arXiv:2603.02840v1 [cs.LG] 3 Mar 2026

Ada通过数据混合 pting 时间序列基础模型

Thomas L. Lee [1] **Edoardo M. Ponti** [1] **Amos Storkey** [1]

摘要

时间序列基础模型在零样本预测中日益流行。然而，对于预训练集未能完全覆盖的新时间序列领域，性能可能会下降。因此，当从业者关注一个新领域并能够获取一组相关数据集时，便会出现这样一个问题：如何以最佳方式微调时间序列基础模型，以改进零样本预测？针对此类问题的典型做法，是在所有数据集上微调一个 LoRA 模块，或分别在每个数据集上进行微调。在每个数据集上单独微调一个模块，可以通过为不同的时间序列情境选择不同的逐数据集模块组合，使时间序列基础模型针对不同类型的数据分布实现专门化。然而，我们发现，使用逐数据集模块可能并非最优，因为一个时间序列数据集可能包含来自多种分布类型的数据，即子领域。这可能是由于分布漂移，或时间序列的不同维度具有不同分布所致。因此，我们提出*MixFT*，它利用贝叶斯混合模型重新划分数据，将其分成最能代表数据中所含子领域的集合，并分别在每个集合上进行微调。这种对数据的重新划分确保每个集合更加同质，从而使微调后的模块聚焦于特定子领域。我们的实验表明，MixFT 的表现优于逐数据集方法，也优于在全部数据上微调单个模块的做法。这表明，通过重新划分数据以表征子领域，我们可以更好地专门化时间序列基础模型，从而改进零样本预测。

1. 引言

时间序列基础模型（TSFM）在预测领域日益流行（Miller 等人，2024年；Ye 等人，2024年；Liang 等人，2024年）。其关键原因之一是具备零样本预测能力（Ansari 等人，2024年；Woo 等人，2024年；Darlow 等人，2024年）。这使得即使缺乏足够的历史数据、专业知识或计算资源，无法针对感兴趣的特定时间序列拟合模型，也能生成预测结果（Mercier 等人，2021年）。然而，尽管某个特定时间序列的历史数据可能不多，从业者通常可以获取其他相关时间序列。因此，一个问题随之而来：如何利用这些其他相关时间序列，最好地适配时间序列基础模型，从而提升其在感兴趣时间序列上的零样本性能？当感兴趣的时间序列来自时间序列基础模型预训练集以外的领域时，这个问题尤为重要。这是因为已有研究发现，对于这类时间序列，零样本性能可能会大幅下降（Toner 等人，2025年；Diao 等人，2024年；Aksu 等人，2024年），因此需要进行时间序列基础模型适配，才能获得有用的预测结果。

给定一组时间序列数据集，有多种潜在方法可以对时间序列基础模型进行适配。本文重点关注参数高效微调（PEFT）方法，这是因为此类方法在大语言模型中表现出色（Han 等人，2024年），近来也应用于时间序列基础模型（Gupta 等人，2024b；a）。最直接的方法是在所有可用的微调数据集上微调 LoRA 适配器（Hu 等人，2021年），这称为共享微调（Prabhu 等人，2023年）。通过考察大语言模型微调领域的相关研究，我们发现，另一种常见的跨多个数据集进行微调的方法，是在每个数据集或每组数据集上分别训练 LoRA 模块（Ostapenko 等人，2024年）。这些方法统称为逐数据集方法 （Ostapenko 等人，2024年；Chronopoulou 等人，2023年）。逐数据集方法的优势在于，每个 LoRA 模块都可以专门对其训练所用数据集中出现的数据分布进行建模，而研究表明，这有助于提升性能（Ponti 等人，2023）。

然而，时间序列往往存在分布偏移，而且多变量时间序列数据集的不同维度可能属于差异显著的分布（Zhang 等人，2024年；Kim 等人，2025年），即子领域。因此，我们发现，与使用数据集标签相比，还有更好的数据划分方式：将每个子领域的数据划分到不同的数据分区中。更具体地说，我们观察到，可以将不同时间序列数据集中的数据分布划分为一组子分布。在这里，子分布指一种特定的抽象时间序列数据类型，例如不同的季节性模式

[1]爱丁堡大学信息学院。通讯请联系：Thomas L. Lee < t.l.lee-1@sms.ed.ac.uk>。

预印本。*2026*年*3*月*4*日。

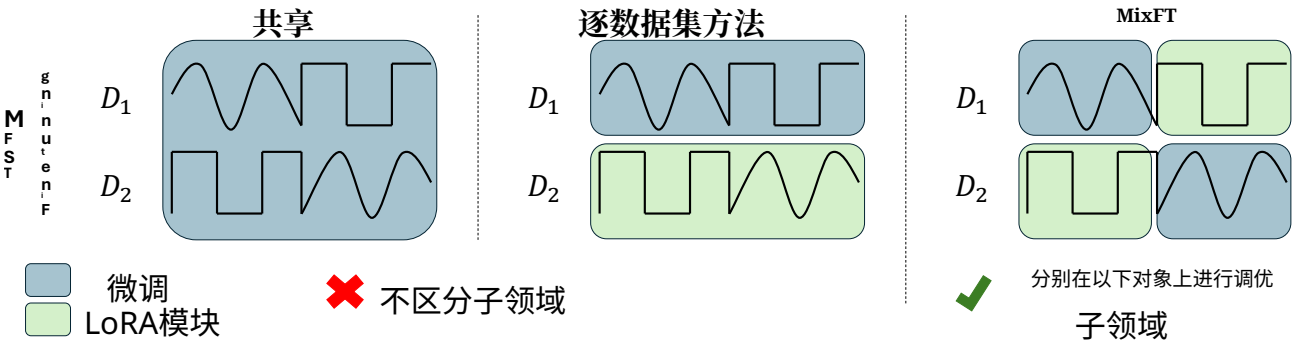


图 *1*。 **MixFT 根据微调数据集中的不同子领域，识别并训练独立的 LoRA 模块。** 这不同于以往的方法：前者使用数据集边界分别微调 LoRA 模块（逐数据集方法），后者则在全部数据上训练单个 LoRA 模块（共享）（Ostapenko 等人，2024年）。通过在子领域上训练独立的 LoRA 模块，MixFT 旨在使每个 LoRA 的训练数据更加同质化，从而得到更加专门化且一致的 LoRA 模块。这也应有助于在进行零样本预测时，更好地识别应使用哪些 LoRA 模块。

或尖峰程度。我们将每个子分布称为一个子领域。本文旨在探索这种方法：识别数据的各个子领域，并针对它们分别训练 LoRA 模块，而不是按数据集进行训练。

我们提出了一种方法*MixFT*，它能够识别微调数据中比逐数据集划分更合理的划分方式，并据此进行微调。MixFT 利用贝叶斯混合模型（Murphy，2022年）学习并识别子领域，从而根据子领域而非数据集边界划分数据。随后，MixFT 为每个子领域训练一个 LoRA 模块。这确保每个 LoRA 模块专门对应一个子领域，从而简化学习过程，因为用于训练每个 LoRA 模块的数据更加同质化。通过使用贝叶斯混合模型，我们还能够在进行零样本预测时识别时间序列上下文所属的子领域，进而确定用于预测的 LoRA 模块。这样可以确保用于微调的数据与待预测数据处于（子）分布之内。我们在实验中发现，MixFT 表现良好，优于逐数据集微调方法，也优于在全部数据上统一进行微调的方法。这表明，相较于在数据集而非子领域上拟合 LoRA 模块，后者由于训练数据更加一致，能够实现更好的专门化，从而带来更准确的预测结果。

本研究有以下三项主要贡献：

1. 发现为零样本预测微调专用 LoRA 模块时，数据集划分往往并非最优。2. 提出 MixFT，它利用贝叶斯混合模型，根据数据的子领域确定更优的数据划分，以用于微调。贝叶斯混合模型还使得在进行零样本预测时，能够识别时间序列上下文所属的子领域，从而选择在与该上下文属于同一子分布的数据上训练的 LoRA 模块

已在与上下文相同的子分布数据上进行训练。

3. 研究了微调方法在时间序列基础模型上的表现，结果表明 MixFT 表现良好，而与完全不进行微调相比，某些逐数据集方法表现较差。

2. 相关工作

本文研究时间序列基础模型（TSFM）。当前的 TSFM 通常是大型 Transformer 架构，在大量时间序列数据上训练而成（Rasul 等人，2023年；Das 等人，2024b；Liang 等人，2024年）。例如，Chronos Bolt（Ansari 等人，2024年）是一种基于 T5架构的 TSFM，训练所用的数据实例约有 1000 亿个。另一类流行的 TSFM 是 Moirai模型（Woo 等人，2024年），采用掩码编码器架构。尽管 TSFM 日益普及，但与大语言模型等其他类型的基础模型相比，目前关于微调方法的研究仍然很少（Miller 等人，2024年；Lee 等人，2025年）。例如，针对 TSFM 探索参数高效微调的研究还很少（Gupta 等人，2024b；a）。此外，大多数关于 TSFM 微调的研究都着重于提升其在当前时间序列上的表现（Ekambaram 等人，2024年）。与此不同，本文着重于在一组数据集上进行微调，以提升模型对相关时间序列的零样本预测能力。

关于如何微调时间序列基础模型以提升零样本性能的研究相对较少，但这一问题在大语言模型文献中已得到充分研究（Han 等人，2024年）。例如，已有大量研究探讨如何使用参数高效微调进行零样本预测（Ostapenko 等人，2024年）。一种通用方法是在每个微调数据集上训练一个 LoRA 模块，然后将它们线性组合，

以进行零样本预测（Chronopoulou 等人，2023年）。此外，还有基于 RAG、基于提示以及基于上下文的方法，用于适配基础模型以提升零样本性能（Gao 等人，2023年；Liu 等人，2023年；Dong 等人，2022年）。然而，这些方法通常专为大语言模型设计，并且需要经过专门训练的时间序列基础模型才能使用（Yang 等人，2025年；Ning 等人，2025年；Das 等人，2024a）。这也是我们采用 LoRA 模块作为 MixFT 基础的原因之一，因为我们希望 MixFT 不受时间序列基础模型选择的限制。更广泛地说，也有研究通过改善不同数据实例梯度之间的干扰，来选择数据子集以提升泛化能力（Maharana 等人，2024年；Swayamdipta 等人，2020年；Gadre 等人，2023年）。

3. 预备知识：时间序列预测

为介绍 MixFT 和我们的实验，我们首先需要夯实时间序列预测以及时间序列基础模型微调方面的基础。时间序列由一系列带有时间索引的向量组成 $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_t, \ldots, \boldsymbol{v}_T$ 其中 $\boldsymbol{v}_t \in \mathbb{R}^c$ 表示时间序列在时间步 t的取值。通常，时间序列的取值可以是多变量的，即 $c > 1$，我们的实验也会考察这类时间序列。不过，为了简化且不失一般性，下文及第4节中，我们只介绍单变量情形，即 $c = 1$。

在微调和零样本预测中，我们考虑长度为L的时间序列上下文窗口，

$$\boldsymbol{x}_{i,j} = [\boldsymbol{v}_{i,t}]_{t=j}^{j+L}, \tag{1}$$

其中 i 用于标识时间序列。这种窗口化是文献中的标准做法（Darlow 等人，2023年；Lim 和 Zohren，2021年；Nie 等人，2022年）。对于每个可用于微调的 M 条时间序列，我们构建一个窗口化时间序列数据集 $D_i = [\boldsymbol{x}_{i,j}]_{j=1}^{N_i}$，其中 $i \in [1, \ldots, M]$。这里 N_i表示该时间序列的上下文窗口数量，$N_i = T_i - L$其中 T_i 表示时间序列的长度。随后，微调方法将使用这些窗口化时间序列来调整时间序列基础模型。

零样本预测是指向时间序列基础模型提供一个来自 $\boldsymbol{x}$一条新时间序列的上下文，而该序列在微调期间未曾出现（Aksu 等人，2024年）。其任务是生成长度为$\hat{}\boldsymbol{y}$ 的预测值 H，用于预测时间序列在观测到 $\boldsymbol{x}$ 之后的未来值。评估时，会将预测结果与该预测时间步对应的时间序列真实值进行比较，记作 $\boldsymbol{y}$。使用窗口的原因在于，它能将时间序列数据转换为机器学习（ML）中通用的格式，即协变量$\boldsymbol{x}$ 和目标值 $\boldsymbol{y}$，从而能够采用标准的机器学习方法（Lim 和 Zohren，2021年；Toner 和 Darlow，2024年）。

3.1. 符号说明

下文列出了本文中使用的符号。

- L：上下文长度（输入中的时间步数）。
- H：预测跨度（要预测的未来时间步数）。
- M：微调时间序列数据集的数量。
- N_i：时间序列 i 中上下文窗口的数量。
- K：MixFT 使用的组件数量
- $\boldsymbol{x}_{i,\ j}$：来自第 i个时间序列的第 j个上下文窗口，$\boldsymbol{x}_{i,\ j} \in \mathbb{R}^L$。
- $\boldsymbol{z}_{i,\ j}$：嵌入的上下文窗口，$\boldsymbol{z}_{i,\ j}$ =时间序列基础模型$_{embed}(\boldsymbol{x}_{i,\ j})$
- $\boldsymbol{y}$，$\hat{}\boldsymbol{y}$：分别为目标值和预测值，$\boldsymbol{y}$，$\hat{}\boldsymbol{y} \in \mathbb{R}^H$ 对于某个上下文窗口 $\boldsymbol{x}$。

4. MixFT：使用数据集混合进行微调

在本研究中，我们提出*MixFT*，一种面向零样本预测的时间序列基础模型微调方法。MixFT 与以往工作的不同之处在于，它会将作为微调数据提供的数据集重新划分成与数据底层子领域更为契合的集合。为实现这一点，MixFT 对微调数据拟合贝叶斯高斯混合模型（高斯混合模型），以识别这些子领域。随后，MixFT 将数据拆分为归属于各个混合成分的集合，并在每个集合上分别微调一个独立的 LoRA 模块。这构成了 MixFT 的微调阶段。在执行零样本预测时，给定时间序列上下文，MixFT 首先识别其最可能归属的子域/混合成分。然后，MixFT 使用针对该子域训练的特定 LoRA 模块来构建预测值。MixFT 的微调阶段与零样本预测将在下文详细描述，并在图2中以示意图展示——算法描述亦见附录B.1。

4.1. 微调

MixFT 微调分为两个阶段：**a)**拟合贝叶斯高斯混合模型，以及**b)**在贝叶斯高斯混合模型给出的数据分区上拟合 LoRA 模块。为执行微调，MixFT 接收一组上下文长度为 L 的窗口化时间序列（详见第 3节）。更形式化地说，MixFT 接收 $D_i = [\boldsymbol{x}_{i,j}]_{j=1}^{N_i}$，其中 $i \in [1, \ldots, M]$。贝叶斯高斯混合模型定义在时间序列基础模型的嵌入空间中，以避免使用更复杂的模型。因此，MixFT 首先使用时间序列基础模型嵌入数据（时间序列基础模型$_{embed}$），从而得到嵌入数据集

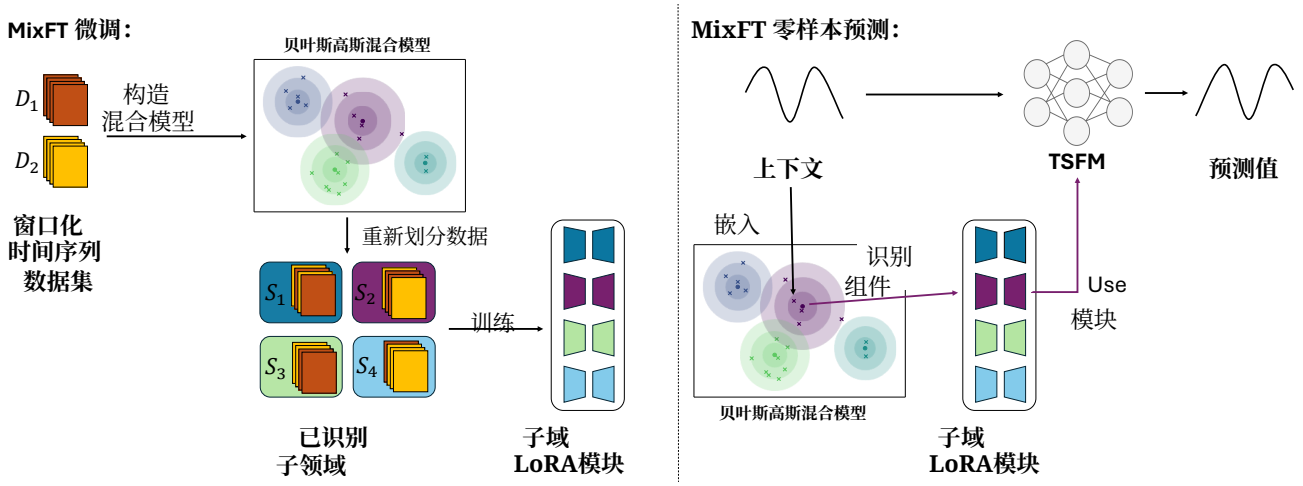


图 2. **MixFT 概览**。图的左侧展示了 MixFT 如何微调用于零样本预测的 LoRA 模块。首先，它会识别微调数据集中的子领域。然后，按子领域重新划分数据。最后，在每个子领域的数据上分别训练一个 LoRA 模块。这样可以确保训练得到的 LoRA 模块专门针对子领域而非数据集进行预测，这与以往的工作不同（Ostapenko 等人，2024年）。图的右侧展示了 MixFT 如何构建预测值。具体做法是识别上下文所属的子领域，然后使用该子领域的 LoRA 模块进行预测。这样可以确保时间序列基础模型利用微调数据所提供的该子领域知识。

$D_i^Z = [\boldsymbol{z}_{i,j} = \text{时间序列基础模型}_{embed}(\boldsymbol{x}_{i,\ j})]_{j=1}^{N_i}$。现在，我们可以定义 MixFT 使用的贝叶斯高斯混合模型，它包含 K 个混合成分，

$$\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \sim \text{NIDW}(\boldsymbol{m}, \kappa, \nu, \boldsymbol{W}) \quad \text{for } k \in [1, \ldots, K], \tag{2}$$

$$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha}), \tag{3}$$

$$c_{i,j} \sim \text{Cat}(\boldsymbol{\pi}), \tag{4}$$

$$z_{i,j} | c_{i,j} \sim \mathcal{N}(\boldsymbol{\mu}_{c_{i,j}}, \boldsymbol{\Sigma}_{c_{i,j}}), \tag{5}$$

其中 $\boldsymbol{m}$，κ，ν，$\boldsymbol{W}$和 $\boldsymbol{\alpha}$ 均为超参数，NIDW 表示正态-逆对角威沙特分布（逆对角威沙特分布由各维度相互独立的逆卡方分布构成（Murphy，2022年））。使用 NIDW 先验后，每个混合成分的协方差$\boldsymbol{\Sigma}_k$ 均为对角矩阵，从而减少参数数量并降低过拟合的可能性（见附录 D.3）。

我们使用均值场变分推断拟合贝叶斯高斯混合模型（Bishop，2006，第 10.2 节），并在变分分布中假设潜变量 $c_{i,j}$ 与 $\boldsymbol{m}$、$\boldsymbol{W}$ 和 $\boldsymbol{\pi}$ 相互独立。通过执行变分推断，MixFT 可以使用后验变分分布，

$$q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \text{NIDW}(\boldsymbol{m}_k^{\text{VAR}}, \kappa_k^{\text{VAR}}, \nu_k^{\text{VAR}}, \boldsymbol{W}_k^{\text{VAR}}), \tag{6}$$

$$q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\alpha}^{\text{VAR}}), \tag{7}$$

由估计值定义 $\boldsymbol{m}_k^{\text{VAR}}$，$\kappa_k^{\text{VAR}}$，$\nu_k^{\text{VAR}}$，$\boldsymbol{W}_k^{\text{VAR}}$和$\boldsymbol{\alpha}^{\text{VAR}}$。我们采用变分推断，因为与最大似然估计（即 K均值）相比，它更加稳定（Murphy，2022年），从而确保找到的混合模型较少依赖其初始化。此外，我们还探索了使用更复杂概率模型的可能性，尤其是嵌入式主题模型（Dieng 等人，2020），但发现与贝叶斯高斯混合模型相比，这类模型更容易过拟合（见附录 D.3）。

在获得后验变分分布后，我们现在可以将数据重新划分为与数据潜在子域相对应的集合。为此，我们使用后验预测分布为每个数据点赋予一个子领域标签，

$$\hat{c}_{i,j} = \underset{c \in [1,\ldots,K]}{\arg\max} [p(c | \boldsymbol{z}_{i,j}, \phi_c^{\text{VAR}}, \boldsymbol{\alpha}^{\text{VAR}})]. \tag{8}$$

其中 $\phi_c^{\text{VAR}} = \{\boldsymbol{m}_c^{\text{VAR}}, \kappa_c^{\text{VAR}}, \nu_c^{\text{VAR}}, \boldsymbol{W}_c^{\text{VAR}}\}$。然后，我们将数据划分为子领域数据集，

$$\mathcal{S}_k = \{\boldsymbol{x}_{i,j} | \hat{c}_{i,j} = k\}, \tag{9}$$

对于 k，范围从 1 到 K。

随后，集合 $\mathcal{S}_k$ 将用于 MixFT 微调的第二阶段：训练子域 LoRA 模块。为此，我们只需在每个集合 $\mathcal{S}_k$ 上分别训练一个 LoRA 模块。这样，MixFT 便拥有一个包含 K 个 LoRA 模块的模块库，每个模块都针对数据中发现的特定子域进行了专门优化。

MixFT 以这种方式拟合 LoRA 模块，原因有几个。首先，这确保每个 LoRA 组件只需建模其所属子域的特征，从而简化任务。其次，这降低了联合建模不同且相互竞争的子域/任务时可能出现破坏性干扰的概率，正如 Liu 等人（2021年所示。最后，通过重新划分数据，子域

MixFT 创建的专家 LoRA 模块缩小了零样本预测中的泛化差距（Mehta& Harchaoui，2025年；He et al.，2024年）。这是因为，对于给定的子域，与使用按数据集LoRA时相比，子域 LoRA 训练所用的数据与该子域在零样本预测中使用的数据之间的分布偏移更小。原因在于，按数据集LoRA是在来自不同子域的各种数据上训练的，因此，相比 MixFT 的子域 LoRA 模块，任何按数据集LoRA 与从某个子域采样的新时间序列上下文之间的分布差异都更大。泛化差距的缩小带来了更好的零样本预测性能，我们的实验结果证实了这一点（见第5节）。

4.2. 零样本预测

进行零样本预测时，我们会获得来自一个相关但不同于微调数据的时间序列的时间序列上下文 $\boldsymbol{x}$。假设该时间序列大致包含与微调数据相同的潜在子域，便可认为它们相关，否则就没有理由进行微调。基于这一点，MixFT 的首要任务是识别其所属的子域 $\boldsymbol{x}$。通过使用贝叶斯高斯混合模型，我们可以直接利用拟合后的后验预测分布给出的最可能子域来完成这一任务，

$$\hat{c} = \underset{c\in[1,\dots,K]}{\arg\max}\,[p(c|\boldsymbol{z}, \phi_c^{\text{VAR}}, \boldsymbol{\alpha}^{\text{VAR}})] \tag{10}$$

其中$\boldsymbol{z} = \text{TSFM}_{embed}(\boldsymbol{x})$和$\phi_c^{\text{VAR}} = \{\boldsymbol{m}_c^{\text{VAR}}, \kappa_c^{\text{VAR}}, \nu_c^{\text{VAR}}, \boldsymbol{W}_c^{\text{VAR}}\}$是 MixFT 的变分估计。现在我们已估计出其所属的子域 $\boldsymbol{x}$，为了给出预测值，我们使用第$\hat{c}$个 LoRA 模块，该模块对第$\hat{c}$个子域进行建模。MixFT 将 LoRA 模块与 TSFM 及输入 $\boldsymbol{x}$结合使用，生成预测值$\hat{y}$。

通过这种方式进行预测，我们旨在使 $\boldsymbol{x}$ 更符合所选 LoRA 模块微调所用的数据分布，二者属于同一子域。这样一来，LoRA 模块便能利用所学到的该子域特征，帮助时间序列基础模型给出更准确的预测值。我们采用硬性子域分配方式，即针对给定上下文，对后验预测分布取 argmax，这主要基于两个原因：**a**）使用来自不太可能的子域的 LoRA 模块时，给出的预测值可能不准确，因为该上下文对于该子域而言属于分布外（OOD）（参见附录D.5.1）；**b**）在实验中，我们发现 MixFT 通常能确定给定上下文来自哪个子域（参见附录D.5.2）。因此，MixFT 只采用最可能子域对应的 LoRA 给出的预测值；在我们的实验中，该子域通常比其他子域的可能性高得多，从而避免了使用来自不太可能子域的 LoRA 所产生的不准确分布外预测值

。此外，以往的方法通常会在事后选择 LoRA 模块或其混合方案。这可能导致微调方法与选择方法不匹配。然而，MixFT 使用概率模型克服了这一问题，因为它直接依据 LoRA 模块的微调方式，来决定如何为零样本预测选择这些模块。

5. 实验

5.1. 实验设置

基准 为了分析 MixFT 的性能，以及更一般地，将微调数据重新划分为更同质的集合是否能带来性能收益，我们进行了一系列实验。在实验中，我们使用了来自 Cloud 和 Gift-Eval 基准的数据集（Toner 等人，2025年；Aksu 等人，2024年）。这些基准已知对当前时间序列基础模型而言难以实现良好的零样本预测，并且包含许多密切相关的数据集。

我们将每个基准中的数据集划分为两组：微调数据和评估数据。具体而言，我们用于微调数据的数据集为：CloudD3、CloudD4、BizITObs-Service、BitBrains Fast Storage（每小时）、M4-Hourly 和 M4-Weekly（Toner 等，2025；Palaskar 等人，2024年；Shen 等人，2024年；Godahewa 等人，2021年）。用于评估的数据集为：CloudD1、CloudD2、BizITObs-L2C、BizITObs-App、US-Births、M4-Daily、M4-Monthly、M4-Quarterly、ETTh2 和 ETTm2（Toner 等，2025；Palaskar 等人，2024年；Godahewa 等人，2021年；Zhou 等人，2021年）。需要指出的是，我们对每个评估数据集中的全部数据进行评估。这不同于以往工作中有时使用的预定义测试集划分（Aksu et al., 2024）。在实验中，我们使用了上下文长度 $L = 520$ 和预测长度 $H = 30$，这与以往工作中的常用设置一致（Ansari et al., 2024；Ekambaram 等人，2024年）。我们采用这一实验设置结构的原因在于，它模拟了我们所针对的实际应用场景：在一系列相关数据集上进行微调，以提升时间序列基础模型在从业者所关注的时间序列上的零样本性能。

我们评估的时间序列基础模型包括 Chronos Bolt（small）和 Moirai-1.1-R（small）（Ansari 等人，2024年；Woo 等人，2024年）。我们选择这些时间序列基础模型，是因为它们具有最先进的性能，且未在我们使用的任何微调或评估数据上进行预训练（Aksu 等人，2024年）。更多实验细节见附录C。

微调方法 我们将 MixFT 与几种旨在提升零样本性能的其他微调方法进行比较。我们首先将其与最直接的方法进行比较：共享 微调，即使用所有可用的微调数据训练单个 LoRA 模块（Ostapenko 等人，2024年）。

表 1. **微调方法的平均绝对缩放误差。** 该表展示了 Chronos Bolt 和 Moirai-1.1-R 时间序列基础模型在标准时间序列数据集上进行零样本预测时，各种微调方法的平均绝对缩放误差表现。平均绝对缩放误差越低越好，我们还提供了训练运行的标准误。值得注意的是，用于微调的时间序列与评估所用的时间序列相关，但并不相同。对于每个数据集和时间序列基础模型，我们分别用粗体和下划线标出表现最佳和次佳的方法。结果显示，从平均排名以及加粗和加下划线的结果数量来看，MixFT 表现最佳。

		方法						
	数据集	Base	共享	μ 数据集	Arrow 数据集	Poly	MBC	MixFT（我们的方法）
Chronos Bolt	CloudD1	1.511	$1.553_{\pm 0.024}$	$1.580_{\pm 0.016}$	$\mathbf{1.352_{\pm 0.006}}$	$\underline{1.463_{\pm 0.012}}$	$1.810_{\pm 0.266}$	$1.485_{\pm 0.008}$
	CloudD2	1.572	$1.342_{\pm 0.004}$	$1.614_{\pm 0.029}$	$\underline{1.341_{\pm 0.005}}$	$\mathbf{1.320_{\pm 0.004}}$	$1.649_{\pm 0.211}$	$1.343_{\pm 0.004}$
	BizITObs-L2C	3.116	$\underline{3.101_{\pm 0.011}}$	$3.624_{\pm 0.084}$	$3.680_{\pm 0.028}$	$3.124_{\pm 0.001}$	$3.170_{\pm 0.009}$	$\mathbf{3.083_{\pm 0.006}}$
	BizITObs-App	3.494	$\underline{1.060_{\pm 0.020}}$	$3.691_{\pm 0.117}$	$5.002_{\pm 0.197}$	$1.173_{\pm 0.006}$	$1.419_{\pm 0.346}$	$\mathbf{0.989_{\pm 0.024}}$
	US-Births	1.014	$\underline{0.967_{\pm 0.007}}$	$1.175_{\pm 0.003}$	$1.212_{\pm 0.003}$	$0.979_{\pm 0.010}$	$1.041_{\pm 0.005}$	$\mathbf{0.942_{\pm 0.002}}$
	M4-Daily	**7.175**	$7.254_{\pm 0.016}$	$8.415_{\pm 0.088}$	$8.376_{\pm 0.037}$	$7.249_{\pm 0.004}$	$7.530_{\pm 0.121}$	$\underline{7.238_{\pm 0.005}}$
	M4-Monthly	7.670	$\mathbf{7.580_{\pm 0.042}}$	$9.032_{\pm 0.506}$	$10.658_{\pm 0.141}$	$7.735_{\pm 0.013}$	$8.173_{\pm 0.129}$	$\underline{7.638_{\pm 0.014}}$
	M4-Quarterly	$\underline{8.178}$	$8.269_{\pm 0.037}$	$10.916_{\pm 0.187}$	$9.089_{\pm 0.052}$	$8.253_{\pm 0.037}$	$8.275_{\pm 0.123}$	$\mathbf{8.135_{\pm 0.010}}$
	ETTh2	**1.383**	$1.388_{\pm 0.001}$	$1.414_{\pm 0.004}$	$1.431_{\pm 0.003}$	$\underline{1.386_{\pm 0.001}}$	$1.390_{\pm 0.002}$	$1.387_{\pm 0.001}$
	ETTm2	**0.723**	$0.731_{\pm 0.003}$	$0.787_{\pm 0.002}$	$0.785_{\pm 0.003}$	$0.731_{\pm 0.001}$	$0.740_{\pm 0.003}$	$\underline{0.730_{\pm 0.001}}$
	平均排名 (↓)	2.9	3	6.3	5.6	$\underline{2.8}$	5.3	**2**
Moirai-1.1-R	CloudD1	$2.164_{\pm 0.001}$	$1.814_{\pm 0.016}$	$1.923_{\pm 0.015}$	$\mathbf{1.596_{\pm 0.006}}$	$2.183_{\pm 0.094}$	$2.079_{\pm 0.051}$	$\underline{1.813_{\pm 0.011}}$
	CloudD2	$2.522_{\pm 0.001}$	$1.832_{\pm 0.004}$	$1.844_{\pm 0.023}$	$\mathbf{1.506_{\pm 0.004}}$	$\underline{1.721_{\pm 0.032}}$	$2.052_{\pm 0.032}$	$1.774_{\pm 0.015}$
	BizITObs-L2C	$3.262_{\pm 0.001}$	$\underline{3.166_{\pm 0.004}}$	$4.420_{\pm 0.060}$	$3.903_{\pm 0.028}$	$3.174_{\pm 0.010}$	$3.191_{\pm 0.013}$	$\mathbf{3.149_{\pm 0.001}}$
	BizITObs-App	$5.127_{\pm 0.001}$	$3.011_{\pm 0.084}$	$5.313_{\pm 0.186}$	$5.926_{\pm 0.171}$	$\underline{1.574_{\pm 0.068}}$	$\mathbf{1.400_{\pm 0.056}}$	$2.762_{\pm 0.004}$
	US-Births	$1.641_{\pm 0.001}$	$1.663_{\pm 0.014}$	$2.139_{\pm 0.026}$	$1.859_{\pm 0.027}$	$\underline{1.618_{\pm 0.004}}$	$\mathbf{1.597_{\pm 0.005}}$	$1.647_{\pm 0.009}$
	M4-Daily	$\mathbf{7.740_{\pm 0.001}}$	$7.813_{\pm 0.007}$	$9.602_{\pm 0.114}$	$9.102_{\pm 0.061}$	$7.866_{\pm 0.016}$	$7.978_{\pm 0.084}$	$\underline{7.767_{\pm 0.003}}$
	M4-Monthly	$8.012_{\pm 0.001}$	$\underline{7.962_{\pm 0.013}}$	$10.645_{\pm 0.259}$	$9.988_{\pm 0.155}$	$8.173_{\pm 0.172}$	$8.077_{\pm 0.085}$	$\mathbf{7.876_{\pm 0.008}}$
	M4-Quarterly	$\mathbf{7.777_{\pm 0.002}}$	$\underline{7.975_{\pm 0.002}}$	$9.171_{\pm 0.068}$	$8.809_{\pm 0.035}$	$8.083_{\pm 0.039}$	$8.163_{\pm 0.033}$	$7.998_{\pm 0.018}$
	ETTh2	$1.498_{\pm 0.001}$	$\underline{1.475_{\pm 0.002}}$	$1.586_{\pm 0.005}$	$1.546_{\pm 0.004}$	$1.485_{\pm 0.002}$	$1.483_{\pm 0.003}$	$\mathbf{1.474_{\pm 0.001}}$
	ETTm2	$0.948_{\pm 0.001}$	$\underline{0.848_{\pm 0.001}}$	$0.890_{\pm 0.002}$	$0.868_{\pm 0.001}$	$0.866_{\pm 0.008}$	$0.861_{\pm 0.005}$	$\mathbf{0.846_{\pm 0.001}}$
	平均排名（↓）	4.3	$\underline{2.9}$	6.3	5.0	3.7	3.7	**2.1**

我们还将其与几种逐数据集方法进行比较： μ-数据集 (Chronopoulou 等人，2023年；Page-Caccia 等人，2023年），该方法在每个数据集上分别训练一个 LoRA 模块，并在进行零样本预测时使用这些 LoRA 模块的平均值；*Arrow-Dataset* (Ostapenko 等人，2024年），该方法使用表现良好的*Arrow* 零样本路由机制，根据给定的评估时间序列上下文对各数据集适配器进行加权；*Poly* (Prabhu 等人，2023年），这是一种端到端方法，在训练过程中会学习针对所用每个 LoRA 模块的各数据集重要性权重；以及*MBC* (Ostapenko 等人，2024年），该方法使用各数据集 LoRA 的统计信息聚合多个数据集

然后在每个聚合数据上训练一个 LoRA 模块。我们还将其与基线 基线 进行比较，即不对时间序列基础模型进行微调，而是使用其预训练权重进行零样本预测。对于 MixFT、MBC 和 Poly，我们使用$K = 2$ 个 LoRA 模块/数据划分，除非另有说明。在附录 D.1中，我们介绍了一种自动选择 K的方法，该方法使用验证集上的平均排名，从而验证了我们选择 $K = 2$的合理性。

所考察的每种微调方法都会调整一组 LoRA 模块（Hu 等人，2021年），但要确保这些模块不会

发生过拟合颇具挑战性（Gupta 等人，2024b；a）。[1] 为缓解这一问题，我们采用两种策略：**a)** MixUp 数据增强（Zhang 等人，2017年；Ansari 等人，2024年）以及**b)** 将预训练数据与新的微调数据一同用于再训练，这种方法称为经验回放（Ostapenko 等人，2022年；De Lange 等人，2021年）。采用这两种策略，旨在避免对特定微调数据过拟合，同时避免损害对预训练性能的保持。我们的目标是让时间序列基础模型专注于微调数据中呈现的数据分布，从而提升其在相关数据上的零样本性能。

指标 我们使用 平均绝对缩放误差 （MASE）来衡量预测结果的准确性（Hyndman 和 Koehler，2006）。给定预测结果 $\hat{\boldsymbol{y}}$、上下文 $\boldsymbol{x}$和目标值 $\boldsymbol{y}$，MASE 定义为

$$\text{MASE}(\hat{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{x}) = \frac{L-S}{H} \frac{\sum_{i=1}^{H} |\hat{y}_i - y_i|}{\sum_{i=1}^{L-S} |x_i - x_{i+S}|}, \quad (11)$$

其中 S 表示季节性。MASE 是一种归一化的平均绝对误差（MAE），其中使用上下文中朴素季节性预测器的 MAE 进行归一化。归一化有助于更好地汇总数值尺度和复杂程度各不相同的时间序列预测结果的表现。因此，MASE 常用于以往关于时间序列基础模型的研究中（Ansari 等人，2024年），这也是我们采用该指标的原因。

为了概述各微调方法的表现，我们在每张结果表中记录平均排名。平均排名 指某种方法在各个数据集上的名次的平均值。例如，平均排名为 2 表示该方法在各数据集上的平均名次为第二。我们采用平均排名，因为它对每个数据集一视同仁，因此不会受到不同数据集上各方法表现差异程度不同的影响。

5.2. 结果

表 1 展示了不同微调方法的比较结果。我们发现，MixFT 的表现最佳。表中展示了各评估数据集上的零样本性能，并分别对每个数据集和时间序列基础模型表现最佳和次佳的方法加粗及加下划线。根据平均排名以及获得最佳和次佳性能的次数，我们发现，MixFT 的表现优于其他任何微调方法。此外，我们发现，许多微调方法难以胜过不进行微调（基线）的情况。事实上，除 MixFT 外，对于表现更好的时间序列基础模型 Chronos Bolt，每种方法在至少3.01 个数据集上的平均绝对缩放误差都不如基线。MixFT 的表现则不如

[1]不过，这比对时间序列基础模型的所有参数进行微调更简单（Lee 等人，2025年）

表 *2*。 **对于使用 Arrow 路由的微调方法，Chronos Bolt 在所有数据集上的平均平均绝对缩放误差相较于基线（即不进行微调）的提升**。 我们同时给出了绝对值方面的平均提升（Avg. Imp）和相对值方面的平均提升（Avg. Rel. Imp.）。相对值的计算方式是：基线的平均绝对缩放误差除以相应微调方法的平均绝对缩放误差。我们发现，在测试时通过始终使用 Arrow 路由来控制 LoRA 模块的选择时，MixFT 学到的 LoRA 模块表现最佳。这表明，与按数据集划分相比，它能够找到更好的微调数据划分。

方法	平均提升（↓）	平均相对提升（↑）
Arrow-Datasets	0.71	0.90倍
MBC	−0.06	1.11倍
Arrow-MixFT	**−0.17**	**1.17倍**

至少比基线高 0.01 MASE，仅在 M4-Daily 上如此。这与大语言模型文献中的结果不同，在该领域，微调方法几乎总是优于基线（Ostapenko 等人，2024年），这表明时间序列基础模型要实现零样本性能，微调具有相当大的难度。

在表2 中，我们进一步分析 MixFT 找到的数据混合上进行微调与在数据集或数据集集合上进行微调之间的差异。为此，我们通过专门使用 Arrow 选择方法，控制测试时如何选择经过微调的 LoRA 模块。我们比较 MixFT 找到的 LoRA 模块（Arrow-MixFT）与按每个数据集训练一个 LoRA 模块（Arrow-Datasets）或按数据集集合训练（MBC）的情况。只使用 Arrow 选择/路由，可以确保方法之间的性能差异并非源于测试时选择得更好，而是源于训练出的 LoRA 模块更好。表2展示了这项实验的结果，表明在 MixFT 数据混合上训练的 LoRA 模块优于按单个数据集训练的模块。这说明，将微调数据重新划分为更符合数据底层子领域的集合后，MixFT 能够训练出更好的 LoRA 模块。

5.3 MixFT 学到了哪些混合

探索 MixFT 找到的不同数据集划分/混合很有价值，这有助于更好地理解其行为和性能。为此，我们在图 3 中绘制了随时间变化的各微调数据集中，MixFT 分配给第一个通道数据的混合成分。我们发现，两种混合成分都被频繁使用；对于 CloudD3/4、BizITObs-Service 和 M4-weekly 数据集，通常更倾向于使用第一个（紫色）成分，而对于 BitBrains Fast Storage 和 M4-Hourly，则更频繁地使用另一种成分。这表明不同数据集对混合成分的使用方式有所不同。此外，对于 CloudD4 和 BizITObs-Service

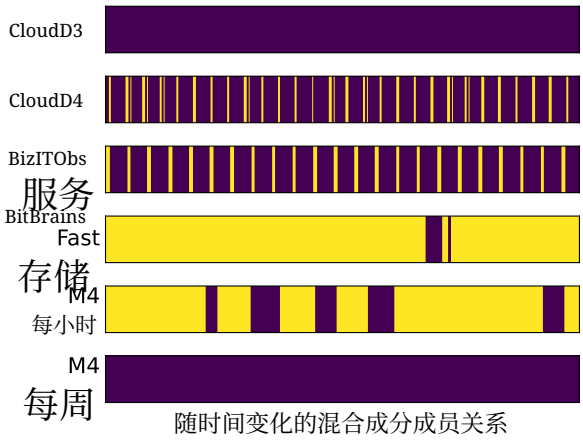


图3. **MixFT 微调数据集的混合成分成员关系。** 该图展示了每个微调数据集的第一个通道中，时间序列被学习为属于第一个混合成分（紫色区域）或第二个混合成分（黄色区域）的时间段。我们发现，有些通道只包含一个混合成分（CloudD3 和 M4-Weekly），大部分属于一个混合成分（BitBrains 和 M4-Hourly），或者呈现周期性模式（CloudD4 和 BizITObs-Service）。这些图表表明，MixFT 的贝叶斯混合模型能够从数据中发现合理的模式。此外，由于在给定数据集中能够发现两个子领域/混合成分，因此逐数据集方法无法发现 MixFT 的数据划分。

我们观察到混合分配呈现周期性模式。这表明 MixFT 的混合机制识别出了一些季节性模式/子领域。此外，在附录 D.5.1 中，我们考察了 MixFT 对混合成分分配的置信程度。我们发现，对于微调数据和评估数据，MixFT 都能高度确信地判断给定上下文应使用哪种混合成分。此外，在附录 D.5.3中，我们考察了被识别为属于不同子领域的时间序列上下文。我们发现，每个子领域都有鲜明的特征。例如，在上下文末尾，时间序列是否正在快速变化。

需要注意的是，MixFT 会对同一数据集使用两个混合成分（见图3），而逐数据集方法无法做到这一点。这展现了 MixFT 的灵活性，以及其识别同一数据集内多个子领域的能力。这是使用 MixFT 而非逐数据集方法的一项关键优势，能够带来更优的微调 LoRA 模块（见表 2）和更出色的零样本预测。

5.4. 消融实验

为了了解 MixFT 各个组件的作用，我们进行了多项消融实验。首先，我们考察 MixFT 在零样本预测时选择 LoRA 模块的方法。我们将其与使用 μ 和 Arrow 选择的方法进行比较，以及使用混合模型给出的概率的方法。结果见附录 D.2，表明 MixFT 的选择方法表现最佳。这是因为，除 CloudD1 外，MixFT 在所有数据集上的表现均为最佳或次佳；在 CloudD1 上排名第三，但平均排名最低。其次，我们考察使用不同概率模型重新划分数据，具体包括通过 K均值拟合的高斯混合模型（Murphy，2022年）和嵌入主题模型（Dieng 等人，2020）。该实验的结果和详情见附录 D.3。我们发现，MixFT 使用贝叶斯高斯混合模型的表现优于另外两种模型。具体而言，在除一个（M4-Daily）之外的所有数据集上，其表现均为最佳或次佳。这表明 MixFT 的贝叶斯高斯混合模型非常适合在时间序列数据中发现子领域。最后，我们在附录 D.1 和 D.4中分析了混合成分的数量。我们发现，在验证数据（见附录 D.1）和评估数据（见附录 D.4）上，使用不同数量的成分时，MixFT 的表现总体一致；在这两种情况下，使用 $K = 2$的表现最佳。这表明，所提出的选择 K 的方法能够找到合理的取值。

6. 结论

在本文中，我们提出了 MixFT，这是一种利用贝叶斯混合模型将微调数据划分为潜在子域的微调方法。通过这种方式，它可以针对每个子域训练一个 LoRA 模块，使其适应该子域的特征，同时避免受到其他子域数据的潜在干扰。这有助于提升零样本预测性能。这是因为，MixFT 可以利用在某个子域上微调得到的 LoRA 权重，预测该子域中的新数据。与每个 LoRA 模块都在多个子域上训练的方法（例如逐数据集方法）相比，这缩小了微调数据与测试数据之间的泛化差距。我们在实验中展示了 MixFT 的优势，也在附录A中列出了它的局限性。我们发现，MixFT 的零样本预测性能优于逐数据集方法，也优于在全部微调数据上训练单个 LoRA 模块的做法。因此，我们认为，MixFT 为未来的时间序列基础模型微调方法指明了一个潜在方向：应着眼于如何更好地划分现有微调数据，而不是将数据集划分视为固定不变。

影响声明

本文旨在改进时间序列预测。我们的工作可能会对社会产生诸多影响。不过，这些影响并非我们的工作所独有，而是机器学习整体领域共同面临的。因此，我们认为无需在此特别强调其中任何影响。

参考文献

Aksu, T., Woo, G., Liu, J., Liu, X., Liu, C., Savarese, S., Xiong, C., 和 Sahoo, D. Gift-Eval：通用时间序列预测模型评估基准。*arXiv* 预印本 *arXiv:2410.10393*，2024年。

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S., Arango, S. P., Kapoor, S., Zschiegner, J., Robinson, D. M., Wang, H., Mahoney, M., Torkkola, K., Wilson, A., Bohlke- Schneider, M., and Wang, Y. B. Chronos：学习时间序列语言。机器学习研究汇刊，2024年。

Bishop, C. M. 模式识别与机器学习。Springer，2006年。

B¨uy¨ukaky¨uz, K. Olora：大语言模型的正交低秩适应。*arXiv*预印本 *arXiv:2406.01775*，2024年。

Chronopoulou, A.、Peters, M. E.、Fraser, A. 与 Dodge, J. Adaptersoup：通过权重平均改进预训练语言模型的泛化。*arXiv* 预印本 *arXiv:2302.07027*，2023年。

Darlow, L.、Deng, Q.、Hassan, A.、Asenov, M.、Singh, R.、Joosen, A.、Barker, A. 与 Storkey, A. Dam：迈向时间序列预测的基础模型。收录于第十二届国际学习表征会议论文集，2024年。

Darlow, L. N.、Joosen, A.、Asenov, M.、Deng, Q.、Wang, J. 与 Barker, A. Foldformer：用于细粒度长期 Faas 预测的序列折叠与季节性注意力。收录于 第三届机器学习与系统研讨会论文集，第 71–77 页，2023年。

Das, A.、Faw, M.、Sen, R. 和 Zhou, Y. 面向时间序列基础模型的上下文内微调。*arXiv* 预印本 *arXiv:2410.24087*，2024a。

Das, A.、Kong, W.、Sen, R. 和 Zhou, Y. 面向时间序列预测的仅解码器基础模型。载于第四十一届国际机器学习会议论文集，2024b。

De Lange, M.、Aljundi, R.、Masana, M.、Parisot, S.、Jia, X.、Leonardis, A.、Slabaugh, G. 和 Tuytelaars, T. 持续学习综述：应对分类任务中的遗忘。*IEEE* 模式分析与机器智能汇刊，44(7)：3366–3385，2021年。

Diao, Y., Horn, D., Kipf, A., Shchur, O., Benito, I., Dong, W., Pagano, D., Pfeil, P., Nathan, V., Narayanaswamy, B., 等。《智能资源扩展预测算法：一项实验分析》。载于会议论文集

*2024*年 *ACM*云计算研讨会，第 126–143 页，2024年。Dieng, A. B., Ruiz, F. J., 和 Blei, D. M. 嵌入空间中的主题建模。《style id='3'>计算语言学协会汇刊，8:439–453，2020年。Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Liu, T., 等。上下文学习综述。*arXiv* 预印本 *arXiv:2301.00234*，2022年。

Ekambaram, V.、Jati, A.、Dayama, P.、Mukherjee, S.、Nguyen, N. H.、Gifford, W. M.、Reddy, C. 和 Kalagnanam, J. 微型时间混合器（TTMs）：用于增强多变量时间序列零样本/少样本预测的快速预训练模型。*CoRR*，2024年。

Gadre, S. Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., 等。Datacomp：寻找新一代多模态数据集。神经信息处理系统进展，36：27092–27112，2023年。

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., 和 Wang, H. 面向大语言模型的检索增强生成：综述。*arXiv* 预印本 *arXiv:2312.10997*，2(1)，2023年。

Godahewa, R. W., Bergmeir, C., Webb, G. I., Hyndman, R., 和 Montero-Manso, P. 莫纳什时间序列预测档案库。载于第三十五届神经信息处理系统进展会议论文集，数据集与基准赛道，2021年。

Gouk, H., Bohdal, O., Li, D., 和 Hospedales, T. 论通用领域泛化方法的局限性。*arXiv* 预印本 *arXiv:2202.00563*，2022年。

Gupta, D., Bhatti, A., 和 Parmar, S. 超越 LoRA：探索适用于时间序列基础模型的高效微调技术。*arXiv* 预印本 *arXiv:2409.11302*，2024a。

Gupta, D., Bhatti, A., Parmar, S., Dan, C., Liu, Y., Shen, B., 和 Lee, S. 面向域外模态预测的时间序列基础模型低秩适配。载于第*26*届国际多模态交互会议论文集，第382–386页，2024b。

Han, Z., Gao, C., Liu, J., Zhang, J., 和 Zhang, S. Q. 大型模型的参数高效微调：全面综述。*arXiv* 预印本 *arXiv:2403.14608*，2024年。

He, B., Ding, N., Qian, C., Deng, J., Cui, G., Yuan, L., Hong, H., Gao, H.-a., Huang, L., Xue, H. 等。时机至关重要：数据排列影响指令微调中的零样本泛化。*arXiv*预印本 *arXiv:2406.11721*，2024年。

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. 和 Chen, W. LoRA：大语言模型的低秩适配。*arXiv*预印本 *arXiv:2106.09685*，10，2021年。

Hyndman, R. J. 和 Koehler, A. B. 再探预测准确度的衡量方法。《国际预测学杂志》，22(4):679–688，2006年。

Kim, H.、Kim, S.、Mok, J. 和 Yoon, S. 通过测试时适配应对时间序列预测中的非平稳性。载于第*39*届*AAAI*人工智能会议论文集，2025年。

Kim, T.、Kim, J.、Tae, Y.、Park, C.、Choi, J.-H. 和 Choo, J. 用于应对分布偏移、实现准确时间序列预测的可逆实例归一化。载于第九届国际学习表征会议论文集，2021年。

Lee, T. L.、Toner, W.、Singh, R.、Joosen, A. 和 Asenov, M. 用于时间序列基础模型预测结果的轻量级在线适配。载于第四十二届国际机器学习会议，2025年。

Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., Pan, S. 和 Wen, Q.《时间序列分析基础模型：教程与综述》。载于第*30*届*ACM SIGKDD*知识发现与数据挖掘会议论文集，第6555–6565页，2024年。

Lim, B. 和 Zohren, S.《深度学习时间序列预测：综述》。《皇家学会哲学汇刊*A*辑》，379(2194):20200209，2021年。

Liu, B., Liu, X., Jin, X., Stone, P. 和 Liu, Q.《用于多任务学习的冲突规避梯度下降》。神经信息处理系统进展，34:18878–18890，2021年。

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. 和 Neubig, G.《预训练、提示与预测：自然语言处理提示方法的系统综述》。《*ACM*计算调查》，55(9):1–35，2023年。

Loshchilov, I. 和 Hutter, F. 解耦权重衰减正则化。*arXiv* 预印本 *arXiv:1711.05101*，2017年。

Maharana, A.、Yoon, J.、Chen, T. 和 Bansal, M. Adapt-inf：通过动态数据选择实现可扩展的持续多模态指令微调。*arXiv* 预印本 *arXiv:2410.10636*，2024年。

Mehta, R. 与 Harchaoui, Z. 零样本预测泛化理论。*arXiv* 预印本 *arXiv:2507.09128*，2025年。

Mercier, D., Lucieri, A., Munir, M., Dengel, A. 与 Ahmed, S. 关键基础设施中隐私保护机器学习的评估：时间序列分类案例研究。《*IEEE*工业信息学汇刊》，18(11):7834–7842，2021年。Miller, J. A., Aldosari, M., Saeed, F., Barna, N. H., Rana, S., Arpinar, I. B. 与 Liu, N. 面向时间序列预测的深度学习与基础模型综述。*arXiv* 预印本 *arXiv:2401.13912*，2024年。Murphy, K. P. 概率机器学习：导论。麻省理工学院出版社，2022年。

Nie, Y.、Nguyen, N. H.、Sinthong, P. 和 Kalagnanam, J. 一个时间序列相当于64个词：使用Transformer进行长期预测。发表于第十一届国际学习表征会议，2022年。

Ning, K.、Pan, Z.、Liu, Y.、Jiang, Y.、Zhang, J. Y.、Rasul, K.、Schneider, A.、Ma, L.、Nevmyvaka, Y. 和 Song, D. Ts-rag：基于检索增强生成的时间序列基础模型是更强大的零样本预测器。*arXiv* 预印本 *arXiv:2503.07649*，2025年。

Ostapenko, O., Lesort, T., Rodriguez, P., Arefin, M. R., Douillard, A., Rish, I., and Charlin, L. 基于基础模型的持续学习：对潜在回放的实证研究。载于第一届终身学习智能体会议论文集，第 60–91 页。PMLR，2022年。

Ostapenko, O., Su, Z., Ponti, E. M., Charlin, L., Roux, N. L., Pereira, M., Caccia, L., and Sordoni, A. 通过构建和复用LoRA库，迈向模块化大语言模型。*arXiv* 预印本 *arXiv:2405.11157*，2024年。

Page-Caccia, L., Ponti, E. M., Su, Z., Pereira, M., Le Roux, N., and Sordoni, A. 用于跨任务泛化的多头适配器路由。神经信息处理系统进展，36:56916–56931，2023年。

Palaskar, S., Ekambaram, V., Jati, A., Gantayat, N., Saha, A., Nagar, S., Nguyen, N. H., Dayama, P., Sindhgatta, R., and Mohapatra, P. Automixer：用于改进业务与IT可观测性数据多变量时间序列预测的工具。载于*AAAI*人工智能会议论文集，第 38 卷，第 22962–22968 页，2024年。

Ponti, E. M., Sordoni, A., Bengio, Y., and Reddy, S. 结合参数高效模块以实现任务级泛化。载于 第 *17* 届欧洲计算语言学协会分会会议论文集，第 687–702 页，2023年。

Prabhu, A., Al KaderHammoud, H. A., Dokania, P. K., Torr, P. H., Lim, S.-N., Ghanem, B. 和 Bibi, A. 计算预算受限的持续学习：什么才重要?

发表于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 3698–3707 页，2023年。

Rasul, K., Ashok, A., Williams, A. R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Bilo˘s, M., Ghonia, H., Hassen, N. 和 Schneider, A. Lag-llama：迈向时间序列预测基础模型。载于 *NeurIPS 2023* 大型基础模型中少样本与零样本学习鲁棒性研讨会论文集 *R0-fomo*，2023年。

Shen, S., Van Beek, V. 和 Iosup, A. 云数据中心托管的业务关键型工作负载的统计特征。载于 第*15*届 *IEEE/ACM* 集群、云和网格计算国际研讨会论文集，2024年。

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A. 和 Choi, Y. 数据集制图：利用训练动态映射和诊断数据集。*arXiv* 预印本 *arXiv:2009.10795*，2020年。

Toner, W. 和 Darlow, L. N. 线性时间序列预测模型分析。载于第四十一届国际机器学习会议论文集，2024年。

Toner, W.、Lee, T. L.、Joosen, A.、Singh, R. 和 Asenov, M. 零样本时间序列基础模型在云数据上的表现。载于*NeurIPS*，“简直难以置信，竟然没有更好：应用深度学习的挑战”研讨会，2025年。

Vayansky, I. 和 Kumar, S. A. 主题建模方法综述。信息系统，94:101582，2020。

Woo, G.、Liu, C.、Kumar, A.、Xiong, C.、Savarese, S. 和 Sahoo, D. 通用时间序列预测Transformer的统一训练。载于第四十一届国际机器学习会议论文集，2024年。

Yang, S.、Wang, D.、Zheng, H. 和 Jin, R. TimeRAG：通过检索增强生成提升大语言模型的时间序列预测。载于*IEEE*声学、语音与信号处理国际会议（*ICASSP*）论文集，第1–5页，2025年。

Ye, J., Zhang, W., Yi, K., Yu, Y., Li, Z., Li, J.和Tsung, F. 时间序列基础模型综述：利用大语言模型泛化时间序列表示。*arXiv*预印本 *arXiv:2405.02358*，2024年。

Zhang, H., Cisse, M., Dauphin, Y. N. 和 Lopez-Paz, D. MixUp：超越经验风险最小化。*arXiv* 预印本 *arXiv:1710.09412*，2017年。

Zhang, Y., Chen, W., Zhu, Z., Qin, D., Sun, L., Wang, X., Wen, Q., Zhang, Z., Wang, L. 和 Jin, R. 解决在线时间序列预测中的概念漂移：先检测，再适应。*arXiv* 预印本 *arXiv:2403.14949*，2024年。

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H. 和 Zhang, W. Informer：超越高效 Transformer，实现长序列时间序列预测。载于第三十五届*AAAI*人工智能大会论文集，第 11106–11115 页，2021年。

A. 局限性

在正文中，我们旨在突出说明 MixFT 的设计决策与局限性。此外，我们还想在此阐述 MixFT 的主要局限性。首先，我们假设微调数据与评估数据共享相同的子领域。因此，如果情况并非如此，我们不认为 MixFT 会有良好表现。不过，我们注意到，在实验中，子领域似乎确实是共享的，这一点从 MixFT 的良好表现也可看出。此外，有必要假设微调数据与评估数据之间存在某种关联，才能对零样本泛化作出任何论断（Gouk 等人，2022 年）。因此，尽管我们明确说明了如何关联微调数据与评估数据的假设，任何类似的方法也都必须作出某种假设，无论是明确还是隐含的。最后，我们面向的现实应用场景是：由从业者选择并使用微调数据，以提升特定类型时间序列的零样本预测能力。因此，从业者应当清楚自己为何认为所选微调数据与其想要评估的时间序列相关，从而判断 MixFT 的假设是否成立。

我们想提及的第二个局限性是使用 MixFT 带来的内存/参数开销和计算开销。就基础模型之外所需增加的内存或参数而言，MixFT 需要为每个子域存储一个 LoRA 模块，而贝叶斯高斯混合模型所需的内存则可以忽略不计。与我们实验中作为对比的先进方法 Poly 和 MBC 相比，这一开销大致相同。此外，它低于 Arrow-Dataset，后者为每个数据集使用一个 LoRA 模块。另一方面，在进行零样本预测时，Shared 和 μ-Datasets 方法所需的参数/内存不会超过基础模型。这是因为，这些方法使用的单个 LoRA 模型可以合并到基础模型中。就额外的计算成本而言，微调时，MixFT 必须拟合贝叶斯高斯混合模型，这可能需要一些时间，因此其训练速度比共享微调慢。不过，我们发现，MixFT 的训练时间与 MBC 等其他方法大致相同。进行预测时，MixFT 必须对上下文所属的子域进行分类，这会带来计算开销，因为它需要基础模型提取上下文的特征表示。因此，与使用共享微调相比，MixFT 生成预测结果所需的时间更长。不过，MixFT 生成预测结果的速度低于我们实验所用数据集的采样频率，因此其计算开销的影响较小。

B. 附加方法论细节

B.1. MixFT 微调与零样本预测算法

算法 1：MixFT 零样本预测

输入： 时间序列上下文 $\boldsymbol{x}$；每个子域 LoRA 模块 $[\text{LoRA}_k]_{k=1}^K$；贝叶斯 GMM $[\boldsymbol{m}_k^{\text{VAR}}, \kappa_k^{\text{VAR}}, \nu_k^{\text{VAR}}, \boldsymbol{W}_k^{\text{VAR}}]_{k=1}^K$，$\boldsymbol{\alpha}^{\text{VAR}}$；以及时间序列基础模型

识别 $\boldsymbol{x}$ 的子域：

$\hat{c} = \arg\max_{c\in[1,\dots,K]} [p(c|\boldsymbol{z} = \text{时间序列基础模型}_{embed}(\boldsymbol{x}),\ \boldsymbol{m}_k^{\text{VAR}},\ \kappa_k^{\text{VAR}},\ \nu_k^{\text{VAR}},\ \boldsymbol{W}_k^{\text{VAR}},\ \boldsymbol{\alpha}^{\text{VAR}})]$

使用第 $\hat{c}$ 个子域的 LoRA 模块计算并返回时间序列基础模型的预测值：

$\hat{\boldsymbol{y}} = \text{预测值}(\text{时间序列基础模型},\ \text{LoRA}_c,\ \boldsymbol{x})$

返回 $\hat{\boldsymbol{y}}$

算法 2：MixFT 微调

输入： 微调数据集 $D_i = [\boldsymbol{x}_{i,j}]_{j=1}^{N_i}$ for $i \in [1, \ldots, M]$; 时间序列基础模型（TSFM$_{embed}$）；以及混合成分的数量 K

嵌入数据集：
for $i \in [1, \ldots, M]$ **do**
$D_i^Z = [\boldsymbol{z}_{i,j} =$时间序列基础模型$_{embed}(\boldsymbol{x}_{i,j})]_{j=1}^{N_i}$
结束循环

拟合贝叶斯高斯混合模型：
$[\boldsymbol{m}_k^{\text{VAR}}, \kappa_k^{\text{VAR}}, \nu_k^{\text{VAR}}, \boldsymbol{W}_k^{\text{VAR}}]_{k=1}^K,\ \boldsymbol{\alpha}^{\text{VAR}} =$贝叶斯高斯混合模型$(\bigcup_{i=1}^M D_i^Z)$

为每个已学习的子域构造伪数据集：
对于 $i \in [1, \ldots, M]$ **执行**
对于 $j \in [1, \ldots, N_i]$ **执行**
$\hat{c}_{i,j} = \arg\max_{c \in [1,\ldots,K]} [p(c|\boldsymbol{z}_{i,j}, \boldsymbol{m}_k^{\text{VAR}}, \kappa_k^{\text{VAR}}, \nu_k^{\text{VAR}}, \boldsymbol{W}_k^{\text{VAR}}, \boldsymbol{\alpha}^{\text{VAR}})]$
结束 for
结束 for

对于 $k \in [1, \ldots, K]$ **执行**
$\mathcal{S}_k = \{\boldsymbol{x}_{i,j} | \hat{c}_{i,\ j} = k\}$
结束 for

训练各子域LoRA模块：
对于 $k \in [1, \ldots, K]$ **执行**
LoRA$_k$ = TrainLoRA($\mathcal{S}_k$,时间序列基础模型)
结束循环

返回各子域LoRA模块和贝叶斯高斯混合模型：
返回 $[\text{LoRA}_k]_{k=1}^K, [\boldsymbol{m}_k^{\text{VAR}}, \kappa_k^{\text{VAR}}, \nu_k^{\text{VAR}}, \boldsymbol{W}_k^{\text{VAR}}]_{k=1}^K,\ \boldsymbol{\alpha}^{\text{VAR}}$

C. 其他实验细节

虽然我们已在正文中介绍了最重要的实验细节，但还有一些细节值得说明。首先，在我们的实验中，MixFT 的贝叶斯高斯混合模型超参数设定如下：

$$\boldsymbol{m} = \text{Mean}(\bigcup_{i=1}^{M} D_i^Z) \tag{12}$$

$$\kappa = 1 \tag{13}$$

$$\nu = \text{Dim}(Z) \tag{14}$$

$$\boldsymbol{W} = \text{Diag}(\text{Diag}(\text{Covariance}(\bigcup_{i=1}^{M} D_i^Z))) \tag{15}$$

$$\boldsymbol{\alpha} = \frac{1}{K}\mathbf{1}, \tag{16}$$

其中，Dim(Z) 表示时间序列基础模型嵌入空间的维度，Diag(Diag(.)) 接收一个矩阵，并返回仅保留对角元素、将非对角元素设为零的矩阵。选择这些超参数是因为它们是以往工作中常用的默认值。其次，我们尽量将其他微调方法的所有超参数保留为默认设置。第三，为确保在不同方法间以一致的方式调整 LoRA，我们对所有微调方法采用相同的 LoRA 超参数。我们在 ETTh1 和 ETTm1 数据集上手动调优了所用的 LoRA 超参数，效果良好；这些参数的设置参考了（Gupta et al., 2024b）。具体而言，我们使用的 LoRA 超参数为 $r = 2$、$\alpha = 16$、dropout $= 0.1$；此外，我们还采用 OLoRA（B¨uy¨ukaky¨uz，2024年）来提升低秩训练性能。为调优每个 LoRA 模块，我们使用 AdamW（Loshchilov & Hutter, 2017），学习率为 $lr = 0.00005$，批量大小为 256。第四，为了将时间序列基础模型提供的词元转换为贝叶斯高斯混合模型所用的表示，我们构造一个由每个词元的均值组成的向量。最后，鉴于我们关注的是零样本预测，因此不会像一些以往工作（Nie et al., 2022）那样使用训练划分对数据集进行缩放。相反，我们依赖时间序列基础模型内部的缩放机制。对于 Chronos Bolt 和 Moirai-1.1-R，该机制均为可逆实例归一化（RevIN）（Kim et al., 2021）。

在正文中，我们提到，为了改进时间序列上的 LoRA 微调，我们采用了 **a)** MixUp（Zhang 等人，2017年）和**b)** 重放预训练数据。此处我们将更详细地介绍这两种方法的用法。对于 MixUp，我们采用标准的动态变体，按照 Zhang 等人（2017年）所述对每个批次进行混合。不过，为确保不同尺度的时间序列不会使混合结果产生偏差，我们使用 RevIN 对数据进行缩放，做法与 Ansari 等人（2024年）相同。对于预训练数据的重放，我们针对每个微调数据训练批次，从预训练数据中采样与该批次大小相同数量的数据点。随后将这些重放数据追加到训练批次中。我们在使用 MixUp 之前追加重放数据，因此预训练数据也会与微调数据混合。对于所考察的两个时间序列基础模型，我们都使用 Chronos Bolt 的预训练数据，因为我们无法完全确定 Moirai-1.1-R 使用了哪些数据进行预训练。

D. 补充实验结果

D.1. 混合组件数量的选择

表3. **MixFT使用不同数量组件时的验证性能** K。 该表列出了MixFT使用不同数量组件时的平均排名。平均值是对微调数据集的验证集取平均得到的。表中显示， $K = 2$ 的平均排名最低，因此应当选择该数量。

组件数量	验证集平均排名（↓）
$K = 2$	**2.17**
$K = 3$	3.17
$K = 4$	2.83
$K = 5$	3.33
$K = 10$	3.50

要选择MixFT使用的混合成分数量，我们可以采用微调数据上的验证性能，这也是标准做法（Ostapenko 等人，2024年）。具体而言，我们使用平均排名作为性能指标，并将每个微调数据集最后 10% 的数据用作验证数据。验证性能结果见表3。表中显示，使用两个组件（$K = 2$）可获得最佳验证性能。这一结果证明了我们在主要实验中采用 $K = 2$ 的合理性。此外，将MixFT使用不同数量组件时的验证性能与其在相应评估数据集上的性能进行对比也很有意义，结果见表6。我们发现，验证性能与评估性能总体一致，其中 $K = 2$ 在两种评估中都取得了最佳表现。这表明我们选择混合成分数量的方法能在评估数据集上取得良好表现。不过，正如预期的那样，验证性能无法完全预测不同组件数量在评估中的相对表现。例如， $K = 3$ 和 $K = 4$在验证性能和评估性能上的排序并不相同。

D.2. 零样本混合选择消融实验

表4. **使用 Chronos Bolt 进行零样本预测时，采用不同组件选择方法的 MixFT 版本的平均绝对缩放误差**。该表展示了带有 μ 和 Arrow 路由的 MixFT，这些方法被正文中所比较的其他一些微调方法所使用。该表还展示了两种利用 MixFT 的贝叶斯高斯混合模型给出的概率来进行组件选择的方法，即软方法和 Ensemble-MixFT。在表中，我们分别对最佳和次佳结果进行加粗和下划线标注。我们发现，除 CloudD1 外，MixFT 在所有时间序列上均取得最佳或次佳表现，并获得最低的平均排名。这表明 MixFT 选择用于零样本预测的 LoRA 的方式表现良好。

	采用不同组件选择方法的 MixFT				
数据集	μ MixFT	Arrow MixFT	Soft MixFT	集成 MixFT	MixFT（我们的方法）
CloudD1	$\mathbf{1.411}_{\pm 0.003}$	$1.594_{\pm 0.055}$	$1.697_{\pm 0.009}$	$1.480_{\pm 0.003}$	$1.485_{\pm 0.008}$
CloudD2	$1.425_{\pm 0.001}$	$1.384_{\pm 0.006}$	$1.401_{\pm 0.010}$	$1.351_{\pm 0.001}$	$\mathbf{1.343}_{\pm 0.004}$
BizITObs-L2C	$3.087_{\pm 0.002}$	$3.163_{\pm 0.013}$	$3.237_{\pm 0.012}$	$3.088_{\pm 0.003}$	$\mathbf{3.083}_{\pm 0.006}$
BizITObs-App	$1.907_{\pm 0.016}$	$1.277_{\pm 0.038}$	$1.168_{\pm 0.014}$	$1.015_{\pm 0.002}$	$\mathbf{0.989}_{\pm 0.024}$
US-Births	$0.973_{\pm 0.001}$	$0.977_{\pm 0.002}$	$0.968_{\pm 0.001}$	$0.946_{\pm 0.001}$	$\mathbf{0.942}_{\pm 0.002}$
M4-Daily	$\mathbf{7.172}_{\pm 0.001}$	$7.450_{\pm 0.024}$	$7.535_{\pm 0.032}$	$7.238_{\pm 0.005}$	$7.238_{\pm 0.005}$
M4-Monthly	$7.639_{\pm 0.007}$	$7.780_{\pm 0.008}$	$7.911_{\pm 0.052}$	$7.690_{\pm 0.043}$	$\mathbf{7.638}_{\pm 0.014}$
M4-Quarterly	$\mathbf{8.100}_{\pm 0.008}$	$8.344_{\pm 0.017}$	$8.423_{\pm 0.020}$	$8.140_{\pm 0.024}$	$8.135_{\pm 0.010}$
ETTh2	$\mathbf{1.381}_{\pm 0.001}$	$1.411_{\pm 0.002}$	$1.413_{\pm 0.001}$	$1.389_{\pm 0.001}$	$1.387_{\pm 0.001}$
ETTm2	$\mathbf{0.723}_{\pm 0.001}$	$0.749_{\pm 0.001}$	$0.747_{\pm 0.001}$	$0.730_{\pm 0.001}$	$0.730_{\pm 0.001}$
平均排名 (↓)	2.3	4.1	4.4	2.4	**1.6**

为分析使用 MixFT 的零样本混合选择方法的收益，我们将其与其他选择方法进行比较。首先，我们将其与本研究中所考察的其他微调方法的混合选择方法进行比较，即 μ 和 Arrow 路由。由此，我们得到替代的 MixFT 版本 μ-MixFT 和 Arrow-MixFT。其次，我们将 MixFT 的硬 选择机制与利用贝叶斯高斯混合模型给出的概率（而非对其取 argmax）的软选择机制进行比较。Soft-MixFT 是指对各子域 LoRA模块取加权平均。我们使用给定上下文属于某一子域的概率作为该子域 LoRA 的权重。Ensemble-MixFT 则不是对 LoRA模块取平均，而是对将每个子域 LoRA模块与时间序列基础模型结合所得到的预测结果取平均。同样，所用权重是给定上下文属于各子域的概率。

该消融实验的结果如表 4所示，其中表明 MixFT 的选择机制表现最佳。结果显示，Arrow 和软选择机制表现最差，平均排名相近。而 μ 和集成 MixFT 的表现更好。有趣的是，μ 路由比 Arrow 路由更简单，但在 MixFT 上表现优于后者；而在逐数据集方法上，表现则不如后者。这表明 Arrow 路由似乎非常适合逐数据集方法，但在 MixFT 上效果不佳。结果还显示，MixFT 的硬选择机制表现最佳，因为它的平均排名最低，并且在除 CloudD1 之外的所有数据集上都表现最好或并列第二。这证明了采用 MixFT 的零样本选择机制是合理的。此外，结果表明，由于软方法的表现不及 MixFT 的硬选择机制，将概率转换为 LoRA 模块加权可能并非易事。我们认为这是一个值得未来工作探索的方向。

D.3. 混合模型消融实验

表 5. **使用 Chronos Bolt 时，采用不同概率模型的 MixFT 版本的平均绝对缩放误差。** MixFT 使用贝叶斯高斯混合模型来识别和学习子领域。表中展示了使用多种不同概率模型来识别和学习子领域时的性能。我们分别将最佳和次佳结果加粗并加下划线。结果表明，在除 M4-Daily 之外的所有数据集上，使用贝叶斯高斯混合模型的表现均为最佳或次佳，且平均排名最低。这说明它是 MixFT 概率模型的一个不错选择。

	采用不同概率模型的 MixFT		
数据集	K均值	主题模型	贝叶斯高斯混合模型（我们的方法）
CloudD1	$\underline{1.516_{\pm 0.005}}$	$1.532_{\pm 0.004}$	$\mathbf{1.485_{\pm 0.008}}$
CloudD2	$1.363_{\pm 0.003}$	$\mathbf{1.328_{\pm 0.009}}$	$\underline{1.343_{\pm 0.004}}$
BizITObs-L2C	$\underline{3.096_{\pm 0.005}}$	$3.100_{\pm 0.002}$	$\mathbf{3.083_{\pm 0.006}}$
BizITObs-App	$1.048_{\pm 0.006}$	$\underline{1.030_{\pm 0.041}}$	$\mathbf{0.989_{\pm 0.024}}$
US-Births	$0.967_{\pm 0.005}$	$\underline{0.960_{\pm 0.009}}$	$\mathbf{0.942_{\pm 0.002}}$
M4-Daily	$\mathbf{7.230_{\pm 0.004}}$	$\underline{7.237_{\pm 0.008}}$	$7.238_{\pm 0.005}$
M4-Monthly	$7.737_{\pm 0.006}$	$\mathbf{7.552_{\pm 0.002}}$	$\underline{7.638_{\pm 0.014}}$
M4-Quarterly	$\mathbf{8.131_{\pm 0.003}}$	$8.238_{\pm 0.014}$	$\underline{8.135_{\pm 0.010}}$
ETTh2	$\underline{1.390_{\pm 0.001}}$	$\mathbf{1.387_{\pm 0.001}}$	$\mathbf{1.387_{\pm 0.001}}$
ETTm2	$\underline{0.730_{\pm 0.001}}$	$\mathbf{0.728_{\pm 0.001}}$	$\underline{0.730_{\pm 0.001}}$
平均排名（↓）	2.3	<u>1.9</u>	**1.6**

为了探索使用贝叶斯高斯混合模型对各子领域进行建模，相较于使用其他概率模型的优势，我们进行了消融实验。我们比较了 MixFT 使用贝叶斯高斯混合模型的方案与 MixFT 的其他版本：一种使用通过 K均值拟合的高斯混合模型，另一种使用主题模型（Vayansky& Kumar，2020）。我们采用的具体主题模型是在（Dieng 等人，2020）提出的模型基础上调整而来。我们做出的改动包括：**a**）不再学习嵌入，而是使用时间序列基础模型的嵌入；以及 **b**）鉴于我们的“词”（即上下文）是连续的，因此给定文档和主题分配后，我们使用高斯分布作为其分布。

该消融实验的结果见表 5，结果表明贝叶斯高斯混合模型表现最佳。具体而言，从平均排名来看，我们发现使用 K均值的表现较差。这可能是因为它容易像已知的那样过拟合于某些浅层局部极小值（Bishop，2006，第 9 章）。虽然主题模型的表现排名第二，与贝叶斯高斯混合模型

大致相当，但这并不令人意外，因为主题模型与贝叶斯高斯混合模型非常相似，只是额外依赖于数据集。不过有趣的是，这种额外依赖（用于调节每个数据集中会出现哪些子领域）并没有带来帮助。我们认为，未来工作可以探索如何改进这一点，在一定程度上利用数据集标签，同时避免发展成完整的逐数据集方法。最后，我们发现贝叶斯高斯混合模型表现最佳。这验证了我们将其用作 MixFT 子领域概率模型的做法。

D.4. 混合成分数量的消融实验

表 *6.* **使用 Chronos Bolt 时，采用不同数量混合成分的 MixFT 的平均绝对缩放误差。** 表中展示了使用一个到十个混合成分时 MixFT 的表现。使用一个混合成分等同于共享，而使用两个混合成分则是我们其余实验中采用的设置。表中分别以粗体和下划线标出最佳和次佳结果。我们发现，不同数量的混合成分通常表现相近，而从平均排名来看，使用两个混合成分的效果最佳。

	使用不同数量组件的 MixFT					
数据集	$K=1$(共享)	$K=2$(我们的方法)	$K=3$	$K=4$	$K=5$	$K=10$
CloudD1	$1.553_{\pm 0.024}$	$\underline{1.485}_{\pm 0.008}$	$1.500_{\pm 0.010}$	$1.489_{\pm 0.002}$	$1.518_{\pm 0.017}$	$\mathbf{1.448}_{\pm \mathbf{0.004}}$
CloudD2	$\mathbf{1.342}_{\pm \mathbf{0.004}}$	$\underline{1.343}_{\pm 0.004}$	$1.375_{\pm 0.010}$	$1.365_{\pm 0.003}$	$1.384_{\pm 0.007}$	$1.423_{\pm 0.003}$
BizITObs-L2C	$3.101_{\pm 0.011}$	$3.083_{\pm 0.006}$	$\underline{3.061}_{\pm 0.009}$	$\mathbf{3.058}_{\pm \mathbf{0.009}}$	$3.068_{\pm 0.007}$	$3.065_{\pm 0.010}$
BizITObs-App	$1.060_{\pm 0.020}$	$0.989_{\pm 0.024}$	$1.001_{\pm 0.018}$	$\mathbf{0.970}_{\pm \mathbf{0.013}}$	$\underline{0.985}_{\pm 0.019}$	$1.127_{\pm 0.062}$
US-Births	$\underline{0.967}_{\pm 0.007}$	$\mathbf{0.942}_{\pm \mathbf{0.002}}$	$1.007_{\pm 0.005}$	$1.034_{\pm 0.006}$	$1.025_{\pm 0.001}$	$1.012_{\pm 0.002}$
M4-Daily	$7.254_{\pm 0.016}$	$7.238_{\pm 0.005}$	$\mathbf{7.233}_{\pm \mathbf{0.004}}$	$\underline{7.236}_{\pm 0.003}$	$7.244_{\pm 0.002}$	$7.245_{\pm 0.007}$
M4-Monthly	$\mathbf{7.580}_{\pm \mathbf{0.042}}$	$\underline{7.638}_{\pm 0.014}$	$7.707_{\pm 0.041}$	$7.739_{\pm 0.008}$	$7.760_{\pm 0.012}$	$7.708_{\pm 0.017}$
M4-Quarterly	$8.269_{\pm 0.037}$	$\mathbf{8.135}_{\pm \mathbf{0.010}}$	$\underline{8.154}_{\pm 0.015}$	$8.188_{\pm 0.006}$	$8.183_{\pm 0.024}$	$8.169_{\pm 0.021}$
ETTh2	$1.388_{\pm 0.001}$	$1.387_{\pm 0.001}$	$1.391_{\pm 0.001}$	$1.389_{\pm 0.001}$	$\underline{1.385}_{\pm 0.001}$	$\mathbf{1.383}_{\pm \mathbf{0.001}}$
ETTm2	$0.731_{\pm 0.003}$	$0.730_{\pm 0.001}$	$0.730_{\pm 0.001}$	$0.729_{\pm 0.001}$	$\underline{0.728}_{\pm 0.001}$	$\mathbf{0.727}_{\pm \mathbf{0.001}}$
平均排名（↓）	4.3	**2.6**	<u>3.3</u>	3.4	3.9	3.4

D.5. 已识别的子领域分析

D.5.1. MIXFT 的子领域分类有多可靠

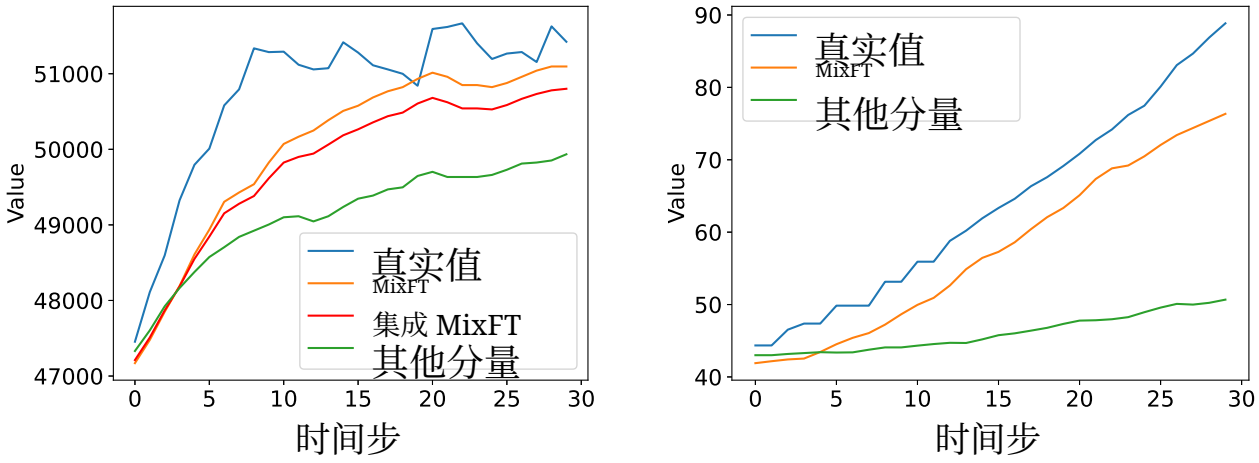
表 7. **MixFT 对上下文子域分类的置信度，以平均分类熵衡量**。 数据集 D 的平均分类熵为 $\mathbb{E}_{\boldsymbol{x}\sim D}[H(c|\boldsymbol{z} = \text{时间序列基础模型}_{embed}(\boldsymbol{x}))]$，用于衡量对 $\boldsymbol{x} \in D$ 中 c 进行分类时的平均不确定性。我们发现，所有微调和评估数据集的平均熵都很小；熵为 0.01 比特对应于上下文属于某一特定子域的估计概率为 0.999。这表明，MixFT 平均而言非常确信哪些上下文属于哪些子领域。

	数据集	平均分类熵 $\mathbb{E}_{\boldsymbol{x}\sim D}[H(c\|\boldsymbol{z} = \text{时间序列基础模型}_{embed}(\boldsymbol{x}))]$
Fine-tuning	CloudD3	0.0095
	CloudD4	0.0075
	BizITObs-Service	0.0053
	BitBrains Fast Storage	0.0101
	M4-Hourly	0.0093
	M4-Weekly	0.0050
	平均值	**0.0078**
Evaluation	CloudD1	0.0124
	CloudD2	0.0110
	BizITObs-L2C	0.0054
	BizITObs-App	0.0037
	US-Births	0.0131
	M4-Daily	0.0062
	M4-Monthly	0.0054
	M4-Quarterly	0.0068
	ETTh2	0.0168
	ETTm2	0.0145
	平均值	**0.0095**

为了了解 MixFT 对上下文子域分类的置信度，表 7 展示了我们实验中所用每个数据集的平均分类熵$\mathbb{E}_{\boldsymbol{x}\sim D}[H(c|\boldsymbol{z} = \text{时间序列基础模型}_{embed}(\boldsymbol{x}))]$。鉴于 MixFT 使用两个组件，对于某个 $\boldsymbol{x}$， $H(c|\boldsymbol{z} = \text{时间序列基础模型}_{embed}(\boldsymbol{x}))$的最大值为 1 比特，而低于 0.01 则表示$p(c|\boldsymbol{z}, \phi_c^{\text{VAR}}, \boldsymbol{\alpha}^{\text{VAR}}) \geq 0.999$最可能的组件 c。因此，表 7 中各数据集的平均熵均接近或低于 0.01，这表明 MixFT 平均而言对上下文子域的分类非常有把握。对于用于拟合贝叶斯高斯混合模型的微调数据集和评估数据集，这一发现都成立；总体而言，后者数据集的平均熵略高。

考察 MixFT 对上下文子域分类置信度的一个原因，是探究预测时是否只使用最可能子域的 LoRA 模块是否合理。这是因为，如果某个上下文的子域有多个可能的候选项，那么只使用最可能子域的 LoRA 模块，就不如使用多个 LoRA 的混合合理。然而，鉴于实际情况并非如此，因为通常给定上下文只可能属于一个子域（见表 7），预测时只使用最可能子域的 LoRA 模块是合理的。此外，下一节将展示，这也使 MixFT 能够在使用不太可能的子域的 LoRA 模块而产生不准确预测值时保持稳健。

D.5.2. 已识别的子领域预测的特征



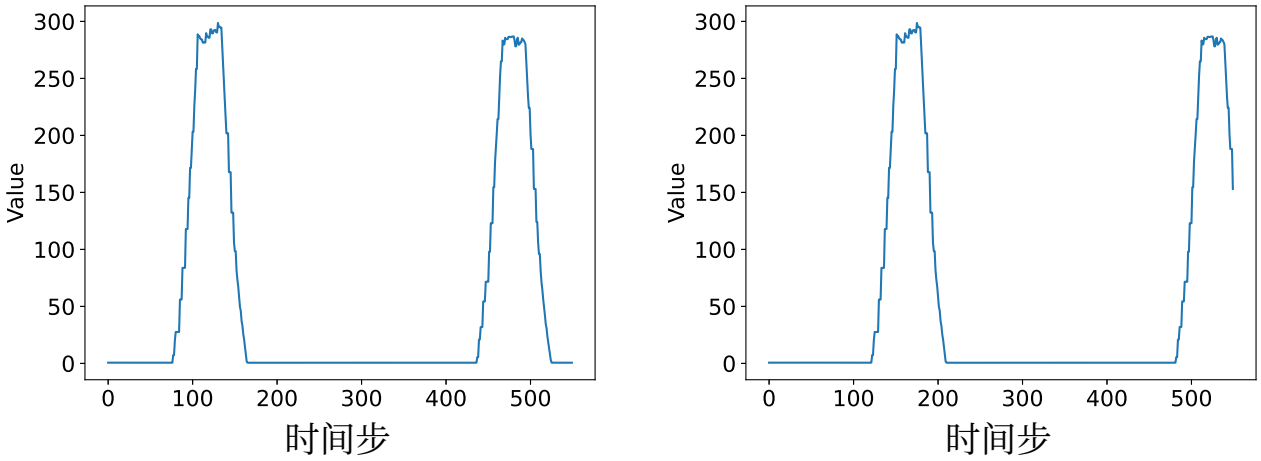
a) **b)**

图*4.* **MixFT 给出的 BizITObs-Application 示例预测，以及混合使用其他子领域的 LoRA 模块或单独使用该模块时给出的预测。** 这两幅图展示了 MixFT、集成 MixFT（见附录 D.2）的预测，以及使用 MixFT 未选中的 LoRA 模块所给出的预测，在图中称为 其他分量 。图中还展示了真实值。在图 **b)** 中，我们未展示集成 MixFT 的预测，因为它与 MixFT 的预测完全相同——MixFT 以 1。0 的概率将给定上下文判定为属于某个子领域。图中表明，使用 MixFT 选中的 LoRA 模块比使用未选中的模块能得到更好的预测。此外，将两种预测进行集成可能比 MixFT 的表现更差，因为对于其子领域而言，其他分量 的预测可能不准确，因为它预测的是一个分布外上下文。

为了考察使用 MixFT 学得的不同 LoRA 模块所产生的不同预测结果，我们在图 4 中展示了两个示例。我们展示了 *MixFT* 的预测值；集成 *MixFT* 的预测值，即对各子域 LoRA 模块给出的预测结果进行加权平均，权重为上下文属于各子域的概率；其他分量 的预测值，即使用 MixFT 未选中子域的 LoRA 模块所给出的预测值；以及真实值。图中显示，MixFT 的表现优于集成 MixFT；如表 4 所示，这一结论总体上也成立，证明了 MixFT 采用硬子域分配的合理性。图 4展示了 MixFT 表现更优的一个原因：使用未被 MixFT 选中的子域的 LoRA 模块时，得到的预测值并不准确。这很可能是因为该上下文对于这个子域而言属于分布外数据；该子域的 LoRA 模块没有在相似上下文上接受训练，因此给出了较差的预测值。因此，将这个较差的预测值与使用分布内子域 LoRA 模块得到的预测值混合，会降低性能。这说明了为什么在 MixFT 中，硬子域分配优于软子域分配。

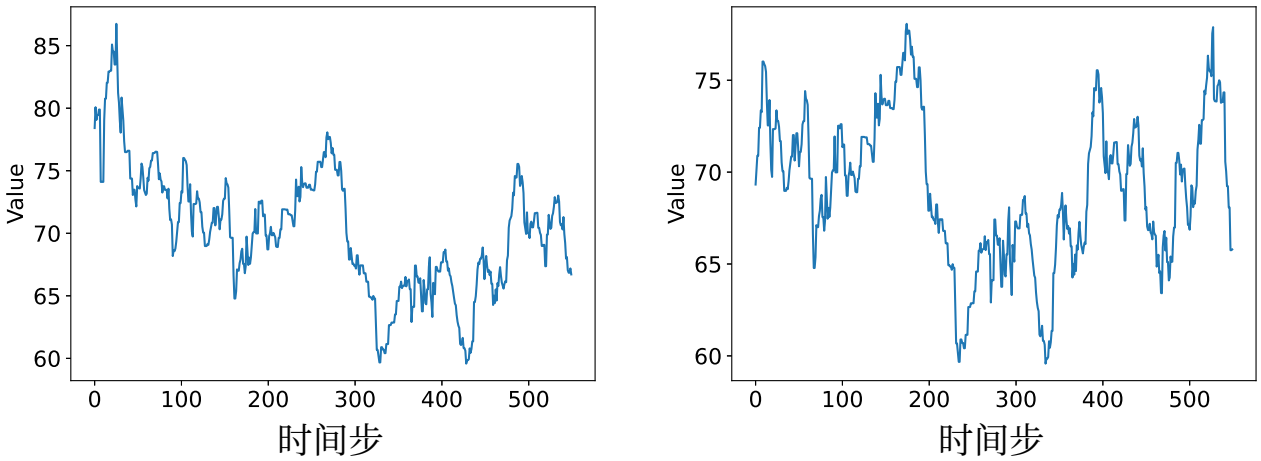
D.5.3. 已识别子域上下文的特征

为了了解 MixFT 混合模型识别出了哪些子域，我们绘制了属于各个子域的上下文示例。示例见图 5和6。在图 5中，我们发现，对于 BizITObs-Service 的第一个通道，属于第一个混合成分的上下文在上下文末尾不会变化（即呈平直线），而属于第二个混合成分的上下文则会变化。这就是图 3中 BizITObs-Service 呈现周期性模式的原因。因此，发现的子域可能部分是根据近期信号波动性划分的。这一点以及其他因素，有助于我们了解 Chronos Bolt 如何为 BizITObs-Service 的第一个通道嵌入上下文：它似乎会将近期波动性相似的上下文归为一组。在图 6中，我们看到并非所有上下文都像图 5那样具有季节性模式；在这种情况下，更难判断每个子域所识别的特征。不过，我们看到图中的一个上下文呈现下降趋势，而另一个则没有，这可能是用于划分子域的一项特征。



a）被识别为来自第一个子域的上下文 **b) 被识别为来自第二个子域的上下文**

图*5.* **分别绘制了在使用 Chronos Bolt 时，微调数据集 BizITObs-service 的第一个通道中被识别为分别属于 MixFT 两个混合成分的两个上下文。** 该图展示了所学子领域的一项特征：当上下文末端发生变化（即图中出现尖峰）时，它被视为属于第二个组件，否则属于第一个组件。这就是为什么在图 3 中，BizITObs-service 存在周期性模式，因为它具有周期性尖峰。该图表明，MixFT 所发现的子领域是可解释且可理解的。



a) 被识别为来自第一个子域的上下文 **b) 被识别为来自第二个子域的上下文**

图*6.* **分别绘制了在使用 Chronos Bolt 时，微调数据集 BizITObs-service 的第八个通道中被识别为分别属于 MixFT 两个混合成分的两个上下文。** 与图 5不同，这些上下文没有清晰的季节性模式。因此，更难识别用于判定子域的特征。不过，趋势可能被用作依据，因为图 **a)** 具有负趋势，而图 **b)** 没有可辨别的趋势。

D.5.4. 零样本预测时 MixFT 的 LoRA 模块选择

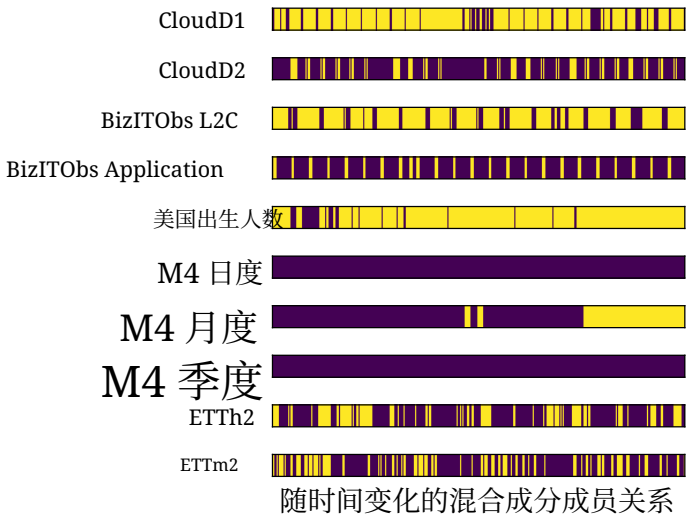


图7. **使用 Chronos Bolt 时，MixFT 评估数据集的混合归属**。该图展示了每个评估数据集的第一个通道中，时间序列被判定为属于第一个混合成分（紫色区域）或第二个混合成分（黄色区域）的情况。正如预期，我们发现，与微调数据相比，混合归属模式的一致性较低（见图3）。这表明从零样本数据中识别子域很困难。不过，数据中仍然存在一些使用模式。例如，对于 BizITObs-Application，我们观察到一种周期性模式。这说明，尽管难以识别零样本混合归属，MixFT 仍然表现不错，这也是它在我们的实验中表现出色的原因之一。