完整结果也可在补充 C 中找到。所有实验均在单个 NVIDIA GeForce RTX 3090 上进行。代码可在https://github.com/kyungeun-lee/tabularbinning。

5.1 与无监督方法的比较：线性评估结果

我们首先比较了针对每个数据集使用相同 MLP 编码器的一系列 SSL 方法。为了确定分箱任务与其他转换函数的兼容性，我们包括使用掩码优化 BinRecon 损失的案例。最后，我们用四种情况进行实验以验证我们的方法；优化 BinXent，将 bins 视为名义类；优化 BinRecon，将 bins 视为普通类。

表 1：当编码器网络固定为 MLP 时，各种 SSL 方法的线性评估结果。对于每种方法，我们还确定了每个数据集的性能排名，最后一列还提供了平均排名。每个数据集的最佳案例都标记为大胆的。

（a）二元分类（指标：准确率）

掩蔽	替换值	SSL 目标	中 你好 广告 马来亚 肺动脉高压 操作系统CS 邮局	平均排名
错误的	-	价值侦察	0.810 0.651 0.837 0.899 0.728 0.883 0.709 0.851 0.807 0.672	7.571
真的	宪法。	MaskXent	0.836 0.899 0.715 0.893 0.708 0.845 0.810 0.653 0.839 0.900 0.	7.286
真的	宪法。	价值侦察	734 0.884 0.718 0.849	6.429
真的	宪法。	掩码Xent+值调整	0.817 0.669 0.835 0.900 0.724 0.877 0.706 0.837 掩码Xent	7.714
真的	随机的		0.814 0.681 0.8430.9010.710 0.883 0.706 0.853 0.811 0.661	5.429
真的	随机的	价值侦察	0.838 0.898 0.736 0.885 0.714 0.842	7.143
真的	随机的	MaskXent+ValueRecon	0.804 0.647 0.826 0.899 0.715 0.879 0.713 0.861	8.571
错误的	-	宾森特	0.817 0.683 0.8450.9010.732 0.8860.7380.851 0.823 0.6870.840	3.571
错误的	-	宾雷康	0.9000.7370.889 0.7240.865 0.820 0.672 0.843 0.899 0.7300.896	2.286
真的	宪法。	宾雷康	0.718 0.858 0.819 0.6820.8460.898 0.735 0.894 0.718 0.858	3.714
真的	随机的	宾雷康		3.571

（b）多类分类（指标：准确率）

掩蔽	替换值	SSL 目标	一氧化碳加时赛 通用电气 画外音 水质 艾尔 他	MNIST	MNIST	平均排名
错误的	-	价值侦察	0.769 0.776 0.527 0.619 0.568 0.931 0.353 0.784 0.777	0.965	0.928	6.333
真的	宪法。	MaskXent	0.518 0.545 0.547 0.909 0.341 0.783 0.791 0.557 0.622	0.793	0.554	9.333
真的	宪法。	价值侦察	0.586 0.931 0.第354章	0.966	0.925	4.111
真的	宪法。	MaskXent+ValueRecon	0.750 0.774 0.519 0.610 0.571 0.931 0.360 MaskXent	0.941	0.907	7.444
真的	随机的		0.763 0.791 0.555 0.549 0.544 0.925 0.336 0.761 0.782	0.945	0.817	8.000
真的	随机的	价值侦察	0.538 0.625 0.573 0.930 0.357	0.956	0.934	5.556
真的	随机的	MaskXent+ValueRecon	0.769 0.779 0.521 0.564 0.519 0.925 0.353	0.945	0.906	8.333
错误的	-	宾森特	0.742 0.781 0.517 0.600 0.565 0.903 0.354 0.784 0.783	0.956	0.908	8.333
错误的	-	宾雷康	0.544 0.6250.5920.935 0.357 0.812 0.792 0.559 0.647	0.964	0.950	3.556
真的	宪法。	宾雷康	0.581 0.943 0.359 0.814 0.794 0.580 0.6550.5740.949	0.974	0.964	2.222
真的	随机的	宾雷康	0.365	0.981	0.971	1.333

（c）回归（指标：RMSE）

掩蔽	替换值	SSL 目标	加州 何 财务信息 MI 碘化钾中央处理器 迪亚 发光	平均排名
错误的	-	价值侦察	0.749 4.241 13900.720 0.784 0.163 3.876 1016.641 0.399 0.709 4.548	8.625
真的	宪法。	MaskXent	13473.750 0.788 0.185 4.475 1259.744 0.396 0.693 4.086 13518.683	8.875
真的	宪法。	价值侦察	0.778 0.160 3.728 952.444 0.394	5.000
真的	宪法。	MaskXent+ValueRecon	0.700 4.157 13915.875 0.775 0.174 5.644 2797.034 0.398 MaskXent	8.750
真的	随机的		0.677 4.297 13826.641 0.782 0.176 3.951 1358.135 0.388 0.713 4.127	7.875
真的	随机的	价值侦察	13668.988 0.777 0.162 3.760 986.306 0.396	6.500
真的	随机的	MaskXent+ValueRecon	0.701 4.136 14107.645 0.780 0.166 4.506 1917.875 0.397	8.750
错误的	-	宾森特	0.690 4.11613038.7620.776 0.170 3.717 1207.923 0.383 0.622 3.766	4.875
错误的	-	宾雷康	13453.3090.767 0.1583.208 0.634 3.765 13208.133 0.7780716453.0376	2.250
真的	宪法。	宾雷康	0.619 3.70313075.474 0.773 0.160 3.183 957.801 0.371	2.375
真的	随机的	宾雷康	870.283 0.368	1.625

最终值，没有任何增强；优化 BinRecon 时使用掩码作为常量值；优化 BinRecon 时使用掩码作为随机值。在表 1 中，底部的四行对应于我们的方法。

二元分类： 首先，我们比较一下表 1a 中下游任务为二分类的八个数据集的样本。有趣的是，当我们将重构损失的目标从原始值（ValueRecon）更改为 bin 索引（BinRecon）时，我们发现了一致的改进，而其他训练细节保持不变。这些结果表明，学习不规则函数

在表格表示学习中，学习离散函数（从连续到离散）比学习平滑函数（从连续到连续）更有益。

多类分类： 接下来我们调查九个表 1b 中下游任务为多类分类的数据集。与二分类任务不同，我们观察到优化 BinRecon 损失时使用掩码与不使用掩码的情况相比始终会带来额外的改进，而优化 BinXent 效果不佳。这些结果表明顺序信息对于多类分类很重要，BinRecon 可以有效地操纵它们。进一步的讨论将在

表 2：与基于树和深度学习方法（包括最新模型）的比较。对于基线，我们直接参考论文中的性能值，以尽量减少选择超参数时的歧义。当性能不可用时，我们将其留空（-）。对于每个数据集，深度学习方法中的最佳案例都标记为大胆的，第二好的结果是下划线。SSL+Fine-tuning 方法参考了第 5.1 节中研究的基线 SSL 方法的微调结果。对于对应于底部四行的 SSL+Fine-tuning 方法，我们在各种输入变换（None、Masking 为常数、Masking 为随机）和编码器网络（MLP、FT-Transformer、T2G-Former）的组合中提供了最佳结果。培训细节和完整结果在补充材料 C 中提供。

训练网络及方法	二元分类					多类分类							回归		
	你好 ↑	肺动脉高压 ↑	操作系统 ↑	邮局 ↑		一氧化碳 ↑	通用电气 ↑	画外音	艾尔 ↑	他 ↑	MNIST ↑	加州 ↓	何 ↓	财务信息 ↓
基于树的机器学习算法 XGBoost														
	0.726	0.721	0.840	0.711	0.969	0.683	0.699	0.924	0.348	0.727	0.728	0.977	0.434	3.152	10372.778
CatBoost	0.833	0.897	0.967	0.692	0.711	0.948	0.386					0.979	0.430	3.093	10636.322
深度学习方法															
多层感知处理器	0.714	0.724	0.896	0.901	0.968	0.659	0.692	0.960	0.378	0.688	0.728	0.983	0.513	3.146	10086.080
残差网络	0.885	0.795	0.729	0.484	0.550	0.220	0.229					0.826	0.706	4.004	10226.508
TabNet（Arik & Pfister，2021；Gorishniy 等人，2021）	0.719	-	-	-		0.957	0.587	0.568	0.954	0.378	0.958	- 0.968	0.510	-	-
NODE（Popov 等人，2019；Gorishniy 等人，2021）	0.726	DCN V2	-	-	-				-	0.918	0.359	-	0.464	-	-
（Wang 等人，2021；Gorishniy 等人，2021）	0.723	-	-	-		0.965-			-	0.955	0.385	-	0.484	-	-
SCARF（Bahri 等人，2021）	SAINT（Somepalli 等	0.585	0.710	0.878	0.838	0.654	0.325	0.289	0.731	0.050	0.713	0.728	0.801	1.084	5.595	13632.255
人，2021）	FT-Transformer（Gorishniy 等人，	0.886	0.877	0.943	0.691	0.713	0.932	0.378	0.729	0.724	0. 882	0.890	0.981	0.581	6.186	19366.582
2021）	PLR（MLP-Ensemble）（Gorishniy 等人，	0.970	0.664	0.705	0.960	0.391	0.734						0.966	0.487	3.319	10206.127
2022）	PLR（FT-T-Ensemble））（Gorishniy 等		-	-	-		0.970	0.674	-	-	-	-	0.467	3.050	-
人，2022）	T2G-Former（Yan 等人，2023）	0.734	-	-	-		0.972	0.646	-	-	-	-	0.464	3.162	-
		0.734	0.746	0.884	0.881	0.968	0.656	0.717	0.964	0.391	0.725	0.751	0.985	0.455	3.138	10750.850
SSL(MaskXent)+微调	SSL(ValueRecon)	0.892	0.897	0.970	0.698	0.717	0.963	0.383	0.719	0.731	0.894	0.899	0.985	0.479	3.086	10204.559
+微调	SSL(MaskXent+ValueRecon)+微	0.969	0.690	0.712	0.963	0.381	0.727	0.737	0.894	0.896	0.968	0.658	0.984	0.478	3.119	10333.400
调		0.709	0.959	0.第382章									0.984	0.475	3.257	10708.780
我们的 – SSL(BinRecon)+微调		0.737	0.764	0.897	0.904	0.971	0.720	0.728	0.966	0.388			0.986	0.464	2.989	9757.950

第六节规定。

回归：最后，我们测试了表 1c 中下游任务为回归的八个数据集。由于评估指标是 RMSE，因此值越低，表示情况越好。与其他下游任务相比，回归任务在分箱借口任务中表现出最显著的改进。例如，当将我们的方法与最佳基线进行比较时，我们观察到 HO 数据集的改进为 10.27%，DIA 数据集的改进为 8.63%，CA 数据集的改进为 8.57%。

5.2 与监督方法的比较：微调结果

我们观察到，分箱在各种表格数据集和下游任务中持续提高了无监督学习的性能。在本节中，我们将我们的方法与在整个训练过程中利用标签信息的监督方法进行比较。我们的监督基线包括基于树的算法，例如 XGBoost（Chen & Guestrin，2016）和 CatBoost（Prokhorenkova 等，2018），最近的深度学习方法和第 5.1 节中讨论的 SSL 方法的微调结果。由于监督基线通常需要大量的超参数调整，我们直接参考论文中报告的性能。当论文中没有报告性能时，我们训练使用论文中描述的默认设置或将其留空。对于我们的方法，我们首先使用默认设置和 BinRecon 损失以无监督的方式训练编码器网络。然后，我们对预训练的编码器进行微调。训练细节在补充 C 中提供。

结果总结在补充材料中的表 2 和表 7、10 中。令人惊讶的是，我们的方法始终优于基于树的方法和深度学习方法，即使它仅依赖于在预训练期间将目标函数更改为离散化箱。平均而言，我们的表现优于 XGBoost 5.55%（最大 27.14%）、CatBoost 2.18%（最大 8.26%）、最先进的深度学习方法 (T2G-Former) 2.30%（最大 9.76%），以及其他 SSL 方法的微调结果 1.55%（最大 4.38%）。

我们方法的卓越性能主要归功于其无监督预训练阶段，这种策略在深度学习中特别有效，而在基于树的算法中却不存在。其成功的关键在于在预训练期间操纵适当的归纳偏差。对于我们的方法，分箱目标有效地利用了不规则性并减轻了特征之间的异质性，如第 4 节所述。由于成功实施了这种预训练策略，我们的方法在广泛的领域中实现了卓越的性能

数据集。

6.讨论

6.1. 分箱个体因素的消融研究

在本节中，我们将仔细研究分箱各个组成部分的贡献，详见第 4 节。具体来说，我们将研究辨别每个特征中样本的顺序、将所有特征标准化为相等的集合以及对相似的值进行分组的作用。BinRecon 封装了所有三个元素，而 ValueRecon 则完全忽略了它们。为了剖析每个因素的影响，我们系统地将它们从 Bin-Recon 损失中逐一消除，如下所示。

- 排序：我们针对每个特征使用不同的随机种子来打乱 bin 索引。
- 标准化为相等集合：我们用每个箱的平均值替换箱索引。然后，每个特征包含不同范围内的不同元素。
- 分组：我们设置*电视杰=/德杰* 火车/针对每个特征。在此在这种情况下，每个唯一值对应一个单独的箱，并且仅保留顺序信息。

如补充材料中的表 11 所示，我们发现消除标准化因子导致的性能下降最为显著，在 25 个数据集中，有 15 个数据集的性能平均下降了 6.85%。这种下降幅度比消除所有三个因子的效果要大得多。从这些观察结果中，我们推断，标准化因子对于成功实施分箱至关重要，它使所有特征都处于具有相同元素的均匀分布上。

6.2. 箱体数量与下游任务性能之间的依赖关系

在本节中，我们研究了 BinXent 和 BinRecon 中箱数与下游任务性能之间的关系，无需输入转换。如补充材料中的图 6 所示，箱数与归一化性能之间没有明确的关系（皮尔逊相关系数ρ_2=0。01，Kendall 等级相关τ=0。16 对于 BinXent，ρ_2=0。04，τ= 0。二十七对于 BinRecon)，除了箱子数量不应太小，但箱子数量并不总是越大越好。这个结果并不奇怪，因为使用太少的箱子会消除必要的信息，而使用太多的箱子会削弱分箱的好处。然而，ρ_2=0。三十四和τ=0。60在一个示例子集中观察到：BinRecon 损失的回归任务，其中 binRecon 损失少于 100 个 bin。

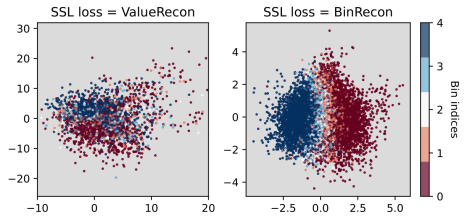


图 4：使用 HO 数据集进行可视化分析。为了提高可解释性，我们根据不同的目标函数对学习到的表示向量实施 PCA，绘制前两个主成分。颜色表示每个样本的 bin 索引。

6.3. 除非作为前置任务提供，否则 bin 信息不可用

到目前为止，我们发现 bin 信息对于在各种表格数据问题中实现出色的表示至关重要。但是，即使我们不将 bin 信息用作显式借口任务，它仍然可以从原始值中访问。在本节中，我们将评估在 SSL 期间优化 ValueRecon 或 MaskXent 时，学习到的表示能够多准确地预测 bin 索引。为了衡量这一点，我们测量了与 BinRecon 案例结果的相对误差增加。如补充材料中的表 12 所示，当不提供 bin 信息时，预测误差急剧增加，平均为 66.3%。这强调了虽然 bin 信息可以从数据中得出，但除非将其用作借口任务，否则其效用会大大降低。

6.4 可视化分析

为了证明与其他方法相比，分箱任务在有效捕获将连续输入映射到离散化箱索引的不规则函数方面具有更优越的能力，我们在图 4 中展示了 SSL 之后的表示向量的可视化分析。由于表示向量的高维性质，我们实施了 PCA 以提高可解释性。在可视化中，箱索引表示为不同的颜色。从该分析中得出一个独特的模式：在 BinRecon 的情况下，表示向量根据其箱索引进行特定分组。当使用其他借口任务时，这种明显的聚类并不明显。这些发现凸显了分箱作为借口任务的有效性。它展示了这种方法的独特能力，使编码器能够准确捕捉不规则函数，从而将其与其他方法区分开来。

6.5. SSL 期间优化多个损失函数

在第 4 节中，我们介绍了通过使用共享共同输入表示的各种解码器在 SSL 中集成多个损失函数的潜力，*是*。这一策略强调了我们方法的灵活性，尽管它并不是

表 3：使用分箱作为数据增强（随机量化，RQ）与使用分箱来定义输出标签（我们的）时表格 SSL 的微调性能比较。

训练方法	你好 ↑ 肺动脉高压 ↑	操作系统 ↑	邮局 ↑ 一氧化碳 ↑	通用电气 ↑	画外音 ↑ 艾尔 ↑	他 ↑	MNIST ↑	加州 ↓ 何 ↓ 财务信息 ↓
RQ (Wu et al., 2023)	0.717	0.736	0.896	0.886	0.969	0.690	0.719	0.959 0.379 我们的 0.984 0.475 3.159 10398.616
	0.737	0.764	0.897	0.904	0.971	0.720	0.728	0.966 0.388 0.986 0.464 2.989 9757.950

我们当前研究的主要焦点。

我们对 GE 数据集的初步调查显示，这种多解码器策略可以提高性能。具体来说，在 2 个 bin 和 0.1 随机掩蔽概率的条件下，使用单个 MaskXent 或 ValueRecon 损失进行训练分别可获得 0.509 和 0.553 的线性探测性能。另一方面，使用单个 BinRecon 损失进行训练可获得 0.560 的线性探测性能。此外，引入额外的解码器以同时优化 BinRecon 和 MaskXent 损失（具有相同的权重）或 BinRecon 和 ValueRecon 损失（具有相同的权重），可将线性探测性能提高到 0.577。这些观察结果表明，将 binning 损失与其他 SSL 目标（如 MaskXent）结合起来可以改善表格表示学习。

6.6. 分箱作为输入转换

Wu 等人（2023 年）引入了随机量化（RQ）作为对比表征学习的一种数据无关的增强策略，该策略将分箱直接应用于*输入样本*相比之下，我们的方法主要利用经典的分箱技术*输出标签*在基于自动编码的自监督学习框架内。

为了确定分箱在表格表示学习中的更有效应用，我们使用官方代码实现了 RQ 增强，并遵循与我们的基线方法相同的实验设置，如手稿和补充材料中所述。对于 RQ 增强的超参数，我们选择了与我们的方法相同的范围：{2，5，10，20，50，100}。

如表 3 所示，我们的方法始终优于 RQ 方法。根据 Wu 等人 (2023) 的说法，使用分箱作为增强策略会导致输入样本中不可避免地出现信息丢失。虽然在具有固有冗余的域（例如图像）中，可以通过其他渠道或局部模式减轻一些信息丢失，但表格域通常缺乏这种补偿机制。例如，在预测糖尿病的医疗数据集中，由于表格数据中特征的独立性，减少血糖水平等关键特征的细节无法通过其他变量来补偿。因此，我们预计在表格域中使用分箱作为输入转换可能并不有效，因为它会导致系统地删除重要信息。

7. 结论

在这项工作中，我们提出了一种基于分箱的新型借口任务，它可以操纵表格数据集的独特属性。分箱任务可以有效地解决表格 SSL 中的挑战，包括减轻特征异质性和学习不规则性。重要的是，我们的方法专注于修改目标函数，并且独立于特定的架构或增强方法。基于大量实验，我们发现分箱任务不仅可以改进无监督表示学习，而且是一种强大的预训练策略，可以实现与基于树和其他深度学习方法相比始终如一的卓越性能。在这项研究中，我们发现了利用表格数据的固有属性作为 SSL 借口任务的潜力。然而，许多独特的特性仍未被探索，例如特征之间的层次关系。我们希望我们的工作能够激发未来对特定于表格数据的 SSL 的进一步研究。

潜在的更广泛影响

本文有助于推动机器学习领域的发展，尤其关注表格数据，这是众多实际应用中普遍存在的领域。我们的工作有可能显著增强各个领域的数据分析和预测建模，包括医疗保健、金融和社会科学，表格数据在这些领域被广泛使用。虽然我们相信我们的方法可以带来积极的社会影响，例如改善决策和更高效的数据处理，但我们也承认负责任地使用的重要性。至关重要的是要确保在部署这些先进的机器学习技术时考虑到道德因素并致力于减轻偏见。我们希望这项研究能够激发机器学习的进一步创新，同时促使人们继续讨论其道德和社会影响。

参考

Arik, S. Ö. 和 Pfister, T. Tabnet: 专注的可解释性表格学习。在*AAAI 人工智能会议论文集*，第 35 卷，第 6679-6687 页，

2021 年。

巴迪尔利，S.，刘，X.，邢，Z.，博米克，A.，多安，K.，和 Keerthi，SS 梯度增强神经网络：Grownet。*arXiv 预印本 arXiv:2002.07971*，2020年。

Bahri, D.、Jiang, H.、Tay, Y. 和 Metzler, D. Scarf：使用随机特征损坏的自监督对比学习。*arXiv 预印本 arXiv:2106.15147*，2021年。

Baldi, P.、Sadowski, P. 和 Whiteson, D. 寻找前利用深度学习研究高能物理中的耳粒子。 *自然通讯*, 5(1):4308, 2014.

Beyazit, E.、Kozaczuk, J.、Li, B.、Wallace, V. 和 Fadlallah，BH 表格深度学习的归纳偏差。在 *第三十七届神经信息处理系统会议*，2023年。

Blackard, JA 和 Dean, DJ 比较准确度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中的应用。*农业中的计算机和电子产品*, 24(3): 131–151, 1999.

Borisov, V.、Leemann, T.、Seßler, K.、Haug, J.、Pawelczyk, M. 和 Kasneci, G. 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。

Chen, J.、Liao, K.、Wan, Y.、Chen, DZ 和 Wu, J. Danets：用于表格数据分类和回归的深度抽象网络。*AAAI 人工智能会议论文集*，第 36 卷，第 3930–3938 页，2022 年。

Chen, K.-Y.、Chiang, P.-H.、Chou, H.-R.、Chen, T.-W. 和 Chang, T.-H. Trompt：面向表格数据的更好的深度神经网络。*arXiv 预印本 arXiv:2305.18446*，2023 年。

Chen, P.、Sarkar, S.、Lausen, L.、Srinivasan, B.、Zha, S.、Huang、R. 和 Karypis、G. Hytrel：超图增强表格数据表示学习。*arXiv 预印本 arXiv:2307.08623*，2023 年。

Chen, T. 和 Guestrin, C. Xgboost：可扩展的树提升 - 系统。在 *第 22 届 acm sigkdd 知识发现和数据挖掘国际会议论文集*，第 785–794 页，2016 年。

Cherepanova, V.、Levin, R.、Somepalli, G.、Geiping, J.、Bruss, CB、Wilson, AG、Goldstein, T. 和 Goldblum, M. 表格深度学习中特征选择的性能驱动基准。*神经信息处理系统的进展*，36，2024。

Cortez, P.、Cerdeira, A.、Almeida, F.、Matos, T. 和 Reis, J. 通过物理化学特性的数据挖掘来建模葡萄酒偏好。*决策支持系统*, 47(4): 547–553, 2009。

Dougherty, J.、Kohavi, R. 和 Sahami, M. 监督并连续特征的无监督离散化。在 *机器学习论文集 1995*，第 194-202 页。爱思唯尔，1995 年。

Geusebroek, J.-M.、Burghouts, GJ 和 Smeulders, AW 阿姆斯特丹物体图像库。*国际计算机视觉杂志* ,61:103–112,2005。

Gorishniy, Y.、Rubachev, I.、Khrulkov, V. 和 Babenko, A. 重新审视表格数据的深度学习模型。*神经信息处理系统的进展*, 34: 18932–18943，2021年。

Gorishniy, Y., Rubachev, I., 和 Babenko, A. 关于嵌入表格深度学习中的数值特征。*神经信息处理系统的进展*, 35: 24991–25004, 2022年。

Goyal, P.、Dollár, P.、Girshick, R.、Noordhuis, P.、Wesolowski, L.、Kyrola, A.、Tulloch, A.、Jia, Y. 和 He, K. 准确的大型 minibatch sgd：1 小时内训练 imagenet。*arXiv 预印本 arXiv:1706.02677*，2017年。

Grinsztajn, L.、Oyallon, E. 和 Varoquaux, G. 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吗？*神经信息处理系统的进展*, 35: 507–520, 2022年。

Guyon, I.、Sun-Hosoya, L.、Boullé, M.、Escalante, HJ, Escalera, S., Liu, Z., Jajetic, D., Ray, B., Saeed, M., Sebag, M., et al. 对 automl 挑战系列的分析。 *自动机器学习*，177，2019。

韩菁、裴建军和童华.*数据挖掘：概念和技术*. 摩根・考夫曼，2022 年。

Hollmann, N.、Müller, S.、Eggensperger, K. 和 Hutter，F.Tabpfn：一种可在一秒钟内解决小型表格分类问题的变换器。*arXiv 预印本 arXiv:2207.01848*，2022年。

Huang, X.、Khetan, A.、Cvitkovic, M. 和 Karnin, Z. Tabtransformer：使用上下文嵌入的表格数据建模。*arXiv 预印本 arXiv:2012.06678*，2020年。

Kohavi，R. 等人。提高朴素贝叶斯的准确性分类器：决策树混合体。在*克德*，第 96 卷，第 202-207 页，1996 年。

Kotelnikov, A.、Baranchuk, D.、Rubachev, I. 和 Babenko，A. Tabddpm：使用扩散模型对表格数据进行建模。在*国际机器学习会议*，第 17564-17579 页。PMLR，2023 年。

Loshchilov, I. 和 Hutter, F. Sgdr: 随机梯度通过热重启来实现下降。 *arXiv 预印本 arXiv:1608.03983*，2016 年。

Loshchilov, I. 和 Hutter, F. 解耦权重衰减调节化。*arXiv 预印本 arXiv:1711.05101*，2017年。

Majmundar, K.、Goyal, S.、Netrapalli, P. 和 Jain, P. 满足：表格数据的掩码编码。*arXiv 预印本 arXiv:2206.08564*，2022年。

McElfresh, D.、Khandagale, S.、Valverde, J.、Ramakrishnan, G.、Goldblum, M.、White, C. 等。神经网络在表格数据上的表现何时优于提升树？*arXiv 预印本 arXiv:2305.02997*，2023年。

Moro, S.、Laureano, R. 和 Cortez, P. 使用数据挖掘银行直接营销：crispdm 方法论的一种应用。2011年。

Müller, S.、Hollmann, N.、Arango, SP、Grabocka, J. 和 Hutter，F. Transformers 可以进行贝叶斯推理。*arXiv 预印本 arXiv:2112.10510*，2021年。

Pace, RK 和 Barry, R.稀疏空间自回归。*统计与概率快报*, 33(3):291–297, 1997.

Popov, S., Morozov, S., 和 Babenko, A. 神经遗忘用于表格数据深度学习的决策集成。 *arXiv 预印本 arXiv:1909.06312*，2019年。

普罗霍伦科娃，L.，古谢夫，G.，沃罗贝夫，A.，多罗古什，AV 和 Gulin，A. Catboost：具有分类特征的无偏提升。*神经信息处理系统的进展*，2018 年 31 日。

Qin, T. 和 Liu, T.-Y. 介绍 letor 4.0 数据集。*论文集预印本 arXiv:1306.2597*，2013 年。

Rubachev, I.、Alekberov, A.、Gorishniy, Y. 和 Babenko，A.重新审视表格深度学习的预训练目标。*arXiv 预印本 arXiv:2207.03208*，2022年。

Sakar, CO、Polat, SO、Katircioglu, M. 和 Kastro, Y. 使用多层感知器和 lstm 循环神经网络实时预测在线购物者的购买意愿。*神经计算与应用*, 31: 6893–6908, 2019年。

Somepalli，G.、Goldblum，M.、Schwarzschild，A.、Bruss，CB 和 Goldstein、T. Saint：通过行注意和对比预训练改进表格数据的神经网络。*arXiv 预印本 arXiv:2106.01342*，2021年。

Stewart, L.、Bach, F.、Berthet, Q. 和 Vert, J.-P. Regres-分类：任务制定对神经网络特征的影响。在*国际人工智能与统计学会议*，第 11563-11582 页。PMLR，2023 年。

Ucar, T.、Hajiramezanali, E. 和 Edwards, L. 子选项卡：子-为自监督表征学习设置表格数据的特征。*神经信息处理系统的进展*, 34: 18853–18865，2021年。

Vanschoren, J.、Van Rijn, JN、Bischl, B. 和 Torgo, L. Openml：机器学习中的网络科学。*ACM SIGKDD 探索简讯*, 15(2):49–60, 2014。

Vincent, P.、Larochelle, H.、Bengio, Y. 和 Manzagol, P.-A. 使用去噪自动编码器提取和组合稳健特征。在*第 25 届机器学习国际会议论文集*，第 1096–1103 页，2008年。

Wang, R.、Shivanna, R.、Cheng, D.、Jain, S.、Lin, D.、Hong, L. 和 Chi, E. 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。*2021 年网络会议论文集*，第 1785–1797 页，2021 年。

吴华、雷晨、孙鑫、王佩珊、陈倩、程K.-T.、Lin、S. 和 Wu、Z. 随机 量化：数据不可知自监督学习的通用增强。在*IEEE/CVF 国际计算机视觉会议论文集*，第 16305–16316 页，2023 年。

Yan, J.、Chen, J.、Wu, Y.、Chen, DZ 和 Wu, J. T2g-前者：将表格特征组织成关系图可以促进异构特征交互。在*AAAI 人工智能会议论文集*，第 37 卷，第 10720–10728 页，2023 年。

Yoon, J.、Zhang, Y.、Jordon, J. 和 van der Schaar，M. Vime：将自我和半监督学习的成功扩展到表格领域。*神经信息处理系统的进展*, 33: 11033–11043，2020年。

Zhu, B.、Shi, X.、Erickson, N.、Li, M.、Karypis, G. 和 Shoaran，M. Xtab：表格转换器的跨表预训练。*arXiv 预印本 arXiv:2305.06090*，2023年。

A.数据集详细信息

在本研究中，我们使用了 25 个公共数据集，其中大部分来自 OpenML（Vanschoren 等人，2014 年）库，包括以前研究中经常使用的数据集（Yoon 等人，2020 年；Ucar 等人，2021 年；Gorishniy 等人，2021 年；2022 年）。每个数据集只有一个训练-验证-测试分割，因此所有算法都使用与以前研究相同的分割（Gorishniy 等人，2021 年；2022 年；Rubachev 等人，2022 年）。我们在表 4 中总结了数据集的主要属性。对于每个数据集，我们根据训练样本的数量使用预定义的批量大小：当训练样本数量小于 1000 时为 64，当训练样本数量大于 1000 且小于 5000 时为 128，当训练样本数量大于 5000 且小于 10000 时为 256，当训练样本数量大于 10000 且小于 50000 时为 512，当训练样本数量大于 50000 时为 1024。

当训练数据集中唯一值的数量小于 20（AL、MNIST、p-MNIST、MI 为 5）时，我们将该特征视为分类特征。分类变量被输入到 FT-Transformer 的特征标记器中，而 MLP 对它们没有其他操作。对于 MNIST 和 p-MNIST 数据集，我们忽略整个训练数据集中只有一个可能值的特征。

在本研究中，我们引入了一个新的 p-MNIST 数据集，作为著名 MNIST 数据集的简单修改。在构建 p-MNIST 数据集时，我们根据单一的预定义顺序对所有样本的像素值进行排列。具体来说，我们首先使用固定的随机种子生成像素索引的排列 ([0, 783])。然后将此预定顺序一致地应用于整个数据集内所有图像的像素值。此方法背后的主要目的是破坏 MNIST 图像中存在的固有局部性（即附近的列更相关），从而使数据更像表格，其中空间局部性不太明显或可量化（即附近的列不一定更相关）。

表 4：数据集摘要。

缩写	姓名	# 火车	# 验证	# 测试	# 数量	# 猫	任务类型	批次大小
中	客户流失建模[2]	6400	1600	2000	4	6	宾类	256
你好	Higgs Small（Baldi 等人，2014） 成	62751	15688	19610	24	4	宾类	1024
广告	人（Kohavi 等人，1996） 银行营销	26048	6513	16281	2	12	宾类	512
马来亚	（Moro 等人，2011） 菲律宾	28934	7234	9043	7	9	宾类	512
肺动脉高压	（Guyon 等人，2019） 在线购物者	3732	933	1167	308	0	宾类	128
操作系统	（Sakar 等人，2019） 德国信贷数据	7891	1973	2466	8	9	宾类	256
CS	集[3]	640	160	200	20	0	宾类	64
邮局	音素	3458	865	1081	5	0	宾类	128
一氧化碳	Covertype (Blackard & Dean, 1999)	371847	92962	116203	四十四	7	多类别	1024
加时赛	Otto Group Products[4]	39601	9901	12376	80	十三	多类别	512
通用电气	手势阶段	6318	1580	1975	三十二	0	多类别	256
画外音	福尔克特[5]（Guyon 等人，2019）	37318	9330	11662	147	33	多类别	512
水质	葡萄酒质量（Cortez 等人，2009）	4157	1040	1300	11	0	多类别	128
艾尔	ALOI（Geusebroek 等人，2005）	69120	17280	21600	124	4	多类别	1024
他	Helena（Guyon 等人，2019） 手	62752	15688	19610	二十七	0	多类别	512
MNIST	写数字图像	50000	10000	10000	627	90	多类别	512
MNIST	排列 MNIST	50000	10000	10000	627	90	多类别	512
加州	加州住宅 (Pace & Barry, 1997) 16H 号	13209	3303	4128	8	0	回归	512
何	住宅[6]	14581	3646	4557	16	0	回归	512
财务信息	国际足联	12273	3069	3836	二十八	0	回归	512
MI	MSLR-WEB10K(Fold 1) (Qin & Liu, 2013) 8	723412	235259	241521	131	5	回归	1024
碘化钾	连杆机械臂的正向动力学[6]	5242	1311	1639	8	0	回归	256
中央处理器	计算机活动数据库[6]	5242	1311	1639	8	0	回归	256
迪亚	钻石	34521	8631	10788	9	0	回归	512
发光	电[7]	24623	6156	7695	7	0	回归	512

[2]https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling
[3]https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data
[4]https://www.kaggle.com/c/otto-group-product-classification-challenge/data
[5]https://automl.chalearn.org/data
[6]http://www.ncc.up.pt/ ltorgo/Regression/DataSets.html
[7]https://github.com/LeoGrin/tabular-benchmark

B. 实施细节

我们对 SSL 使用的优化策略如下。我们不调整任何超参数，所有情况都应用相同的配置。

- 优化器：AdamW（Loshchilov & Hutter，2017）
- 学习率：1e-4
- 权重衰减：1e-5
- 纪元：1000
- 学习率调度器：余弦退火调度器（Loshchilov & Hutter，2016；Goyal et al.，2017）

对于与 SSL 相关的超参数，我们尝试了*页米* *∈{*0。1，0。2，0。3，0。4，0。5，0。6，0。7，0。8，0。9*}*和*电视∈ {*2，5，10，20，50，100*}*。当我们结合变换函数和分箱方法时，为了减少超参数空间，我们尝试了*页米∈{*0。1，0。2，0。3*}*和*电视∈{*2，10*}*对于 MLP，以及*页米∈{*0。1，0。2*}*和*电视∈{*2，10*}*对于 FT-Transformers。在 SSL 之后，我们使用线性头评估预训练的表示。使用不同的随机种子对线性头进行 10 次训练，并报告平均性能。

对于其他最先进的模型，我们直接参考论文中报告的性能，以减少随机种子或调整细节带来的歧义。

B.1. 多层感知器

对于 MLP，我们设置了在有监督设置下验证性能最佳的架构，其中编码器网络*F埃*在深度（1、2、3、4、5）和宽度（128、256、512、1024）上进行网格搜索后。表示大小确定为与 MLP 的宽度相同。以下解码器网络*Fd*定义为对称*F埃*。

对于监督学习，我们使用与上面总结的 SSL 相同的配置，只是学习率为 0.001，迭代次数为 100。我们总结了所有数据集的最佳设置，如下所示。

表 5：MLP 架构。

深度	数据集	宽度	数据集
1	CH、HI、AD、BM、OS、FI、CS	128	CH、HI、AD、BM、OS、FI、MI、CA
2	MI、CPU、HE、OT、AL	256	CS、HE、KI、PH、HO
3	CA、KI、MNIST、EL	512	CPU、WQ、p-MNIST、DIA
4	WQ、p-MNIST、PH、HO、CO、GE、VO、PO	1024	CO、GE、VO、PO、MNIST、EL、OT、AL
5	DIA		

B.2. FT-Transformer

我们没有对 FT-Transformer 进行任何超参数调整，我们使用 (Gorishniy et al., 2021) 中定义的默认设置，块数为 3。对于 MI、MNIST 和 p-MNIST 等三个大型数据集，由于计算预算的原因，我们将块数设置为 1。对于表示大小，我们采用 MLP 案例中的值。对于*Fd*，我们使用架构与表5相同的MLP网络。

B.3. T2G-成型机

我们没有对 FT-Transformer 进行任何超参数调整，我们使用 (Yan et al., 2023) 中定义的默认设置，层数为 3，token 维度为 192，heads 数量为 8，激活函数为 ReGLU。对于表示大小，我们采用 MLP 案例中的值。对于*Fd*，我们使用架构与表5相同的MLP网络。

B.4. 线性评估和微调

对于线性评估，我们使用与 SSL 相同的优化配置，但 100 个 epoch 的学习率为 0.01。对于微调，我们使用与监督案例相同的设置，进行 50 或 100 个 epoch。

C. 完整结果

在这里，我们展示了我们手稿的综合结果，并附上了 10 次重复实验得出的标准差。

表 6：表 1 的完整结果。我们重复评估 10 次，并提供了平均值和标准差。

（a）二元分类

掩蔽	掩蔽值	目标	中	你好	广告	马来亚	肺动脉高压	操作系统	CS	邮局
错误的	-	价值侦察	0.810 ±0.001	0.651 ±0.000	0.837 ±0.000	0.899 ±0.000	0.728 ±0.001	0.883 ±0.000	0.709 ±0.003	0.851 ±0.000	0.816 ±0.000
错误的	请求权限	价值侦察	0.654 ±0.000	0.842 ±0.000	0.898 ±0.000	0.727 ±0.001	0.882 ±0.001	0.725 ±0.002	0.842 ±0.000	0.807 ±0.001	0.672 ±0.000
真的	宪法。	MaskXent	0.836 ±0.000	0.899 ±0.000	0.715 ±0.000	0.893 ±0.000	0.708 ±0.004	0.845 ±0.000	0.810 ±0.000	0.653 ±0.000	0.839 ±0.000
真的	宪法。	价值侦察	0.900 ±0.000	0.734 ±0.001	0.884 ±0.000	0.718 ±0.002	0.849 ±0.001				
真的	宪法。	MaskXent+ValueRecon	0.817 ±0.001	0.669 ±0.000	0.835 ±0.000	0.900 ±0.000	0.724 ±0.001	0.877 ±0.000	0.706 ±0.000	0.837 ±0.002	MaskXent
真的	随机的		0.814 ±0.000	0.681 ±0.000	0.843 ±0.000	0.901 ±0.000	0.710 ±0.000	0.883 ±0.000	0.706 ±0.000	0.853 ±0.000	0.811 ±0.000
真的	随机的	价值侦察	0.661 ±0.000	0.838 ±0.000	0.898 ±0.000	0.736 ±0.001	0.885 ±0.000	0.714 ±0.003	0.842 ±0.000		
真的	随机的	MaskXent+ValueRecon	0.804 ±0.001	0.647 ±0.000	0.826 ±0.000	0.899 ±0.000	0.715 ±0.003	0.879 ±0.001	0.713 ±0.003	0.861 ±0.001	
错误的	-	宾森特	0.817 ±0.001	0.683 ±0.000	0.845 ±0.000	0.901 ±0.000	0.732 ±0.001	0.886 ±0.000	0.738 ±0.000	0.851 ±0.001	0.823 ±0.000
错误的	-	宾雷康	0.687 ±0.000	0.840 ±0.000	0.900 ±0.000	0.737 ±0.000	0.889 ±0.000	0.724 ±0.005	0.865 ±0.000	0.820 ±0.000	0.672 ±0.000
真的	宪法。	宾雷康	0.843 ±0.000	0.899 ±0.000	0.730 ±0.000	0.896 ±0.000	0.718 ±0.004	0.858 ±0.000	0.819 ±0.001	0.682 ±0.000	0.846 ±0.000
真的	随机的	宾雷康	0.898 ±0.000	0.735 ±0.000	0.894 ±0.000	0.718 ±0.004	0.858 ±0.000				

（b）多类分类

掩蔽	掩蔽值	目标	一氧化碳	加时赛	通用电气	画外音	水质	艾尔	他	MNIST	MNIST
错误的	-	价值侦察	0.769 ±0.000	0.776 ±0.000	0.527 ±0.001	0.619 ±0.000	0.568 ±0.001	0.931 ±0.000	0.353 ±0.000	0.965 ±0.000	0.928 ±0.000	0.775 ±0.000
错误的	请求权限	价值侦察	0.771 ±0.000	0.531 ±0.001	0.620 ±0.000	0.559 ±0.000	0.930 ±0.000	0.350 ±0.000	0.948 ±0.000	0.706 ±0.000	0.784 ±0.000	0.777 ±0.000
真的	宪法。	MaskXent	0.518 ±0.001	0.545 ±0.000	0.547 ±0.000	0.909 ±0.001	0.341 ±0.000	0.793 ±0.000	0.554 ±0.000	0.783 ±0.000	0.791 ±0.000	0.557 ±0.001
真的	宪法。	价值侦察	0.622 ±0.000	0.586 ±0.001	0.931 ±0.000	0.354 ±0.000	0.966 ±0.000	0.925 ±0.000				
真的	宪法。	MaskXent+ValueRecon	0.750 ±0.000	0.774 ±0.001	0.519 ±0.005	0.610 ±0.000	0.571 ±0.001	0.931 ±0.000	0.360 ±0.000	0.941 ±0.000	0.907 ±0.000	MaskXent
真的	随机的		0.763 ±0.000	0.791 ±0.000	0.555 ±0.000	0.549 ±0.000	0.544 ±0.001	0.925 ±0.000	0.336 ±0.000	0.945 ±0.000	0.817 ±0.000	0.761 ±0.000
真的	随机的	价值侦察	0.782 ±0.000	0.538 ±0.001	0.625 ±0.000	0.573 ±0.001	0.930 ±0.000	0.357 ±0.000	0.956 ±0.000	0.934 ±0.000		
真的	随机的	MaskXent+ValueRecon	0.769 ±0.000	0.779 ±0.001	0.521 ±0.004	0.564 ±0.001	0.519 ±0.004	0.925 ±0.001	0.353 ±0.001	0.945 ±0.000	0.906 ±0.001	
错误的	-	宾森特	0.742 ±0.000	0.781 ±0.000	0.517 ±0.001	0.600 ±0.001	0.565 ±0.001	0.903 ±0.001	0.354 ±0.000	0.956 ±0.000	0.908 ±0.000	0.784 ±0.000
错误的	-	宾雷康	0.783 ±0.000	0.544 ±0.001	0.625 ±0.000	0.592 ±0.001	0.935 ±0.000	0.357 ±0.000	0.964 ±0.000	0.950 ±0.000	0.812 ±0.000	0.792 ±0.000
真的	宪法。	宾雷康	0.559 ±0.001	0.647 ±0.000	0.581 ±0.001	0.943 ±0.000	0.359 ±0.000	0.974 ±0.000	0.964 ±0.000	0.814 ±0.000	0.794 ±0.000	0.580 ±0.000
真的	随机的	宾雷康	0.655 ±0.000	0.574 ±0.001	0.949 ±0.000	0.365 ±0.000	0.981 ±0.000	0.971 ±0.000				

（c）回归

掩蔽	掩蔽值	目标	加州	何	财务信息	MI	碘化钾	中央处理器	迪亚	发光
错误的	-	价值侦察	0.749 ±0.000	4.241 ±0.001	13900.720 ±0.816	0.784 ±0.000	0.163 ±0.000	3.876 ±0.002	0.714 ±	1016.641 ±0.191	0.399 ±0.001
错误的	请求权限	价值侦察	0.000	4.165 ±0.000	13684.367 ±0.778	0.784 ±0.000	0.162 ±0.000	3.751 ±0.002	0.709 ±0.000	4.548	1056.453 ±0.229	0.398 ±0.000
真的	宪法。	MaskXent	±0.000	13473.750 ±1.371	0.788 ±0.000	0.185 ±0.000	4.475 ±0.033	0.693 ±0.000	4.086 ±0.000		1259.744 ±1.066	0.396 ±0.000
真的	宪法。	价值侦察	13518.683 ±0.936	0.778 ±0.000	0.160 ±0.000	3.728 ±0.003					952.444 ±0.130	0.394 ±0.000
真的	宪法。	MaskXent+ValueRecon	0.700 ±0.001	4.157 ±0.045	13915.875 ±18.078	0.775 ±0.000	0.174 ±0.001	5.644 ±0.078	2797.034 ±191.324		0.398 ±0.001	MaskXent
真的	随机的		0.677 ±0.000	4.297 ±0.000	13826.641 ±0.624	0.782 ±0.000	0.176 ±0.000	3.951 ±0.001	0.713 ±		1358.135 ±0.191	0.388 ±0.000
真的	随机的	价值侦察	0.000	4.127 ±0.000	13668.988 ±1.262	0.777 ±0.000	0.162 ±0.000	3.760 ±0.002			986.306 ±0.359	0.396 ±0.000
真的	随机的	MaskXent+ValueRecon	0.701 ±0.003	4.136 ±0.028	14107.645 ±29.125	0.780 ±0.001	0.166 ±0.001	4.506 ±0.034	1917.875 ±123.359		0.397 ±0.008	
错误的	-	宾森特	0.690 ±0.000	4.116 ±0.001	13038.762 ±1.618	0.776 ±0.000	0.170 ±0.000	3.717 ±0.006	0.622 ±		1207.923 ±2.552	0.383 ±0.000
错误的	-	宾雷康	0.000	3.766 ±0.000	13453.309 ±0.832	0.767 ±0.000	0.158 ±0.000	3.208 ±0.001	0.634 ±0.000	3.765	897.645 ±0.182	0.370 ±0.000
真的	宪法。	宾雷康	±0.000	13208.133 ±1.960	0.773 ±0.000	0.158 ±0.000	3.156 ±0.002	0.619 ±0.000	3.703 ±0.000		957.801 ±0.070	0.371 ±0.000
真的	随机的	宾雷康	13075.474 ±0.800	0.773 ±0.000	0.160 ±0.000	3.183 ±0.001					870.283 ±0.093	0.368 ±0.000

表 7：基于 10 次重复实验的标准偏差微调结果。在这种情况下，模型以无监督方式进行预训练（即优化 BinRecon 损失）并以监督方式进行微调。对于每个数据集和编码器，我们尝试了各种输入转换方法和箱数的组合，如补充 B 中所述。然后，我们重复了根据验证性能确定的最佳情况。

编码器	你好 ↑	肺动脉高压 ↑	操作系统 ↑	邮局 ↑	一氧化碳 ↑	通用电气 ↑	画外音 ↑	艾尔 ↑	他 ↑	MNIST ↑	加州 ↓	何 ↓	财务信息 ↓
多层感知处理器	0.717 ±0.001	0.738 ±0.009	0.897 ±0.000	0.893 ±0.004	0.969 ±0.000	0.673 ±0.004	0.728 ±0.001	0.963 ±0.001	0.388 ±0.001	0.986 ±0.000	0.502 ±0.002	2.989 ±0.015	9963.609 ±23.173
FT-Transformer	0.703 ±0.004	0.742 ±0.011	0.882 ±0.004	0.904 ±0.003	0.971 ±0.000	0.698 ±0.006	0.720 ±0.003	0.961 ±0.001	0.374 ±0.002	0.978 ±0.001	0.475 ±0.003	3.173 ±0.024	T2G-成型机 9757.950 ±210.751
	0.737 ±0.001	0.764 ±0.008	0.892 ±0.003	0.895 ±0.005	0.967 ±0.001	0.720 ±0.002	0.725 ±0.001	0.966 ±0.001	0.378 ±0.002	0.985 ±0.000	0.464 ±0.001	3.144 ±0.041	10155.818 ±132.559

表 8：在所有数据集上使用一组固定的超参数进行微调的结果（编码器网络：T2G-Former，输入变换：比率为 0.2 的随机掩码，以及 bin 数量 = 10）。在这种情况下，模型以无监督方式进行预训练（即优化 BinRecon 损失）并以监督方式进行微调。即使使用这些固定参数，我们的方法仍保持着优于基线方法的竞争优势，从而肯定了我们方法的内在优势和适应性。具体而言，在这种设置下，我们的方法仍然在一系列数据集上展示了显着的性能改进，再次证明了其在广泛的超参数优化范围之外的有效性。

编码器	你好 ↑	肺动脉高压 ↑	操作系统 ↑	邮局 ↑	一氧化碳 ↑	通用电气 ↑	画外音 ↑	艾尔 ↑	他 ↑	MNIST ↑	加州 ↓	何 ↓	财务信息 ↓	平均排名
XGBoost	0.726	0.721	0.840	0.711	0.969	0.683	0.699	0.924	0.348	0.714	0.733	0.977	0.434	3.152	10372.778	3.462
SSL+Finetuning(T2G-Former、随机掩码(0.2)、MaskXent)	0.872	0.895	0.925	0.708	0.705	0.960	0.365	0.719	0.727	0. 870	0.892	0.983	0.551	3.174	10201.881	2.692
SSL+Finetuning(T2G-Former、随机掩码(0.2)、ValueRecon)	0.874	0.673	0.713	0.762	0.343							0.982	0.474	3.310	10434.967	4.000
SSL+Finetuning(T2G-Former, 随机掩蔽(0.2), MaskXent+ValueRecon)	0.721	0.757	0.873	0.891	0.839	0.657	0.699	0.958	0.351	我们的(T2G-Former, 随		0.983	0.478	3.101	10063.307	3.000
机掩蔽(0.2), Bin=10, BinRecon)	0.734	0.773	0.880	0.895	0.965	0.699	0.728	0.963	0.380			0.983	0.469	3.193	10006.578	1.462

我们还在表 7 中总结了最佳情况的详细训练设置，如下所示。

表 9：表 7 中最佳案例的训练设置。

数据集	你好	肺动脉高压	操作系统	邮局	一氧化碳	通用电气	画外音	艾尔	他	MNIST	加州	何	财务信息
编码器	T2G-成型机	T2G-成型机	多层感知处理器	FT-变压器	FT-变压器	T2G-成型机	多层感知处理器	T2G-成型机	多层感知处理器	多层感知处理器	T2G-成型机	多层感知处理器	FT-变压器
输入变换	遮蔽（随机）	遮蔽（随机）	无 遮蔽（随机）	遮蔽（固定）	遮蔽（固定）	遮蔽（随机）	遮蔽（随机）	遮蔽（固定）	遮蔽（随机）	遮蔽（固定）			
掩蔽概率（*页米*) 箱子数量	0.1	0.2	-	0.1	0.2	0.2	0.3	0.2	0.2	0.3	0.2	0.2	0.2
	2	10	2	10	10	10	10	10	2	10	10	2	2
微调时期	50	50	50	50	100	100	100	100	100	100	50	100	100

以下是表 2 中未包含的其他数据集列表的结果。同样，我们发现分箱方法始终优于其他方法。

表 10：表 2 中未包含的其他数据集列表的微调结果。

训练网络及方法	二元分类				多类分类			回归			
	中 ↑	广告 ↑	马来亚 ↑	CS ↑	加时赛	水质 ↑	MNIST ↑	MI ↓	碘化钾 ↓	迪亚 ↓	发光 ↓
基于树的机器学习算法 XGBoost											
	0.859	0.875	0.903	0.710	0.827	0.632	0.861	0.873	0.978	0.742	0.128
CatBoost	0.907	0.750	0.825	0.659					0.980	0.090	2.668
深度学习方法											
多层感知处理器	0.838	0.851	0.902	0.666	0.810	0.629	0.827	0.842	0.980	0.753	0.072
残差网络	0.903	0.750	0.745	0.570	0.831	0.836	0.904	0.676	0.806	0.769	0.160
FT-Transformer（Gorishniy 等人，2021 年）	0.796	0.618	0.863	0.860	0.903	0.681	0.819	0.599	0.957	0.746	0.073
T2G-Former（Yan 等人，2023 年）	0.843	0.852	0.900	0.695	0.812	0.627	0.841	0.849	0.980	0.754	0.069
SSL(RQ)+微调	0.904	0.713	0.816	0.639	0.837	0.851	0.904	0.681	0.978	0.757	0.073
SSL(MaskXent)+微调 SSL(ValueRecon)+微	0.816	0.631	0.836	0.848	0.902	0.725	0.815	0.628	0.978	0.753	0.071
调 SSL(MaskXent+ValueRecon)+微调									0.978	0.753	0.072
									0.978	0.752	0.071
我们的——SSL（BinRecon）+线性评估/微调	0.843	0.857	0.910	0.774	0.817	0.648			0.982	0.750	0.068

D. 有待讨论的其他结果

D.1. 分位数和等宽分箱方法的比较

我们发现分组对于成功实施分箱任务至关重要。除了基于分位数的分箱之外，我们还可以操纵等宽分箱。在这里，我们试验哪种方法对分位数和固定大小之间的分箱更有利。我们测试了相同候选集的等宽分箱数量，并在验证性能与基于分位数的分箱性能最佳时比较测试性能。

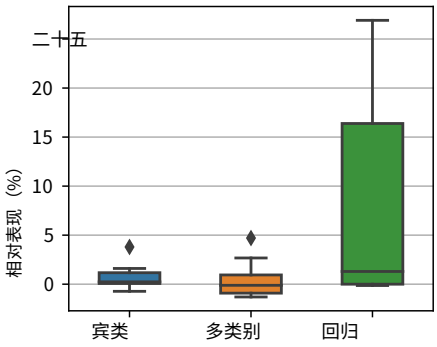
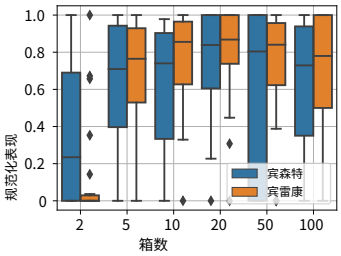


图 5：当我们将分箱方法从分位数改为等宽分箱时，相对性能。当值为正时，基于分位数的分箱比等宽分箱更好。当值为负时，反之亦然。特别是，对于回归任务，基于分位数的分箱比等宽分箱好得多。

结果如图 5 所示。在 25 个数据集中，等宽分箱对 3 个数据集（PH、HE、MNIST）表现出更好的性能，最高可达 0.6%，两种分箱方法对 2 个数据集（OT、AL）表现出相当的性能。对于其他 20 个数据集，基于分位数的分箱表现出更好的性能。特别是，对于回归任务，我们发现当我们将分箱方法从分位数更改为固定大小时，性能最多会下降 27%。最后，我们得出结论，基于分位数的分箱在各种数据集上始终能产生良好的表示。

D.2. 箱体数量与下游任务性能之间的依赖关系

我们研究了 BinXent 和 BinRecon 的箱数与下游任务性能之间的关系，无需输入转换。由于数据集之间的性能范围相差很大，我们使用每个数据集的最佳和最差情况对性能进行归一化。这种方法使我们能够在具有不同范围和不同评估指标的数据集之间对性能指标进行归一化。具体来说，我们评估了六个箱数不同的模型（2、5、10、20、50、100）中的最佳和最差性能，同时保持损失函数（BinXent、BinRecon）一致。因此，归一化比例将最佳性能情况设置为 1，将最差性能情况设置为 0。



图古尔 埃义贡里斯分析 锡所以 编辑英文恩科b 埃箱子数量和下游任务性能之间。

D.3. 第 6 部分的结果

表 11：分箱各个组件的消融测试结果。

订购	标准化	分组	改进	恶化
是的	是的	是的	- (*基线*)	- (*基线*)
不	是的	是的	1（+4.70%）	12（−4.05%
是的	不	是的	1（+5.21%）	15（−6.85%
是的	不	不	3（+1.95%）	12（−5.83%
不	不	不	-	18（−6.02%

表 12：各种 SSL 方法的分箱回归任务性能。我们提供了 BinRecon 案例基线的相对误差。对于所有情况，误差至少增加了 38%。因此，当我们没有明确提供借口目标时，可以从原始输入中实现分箱指标，但无法在结果表示中使用。

掩蔽	掩蔽值	目标	相对误差增加（%）
错误的	-	宾雷康	*（基线）* 0
错误的	-	价值侦察	49.579
真的	宪法。	MaskXent	82.922
真的	宪法。	价值侦察	38.444
真的	宪法。	MaskXent+ValueRecon	68.344
真的	随机的	MaskXent	111.708
真的	随机的	价值侦察	38.135
真的	随机的	MaskXent+ValueRecon	60.016
		平均的	66.285

D.4. 第 6.6 节的附加结果

表 13：使用分箱作为数据增强（随机量化，RQ）与使用分箱来定义输出标签（我们的）时表格 SSL 的微调性能比较。

训练方法	你好 ↑	肺动脉高压 ↑	操作系统 ↑	邮局 ↑	一氧化碳 ↑	通用电气 ↑	画外音 ↑	艾尔 ↑	他 ↑	MNIST ↑	加州 ↓	何 ↓	财务信息 ↓
RQ（Wu 等人，2023 年）	00.717 ±0.002	0.736 ±0.005	0.896 ±0.002	0.886 ±0.004	0.969 ±0.000	0.690 ±0.007	0.719 ±0.003	0.959 ±0.000	0.379 ±0.001	0.984 ±0.001	0.475 ±0.002	3.159 ±0.030	10398.616 ±28.659
我们的	0.737 ±0.001	0.764 ±0.008	0.897 ±0.000	0.904 ±0.003	0.971 ±0.000	0.720 ±0.002	0.728 ±0.001	0.966 ±0.001	0.388 ±0.001	0.986 ±0.000	0.464 ±0.001	2.989 ±0.015	9757.950 ±210.751

E. 附加说明

E.1. 对无信息特征的影响

Müller 等人 (2021) 和 Stewart 等人 (2023) 证明，将离散化纳入深度网络的目标函数不仅可以提高训练效率，而且被证明是一种理论上合理的建模任何分布的方法。这凸显了分箱在增强神经网络性能方面的巨大潜力。我们认为，分箱的一个关键优势是它能够将数据集简化为*电视*每个特征具有不同的值，从而在所有特征之间创建相等的集合。此属性有助于防止无信息特征（与任务标签具有低互信息但可能由于方差或大量唯一值而具有高熵的特征）对训练过程产生主导影响。

在使用直接重建损失应用表格 SSL 的场景中，神经网络可能会无意中更多地关注以高变异性（频率）或唯一值计数为特征的特征。这种现象在表格数据中更为明显，如最近的研究（Beyazit 等人，2023；Cherepanova 等人，2024）所指出的那样，这表明训练可能会偏向这些高频率但信息量较少的特征。通过在 SSL 期间用 bin 索引替换输出标签，我们的方法明确地限制了所有特征以相同的频率或相同的变异性回归 SSL 输出。因此，它有效地避免了任何特定特征在 SSL 期间占主导地位，确保这些不具信息量的特征不会掩盖有意义的表示的学习。