基于动态模型池与集成的多变量时间序列异常检测

Wei Hu、Zewei Yu 和 Jianqiu Xu

中国南京市南京航空航天大学计算机科学与技术学院，邮编 211106 {hwei, yuzewei, jianqiu}@nuaa.edu.cn 通讯作者：Jianqiu Xu

arXiv:2601.02037v1 [cs.LG] 5 Jan 2026

摘要。 多变量时间序列（MTS）异常检测在服务监控、物联网和网络安全等领域至关重要。尽管基于选择或集成的多模型方法优于单模型方法，但仍存在一些局限：（i）选择方法依赖单个选定模型，且对策略较为敏感；（ii）集成方法往往合并所有模型，或仅适用于单变量数据；（iii）大多数方法依赖固定数据维度，限制了可扩展性。为解决这些问题，我们提出 DMPEAD，一种用于多变量时间序列异常检测的动态模型池与集成框架。该框架首先（i）通过参数迁移和多样性度量构建多样化模型池；随后（ii）借助元模型和基于相似度的策略更新模型池，以实现自适应池扩展、子集选择和池合并；最后（iii）在选定子集中通过代理指标排序和 top-k 聚合，对排名靠前的模型进行集成，输出最终异常检测结果。在 8 个真实世界数据集上进行的大量实验表明，我们的方法优于所有基线方法，展现出更强的适应性和可扩展性。

关键词： 多变量时间序列、异常检测、模型池、模型集成、动态更新

1 引言

时间序列是由带时间戳的数据值组成的有序序列，广泛存在于现实世界的各种应用中。随着对时间序列分析需求的不断增长，[7, 8], 异常检测在服务监控、物联网（IoT）和航空航天等领域变得日益重要。时间序列异常表现为异常或非预期行为，需要特别关注，如图 1 所示。

大多数多变量时间序列异常检测方法依赖于单一模型，通过精心设计的架构提取时间和维度特征，例如预测、重构或基于对比学习的方法。然而，没有任何单一的通用模型能够在所有类型的时间序列和异常模式上实现最优性能 [7]。因此，模型选择 和模型集成方法 [1, 2, 6] 利用多个模型，并采用选择或集成策略，以保持稳健的异常检测性能，即使数据分布与原始训练数据集存在显著差异。

然而，现有的选择与集成方法仍存在一些局限：（i）选择方法仅输出单个模型，因此对选择策略高度敏感；（ii）集成方法要么组合所有候选模型 [1] ，要么仅适用于单变量时间序列 [6]；（iii）受候选模型限制，现有方法仍受训练数据集维度的制约，从而限制了可扩展性。

为应对上述挑战，我们提出了**DMPEAD**，即面向多变量时间序列**异常检测**的**动态模型池**与**集成**框架。该框架包含三个步骤：(i) 模型池构建：依次在来自不同领域的不同数据集上训练多个基础模型，采用

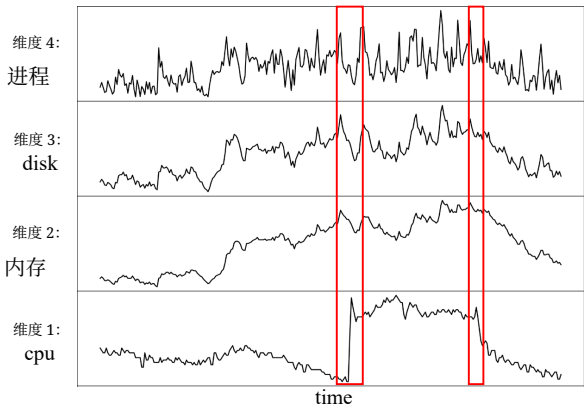


图 1：服务器监控中的多变量时间序列示例，异常以红色标示

参数迁移策略和多样性度量。(ii) 模型池更新：以新的多变量时间序列作为输入，并利用元模型决定是否扩展模型池以及选择合适的模型子集。同时，基于相似度的策略会触发模型池合并，以保持多样性。(iii) 模型集成与异常检测：利用所选模型子集进行评分、代理指标排序、前 k 项聚合和异常阈值设定，最终生成异常检测结果。具体贡献如下：

(1) 我们提出了一种包含构建与更新机制的动态模型池：前者构建初始模型池，以实现广泛的适应性；后者则利用基于元模型的扩展和基于相似度的合并，使模型池能够适应不断变化的数据，同时保持效率。

(2) 我们提出了一种模型集成与异常检测策略，该策略仅集成模型池中的部分模型，而非全部模型，从而提高效率。

(3) 我们在 8 个真实世界多元时间序列数据集上进行了大量实验。结果表明，我们的模型在异常检测性能、效率、泛化能力和参数鲁棒性之间实现了权衡。

2 相关工作

多变量时间序列异常检测可分为单模型方法和多模型方法，后者包括模型选择和集成。

单模型方法包括浅层方法和深度学习方法。浅层方法依赖统计学习、数据挖掘、离群点检测或经典机器学习，而深度方法包括基于预测、重构、对比学习的方法以及通用时间序列模型，例如 Anomaly Transformer [10], DCdetector [11],和 TimesNet [9]。

模型选择方法旨在针对给定的多变量时间序列输入数据，从候选模型集中确定最优模型。根据推理期间执行的模型数量，可分为两类：多模型执行和单模型执行。多模型方法会评估几乎所有候选模型，准确率较高，但计算成本较大，例如 TSADMS [2]。相比之下，单模型方法使用预训练选择器，根据数据特征直接选择一个模型，准确率相对较低，但成本也有所降低。

模型集成方法将候选模型集的输出相结合，以提升异常检测性能，例如 CAE-Ensemble [1] 和 AutoTSAD [6]。前者对所有模型的异常分数取平均，而后者通过对选定模型进行排序和集成来避免全面聚合。我们的方法同样采用排序和集成，但使用动态、与维度无关的模型集，从而降低复杂度并提升可扩展性。

3 问题定义

时间序列 $X = \langle X_1, X_2, \ldots\ldots, X_m \rangle$ 是指由带有时间戳的数据点组成的序列。在多变量时间序列中，每个数据点 X_i 都是一个 n维向量。时间序列异常是指相对于 X的正态分布显著偏离的数据点 $X_i \in X$ 或子序列 $X_{i:j} \in X$ 。

多变量时间序列异常检测方法 M （参数为 θ）以 X 作为输入数据，输出异常分数 $S = \langle S_1, S_2, \ldots\ldots, S_m \rangle$ 或预测标签 $Y = \langle y_1, y_2, \ldots\ldots, y_m \rangle$，其中 $y_i \in \{0, 1\}$表示点 X_i 是异常（1）还是正常（0）。将 S 转换为 Y 涉及阈值选择和异常识别。设定阈值 ε 后，识别过程如下：

$$y_i = \begin{cases} 1 & \text{if } S_i > \varepsilon, \\ 0 & \text{else.} \end{cases} \tag{1}$$

多变量时间序列异常检测模型集成。给定时间序列 X 和一组异常检测方法 方法集 $= \{M_1, M_2, \ldots, M_m\}$，模型集成指的是方法集成 $= \sum W_i M_i(X)$，其中 $M_i(X)$和 W_i 分别表示方法 M_i的异常检测结果及其对应权重，而 $\sum W_i$ 决定集成策略。

4 我们的方法：DMPEAD

如图2所示，所提出的 DMPEAD 框架包含以下三个步骤：(i) 模型池训练与构建，(ii) 模型池测试与更新，以及 (iii) 模型集成与异常检测。

4.1 模型池训练与构建

为了构建初始模型池，我们准备多个训练数据集，并使用基础模型结合参数迁移策略和多样性度量分别对其进行训练。

4.1.1 维度无关重构基础模型 我们选择维度无关的 Transformer 重构模型 [3] 作为模型池和模型集成的基础模型，从而摆脱对固定数据维度的依赖，实现良好的可扩展性。该模型由三部分组成：维度无关输入模块、基于Transformer编码器的骨干网络和输出模块，如图2所示。

该基础模型基于维度独立性和 Transformer 编码器构建，可对每个多变量时间序列维度进行独立重构，从而摆脱对固定维度的依赖。维度独立性使动态模型池机制成为可能，因为该机制要求模型池能够支持维度不同的数据集。此外，该模型还会将每个多变量时间序列维度的原始逐点输入转换为逐段输入，从而降低时间复杂度和空间复杂度。

在训练重构数据。均方误差重构损失被用作训练损失，其中 $\hat{X}$

$$Loss_{\text{MSE}} = \text{MSE}\left(X, \hat{X}\right) = \sum_{i=1}^{m} \text{MSE}\left(X_i, \hat{X}_i\right) \tag{2}$$

4.1.2 初始模型池构建 我们在多个训练用多变量时间序列上依次训练基础模型，以获得相应的训练模型，并将其作为初始模型池的一部分。为加快训练过程，每个模型都会基于前一个模型进行训练，采用参数迁移策略。此外，为确保模型多样性，训练损失不仅包括输入数据与重构数据之间的重构损失，还包括当前训练模型与池中先前训练模型之间的多样性度量 。

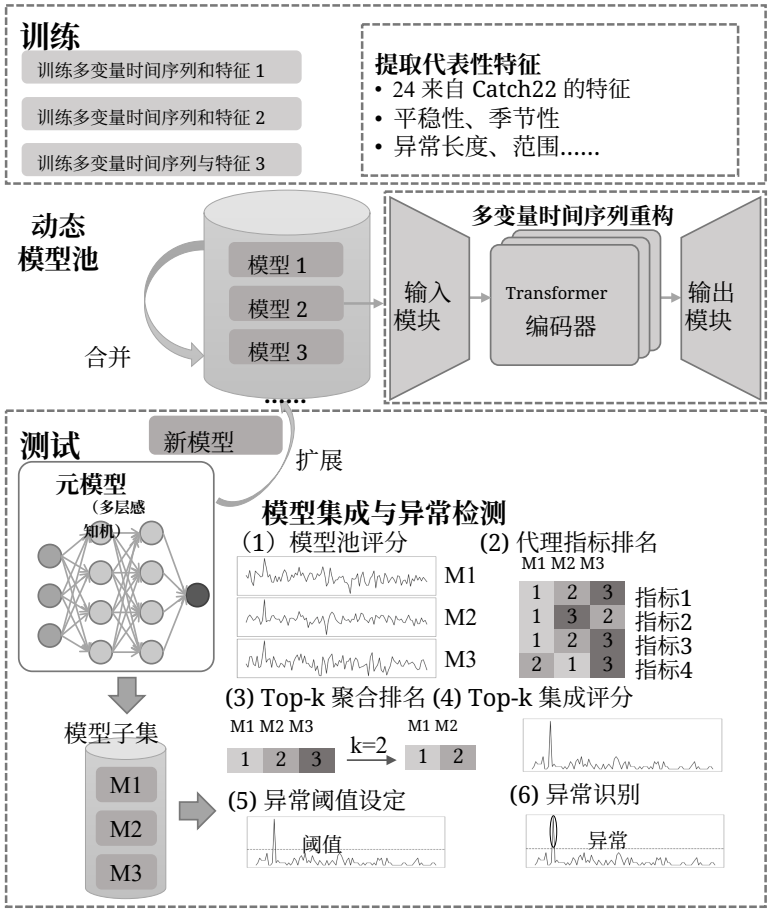


图 2：所提出的 DMPEAD 框架

参数迁移策略。给定一个训练模型 M_{i-1} （参数为 θ_{i-1}），模型 M_i 会从 θ_{i-1}中随机选取因子为 β 的参数，并将其冻结。训练 M_i 期间，仅对剩余因子为 $1-\beta$ 的参数进行迭代更新。与完全顺序训练（即模型训练时不进行参数共享）相比，这种参数迁移策略显著提高了多模型训练效率。不同于 [1], ，我们使用多种数据集训练基础模型，从而生成适用于相应数据集和领域的训练模型。

多样性度量。受 [1], 启发，模型集成方法得益于模型池中包含多样化的基础模型。模型池中的基础模型越多样，模型集成所能达到的准确率就越高。在多变量时间序列异常检测中，模型多样性可体现为异常分数和重构误差。我们定义多样性度量 $Div_{M_i,\ M_j}$ ，用于衡量两个基础模型 M_i 和 M_j 之间的不相似性：

$$Div_{M_i,M_j}(X) = \mathrm{MSE}\left(M_i(X), M_j(X)\right) \tag{3}$$

其中， $M_i(X)$ 和 $M_j(X)$ 表示 M_i 和 M_j 在 X 上的重构数据。$Div_{M_i,\ M_j}(X)$ 越大，意味着 M_i 和 M_j 之间的差异越大。

模型池训练损失。将公式 2 中的重构损失与公式 3 中的多样性度量结合起来，我们可以得到模型池训练损失，其中 μ表示这两个组成部分的重要性：

$$\mathcal{L}_i = Loss_{\mathrm{MSE}} - \mu \sum_{j=1}^{i-1} Div_{M_i,M_j}(X) \tag{4}$$

4.2 模型池测试与更新

自动更新机制通过以下方式在测试期间动态调整模型池：**(i)模型池扩展**，添加新模型；**(ii) 模型池合并**，减少相似模型，从而确保面对输入多变量时间序列发生分布偏移时的可靠性和效率。值得注意的是，池扩展不会重新训练或修改现有模型。

4.2.1 模型池扩展 模型池扩展涉及对新模型创建的决策，以及模型创建和训练策略。

触发新模型创建 最初可以通过直接评估池中模型的输出（例如重构数据或异常分数）来实现。然而，这种方法效率低下且耗时，因为它需要执行所有模型 [6]。因此，我们提出了一种元模型 ，利用数据集特征和模型指纹，只需执行部分模型。我们的方案包括：(i) 构建元模型训练数据；(ii) 训练基于多层感知机的元模型；(iii) 使用元模型选择合适的模型，并对是否创建新模型作出决策；(iv) 同步更新元模型及其训练数据。

构建元模型训练数据包括为每个数据集提取代表性特征并生成模型指纹。代表性特征包括来自 Catch22[4]，的多种特征，如平稳性、季节性和异常信息，涵盖跨维度特征和维内特征。模型指纹表示模型行为和特征，由标准有效多变量时间序列生成，其输出为多变量时间序列重构误差的代表性特征。

基于多层感知机的元模型在上述训练数据上进行训练。为确保可靠性并防止过拟合，我们采用标准策略：K折交叉验证（默认K=5 ）、学习率调度，以及带有耐心值的早停。最终模型选自验证损失最低的折。

我们使用元模型来量化基础模型对新多变量时间序列的适应性。训练完成后，元模型接收新的多变量时间序列作为输入数据，并输出每个基础模型的预测误差。随后，如果某个基础模型的预测误差取反后（例如，模型匹配分数$MS(M_i)$）高于模型匹配阈值 $\varepsilon_{\text{model}}$，即$MS(M_i) > \varepsilon_{\text{model}}$，则认为该基础模型与新的多变量时间序列兼容，并将其加入匹配模型子集$MSet'$。此外，我们还会决定是否触发新模型创建：如果子集$MSet'$ 的大小高于匹配数量阈值 $\varepsilon_{\text{judge}}$，即 $\left|MSet'\right| > \varepsilon_{\text{judge}}$，则认为模型池对新的多变量时间序列具有足够的适应性，因此不创建新模型；否则，将创建并训练一个新模型，以保持模型池的适应性。

考虑到模型池的动态特性，元模型也必须保持自适应。当新的多变量时间序列需要创建新模型时，相应的数据特征和模型指纹会纳入元模型的训练数据，从而触发元模型更新，以适应新多变量时间序列带来的变化。

新模型创建与训练。我们采用与基础模型相同的结构 [3]。训练过程遵循初始模型池构建流程，并引入参数迁移和多样性度量，以帮助保持模型池的动态特性。值得注意的是，我们的参数迁移策略既可以从上一个模型迁移参数，也可以对池中所有模型的权重取平均。

4.2.2 模型池合并 模型池合并包括合并相似模型，并同步更新元模型。

相似模型合并 包含四个步骤：(i) 判断是否触发模型合并；(ii) 根据模型参数计算成对模型差异度；(iii) 使用基于阈值的策略决定是否合并；(iv) 执行合并，并更新模型池及相关组件。

可通过简单的阈值策略触发模型合并：如果模型池大小超过模型合并阈值 $\varepsilon_{\text{merge}}$，即 $|MSet| > \varepsilon_{\text{merge}}$，则启动合并以减少冗余并

降低计算开销。为在一次 MTS 测试期间保持模型池一致性，合并会在测试阶段之前或之后进行。

触发合并时，根据模型参数（θ_i，θ_j）计算池中任意两个模型（M_i，M_j）之间的成对模型差异度，该差异度由三个部分组成：(i) 欧几里得距离；(ii) 统计差异，例如均值、标准差、最小值/最大值；以及 (iii) 余弦差异度，即（$1 -$ 余弦(θ_i，θ_j））。每个部分都会进行归一化，其均值即为模型对（M_i，M_j）的差异度分数DS（M_i，M_j）。

计算差异度分数后，基于阈值的策略会确定需要合并的模型对：如果分数低于差异度分数阈值 $\varepsilon_{\text{disscore}}$，即 DS（M_i，M_j）$< \varepsilon_{\text{disscore}}$，则认为模型 M_i 和 M_j 足够相似，并将其合并。为避免过度压缩（链式合并），每个模型在每轮中最多只能参与一次合并。因此，单次合并所减少的模型数量最多为模型池大小的一半。

对于选定进行合并的模型对（M_i，M_j），将其参数取平均值以形成新模型 M_{new}，即 $\theta_{\text{new}} = (\theta_i + \theta_j)/2$。合并后，从模型池中移除 M_i 和 M_j，并以 M_{new}替代。值得注意的是，由于合并可能进行多轮，M_i 和 M_j 可能是此前轮次合并得到的模型，这表明它们具有更高的相似度。

同步元模型更新。池合并后，元模型训练数据中的代表性数据集特征保持不变，从而保留了新模型创建所产生的增量状态，并维持对模型适应性的广泛洞察。不过，模型池必须更新以反映最新的模型指纹，这会触发元模型训练数据的全面刷新。与池扩展所触发的更新相比，这种更新成本更高，因此需要重新训练元模型。

4.3 模型集成与异常检测

我们在模型池中采用模型集成进行异常检测。与现有方法 [1] 聚合所有基础模型的分数（其中可能包括表现不佳的模型）不同，我们只对所选子集中的前 k 个排序模型进行集成。由于通常无法获得异常标签，我们使用多种代理指标以无监督方式对模型进行排序，从而提高可靠性。前 k 个模型集成与异常检测流程包括（图2）：（i）获取子集中各模型的异常分数；（ii）计算多种代理指标，生成独立的排序结果；（iii）应用稳健排名聚合来合并排序结果、选出前 k 个模型，并对其分数取平均以获得最终异常分数；（iv）根据聚合后的分数确定异常阈值并识别异常。

模型池评分 会计算子集中每个模型的异常分数。异常分数用于量化每个数据点的异常程度，数值越高表示其越有可能是异常。评分计算采用均方误差，与模型重构损失保持一致。

代理指标排序 会计算三类代理指标，以生成多个排序。预测误差 使用 4 种指标衡量输入数据与重构数据之间的差异：均方误差、平均绝对误差、均方根误差和平均绝对百分比误差。合成异常检测 通过 5 种异常类型的合成数据评估模型性能：尖峰、上下文异常、翻转、加速和尺度。模型中心性 反映模型在簇中的相对位置："好"模型通常聚集在中心附近，而"差"模型则分布得更分散。我们通过聚类来衡量中心性，采用 4 种方法：最近邻、k-中心点聚类、亲和传播和贪心最远距离聚类。

Top-k排名聚合 通过鲁棒聚合结合代理指标排序，选出 Top-k 个模型，并对它们的异常分数取平均以生成最终结果。我们采用 Top-k波达排名法，其表现优于其他聚合方法 [2]。如果 k 超过子集 $|MSet'|$ 的大小，则跳过排序和聚合，直接取平均。

异常阈值设定与识别 会根据汇总得分选择异常阈值，并据此识别异常。阈值选择 采用三种方法：(i) 均值-标准差法将阈值设为均值加上 2.5 倍标准差；(ii) epsilon法搜索阈值平衡点和序列异常；(iii) 百分位数法将

阈值设为 s 定的异常比例。异常识别将分数超过阈值的数据点标记为异常，如公式1所示。

5 实验

表1：多变量时间序列异常检测数据集统计信息

数据集	领域	Dim.	多变量时间序列数量	Avg. Len.	AR（%）
Exathlon	监控器	19	31	25,479	8.07
PSM	监控器	25	1	220,322	13.87
SMD	监控器	38	28	25,300	2.10
SMAP	航空航天	25	54	5,313	6.20
MSL	航空航天	55	27	2,445	6.01
SWaT	IoT	51	1	944,919	6.07
WADI	IoT	123	1	957,372	2.88
UAV	飞机	33-41	13	99,400	34.85

我们在来自不同领域的 8 个真实世界多变量时间序列异常检测数据集上评估 DMPEAD，包括服务监控、航空航天、物联网和飞机，如表1所示，其中 UAV 是我们收集的数据集。我们选取 3 项指标进行全面评估：TS-AUC-PR、Range-AUC-PR 和 VUS-PR [5]。我们将模型与 6 种基线方法进行比较：Anomaly Transformer [10], DCdetector [11], TimesNet[9], Patchformer [3], CAE-Ensemble [1],和 TSADMS [2]。

对比实验。由于基线方法依赖将维度作为固定参数，且无人机训练和测试多变量时间序列之间的维度不匹配，该实验使用前7个数据集。具体而言，如表2所示，在来自7个数据集的143个测试多变量时间序列上评估我们的方法和6种基线方法的性能。粗体值表示每个数据集和指标下的最佳性能。对于TS-AUC-PR，我们的方法在Exathlon、SWaT和WADI上取得了最佳结果。对于Range-AUC-PR，我们的方法在PSM、SMD和WADI上表现最佳。对于VUS-PR，我们的方法在PSM和WADI上领先。总体而言，我们的方法在所有数据集和指标中共取得8项最佳结果，优于所有基线方法，展现出强劲的性能。

表2：使用 7 个数据集和 3 项指标进行的性能比较：TS-AUC-PR(TS)、Range-AUC-PR(R)、VUS-PR(V)

数据集	A. Trans. [10]			DCdetect. [11]			TimesNet [9]			Patchfor. [3]			CAE-Ens. [1]			TSADAMS [2]			**Ours**		
	TS	R	V	TS	R	V	TS	R	V	TS	R	V	TS	R	V	TS	R	V	TS	R	V
Exathlon	0.52	0.27	0.27	0.58	0.18	0.16	0.56	0.61	0.63	0.55	0.61	0.63	0.61	**0.64**	**0.66**	0.50	0.33	0.34	**0.61**	0.60	0.61
MSL	0.53	0.22	0.19	0.56	0.19	0.15	**0.58**	0.40	0.37	0.58	**0.40**	**0.38**	0.56	0.28	0.24	0.43	0.31	0.30	0.56	0.35	0.30
PSM	0.64	0.40	0.37	0.26	0.37	0.32	0.64	0.55	0.54	0.64	0.49	0.48	0.64	0.42	0.39	**0.64**	0.53	0.53	0.64	**0.57**	**0.56**
SMAP	0.55	0.15	0.15	0.56	0.18	0.18	**0.58**	**0.26**	**0.24**	0.58	0.24	0.23	0.56	0.20	0.19	0.35	0.21	0.19	0.56	0.24	0.22
SMD	0.44	0.10	0.08	0.29	0.07	0.05	0.49	0.49	**0.45**	0.50	0.48	0.45	**0.52**	0.29	0.25	0.47	0.32	0.31	0.52	**0.50**	0.44
SWaT	0.56	0.21	0.22	0.56	0.31	0.33	0.56	0.11	0.10	0.56	0.10	0.09	0.56	**0.38**	**0.42**	0.06	0.10	0.08	**0.56**	0.10	0.09
WADI	0.14	0.08	0.08	0.53	0.07	0.06	0.27	0.24	0.25	0.08	0.19	0.19	0.53	0.13	0.14	0.03	0.07	0.06	**0.53**	**0.25**	**0.25**

表3：训练和测试阶段的总运行时间

模型	训练运行时间（秒）	测试运行时间（秒）
AnomalyTrans. [10]	**533**	377
DCdetector [11]	49978	1147
TimesNet [9]	11627	2205
Patchformer [3]	3356	**343**
CAEEnsemble [1]	2691	609
TSADAMS [2]	3811	15895
DMPEAD（本文方法）	1362	2146

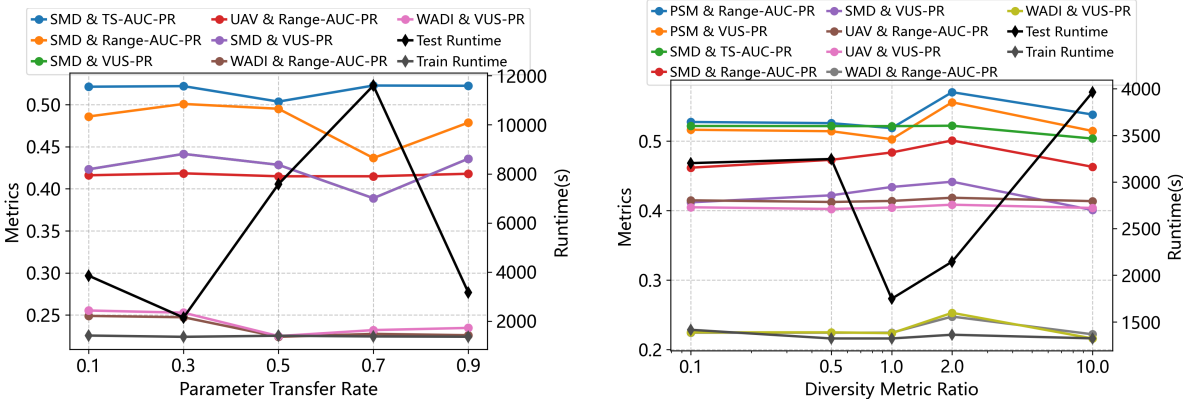
我们还在表3中报告了训练和测试的总运行时间。我们的方法通过参数迁移仅训练一次，在训练效率方面排名第二。在测试阶段，尽管速度略慢，但其性能优于最慢的两种基线方法，这主要得益于元模型和代理指标排序步骤。

消融实验。为阐明各组件对性能的独立贡献，我们针对其主要模块开展消融研究。实验围绕模块级项目展开：(i) 禁用模型池，即不使用模型池，退化为单一模型（即 Patchformer [3]），同时禁用池更新机制和集成；(ii) 禁用池扩展，即不进行扩展，同时禁用池合并，因为初始模型池通常无需合并；(iii) 禁用池合并，即不进行合并。

表4报告了所有数据集在 TS-AUC-PR 和 Range-AUC-PR 指标下，各消融设置的性能和运行时间比较。消融项目的指标结果以相对于未消融项目的差异表示。粗体值表示消融后性能明显下降或运行时间显著增加。不使用模型池 导致性能损失最大，在 PSM（Range-AUC-PR）上达到 7.7%，在 WADI（TS-AUC-PR）上达到 44%；由于失去了加速训练的效果，训练运行时间也大幅增加。不进行扩展 会导致性能适度下降，例如 PSM 在 Range-AUC-PR 指标下约下降 3%。不进行合并 对准确率影响较小，却会显著增加测试运行时间，这表明池合并主要通过减少冗余来提高效率，对准确率影响甚微。总体而言，模型池及其更新机制对于平衡性能与效率都至关重要。

表4：TS-AUC-PR（TS）与Range-AUC-PR（R）指标上的消融结果

消融	Non		无池		无扩展		无合并	
	TS	R	TS	R	TS	R	TS	R
Exathlon	0.61	0.60	**-0.060**	+0.007	0.000	+0.004	0.000	0.000
MSL	0.56	0.35	+0.019	+0.053	0.000	+0.004	0.000	0.000
PSM	0.64	0.57	0.000	**-0.077**	0.000	**-0.032**	0.000	0.000
SMAP	0.56	0.24	+0.013	+0.002	0.000	+0.001	0.000	0.000
SMD	0.52	0.50	**-0.020**	**-0.021**	-0.001	-0.003	0.000	-0.004
SWaT	0.56	0.10	0.000	-0.001	0.000	0.000	0.000	0.000
UAV	0.71	0.42	0.000	-0.001	0.000	-0.003	0.000	-0.001
WADI	0.53	0.25	**-0.444**	**-0.060**	0.000	-0.004	0.000	+0.002
训练时间 (s)	1362		**3356**		1362		1362	
测试时间 (s)	2146		343		2295		**2753**	



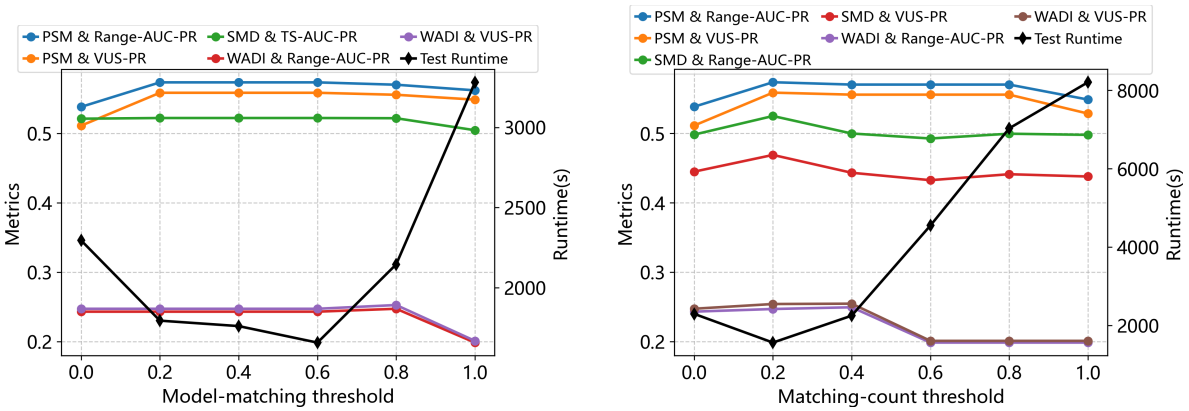
(a) 改变 β，固定 μ　　（b）改变 μ，固定 β

图3：池构建的参数敏感性结果

参数敏感性实验。我们考察了影响模型性能和效率的 3 组超参数：(i) 池构建：参数传递率 β和多样性指标比率 μ；(ii) 池扩展：模型匹配阈值 $\varepsilon_{\text{model}}$ 和匹配数量

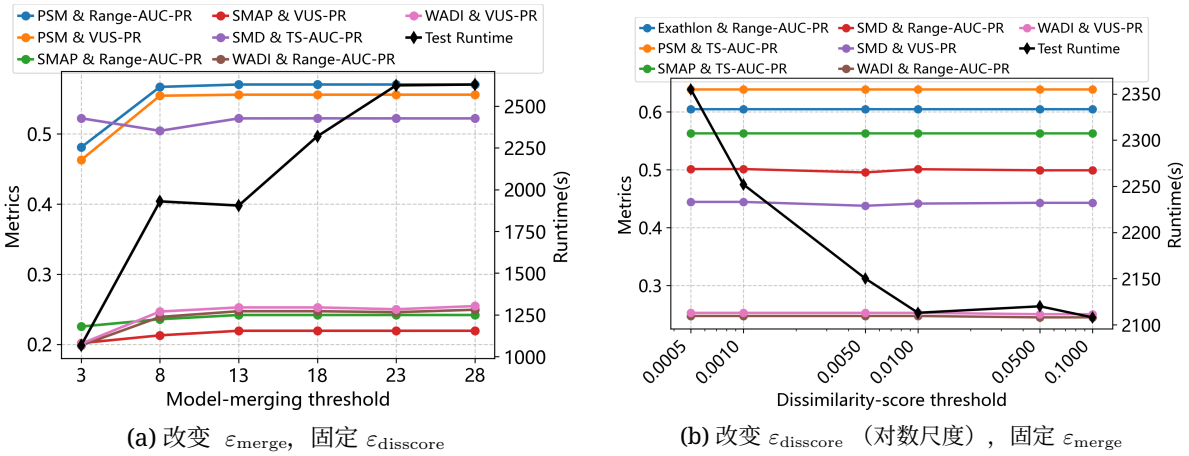
阈值 $\varepsilon_{\text{judge}}$；(iii) 池合并：模型合并阈值 $\varepsilon_{\text{merge}}$ 和差异度分数阈值$\varepsilon_{\text{disscore}}$。鉴于所有参数组合的笛卡尔积过于庞大，我们在每组中采用成对调优策略，固定一个参数并调整另一个参数，以尽可能低的实验成本高效地确定近似最优配置。

图 3 至图 5 展示了若干数据集和指标上的部分结果，其中 X 轴表示参数值，Y 轴表示异常检测指标以及训练/测试总运行时间。对于 β andμ （图 3），最优模型池构建参数为 (0.3,2.0)。对于 $\varepsilon_{\text{model}}$ and $\varepsilon_{\text{judge}}$ （图 4），最优模型池扩展参数为 $(0.8, 0.34 * |MSet|)$。对于 $\varepsilon_{\text{merge}}$ and $\varepsilon_{\text{disscore}}$ （图 5），最优模型池合并参数为 (15,0.01)。



(a) 改变 $\varepsilon_{\text{model}}$，固定 $\varepsilon_{\text{judge}}$ (b) 改变 $\varepsilon_{\text{judge}}$ （需要 $|MSet|$ 次），固定 $\varepsilon_{\text{model}}$

图 4：模型池扩展的参数敏感性结果



(a) 改变 $\varepsilon_{\text{merge}}$，固定 $\varepsilon_{\text{disscore}}$ (b) 改变 $\varepsilon_{\text{disscore}}$ （对数尺度），固定 $\varepsilon_{\text{merge}}$

图 5：池合并的参数敏感性结果

可扩展性实验。为评估模型的可扩展性，即泛化到未见数据的能力，我们在前 7 个数据集上进行留一数据集实验。对于每个目标数据集，模型池在其余 7 个数据集（包括 UAV）上构建，并在留出的测试多变量时间序列上进行评估。结果与以下方法进行比较：(i) 在目标数据集上训练和测试的 6 种基线方法（通过箱线图展示均值和中位数）；(ii) 使用目标数据集训练的标准 DMPEAD（“DMPEAD”，灰色虚线）。可扩展性变体（“DMPEAD Scal.”，黑色虚线）仅在未见数据上进行评估。

黑色虚线超过基线方法的均值/中位数，表明 DMPEAD 无需领域内训练，便能优于至少一半的基线方法，展现出强大的跨数据集适应性。当它甚至超过灰色虚线时，则凸显了其卓越的泛化能力。图 6 展示了

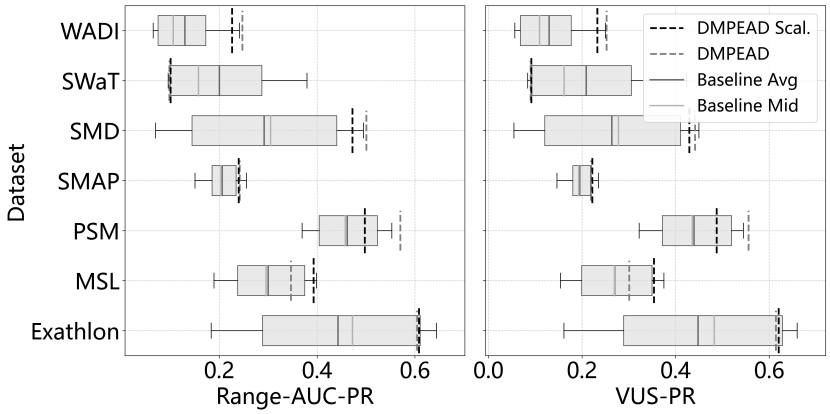


图 6：训练期间在未见数据上的可扩展性结果

对于 Range-AUC-PR 和 VUS-PR，“DMPEAD Scal.”在除 SWaT 之外的所有数据集上始终优于基线均值/中位数，并在 MSL 和 Exathlon 上胜过“DMPEAD”，这证实 DMPEAD 在未见数据上表现稳健，凸显了其实用可扩展性。

6 结论

我们提出了一种用于多变量时间序列异常检测的动态模型池与集成框架，可针对不同的多变量时间序列自适应地更新模型并进行集成。在异常检测对比、消融研究、参数敏感性和可扩展性方面开展的大量实验，验证了该框架的适应性和泛化能力。

参考文献

1. Campos, D., Kieu, T., Guo, C., Huang, F., Zheng, K., Yang, B., Jensen, C.S.：利用多样性驱动的卷积集成进行无监督时间序列离群点检测。VLDB基金会论文集**15**3494142 2 3 611 623 2021 10 14778 3494124 ( ), – ( ). https://doi.org/ . / . . Goswami, M., Challu, C.I., Callot, L., Minorics, L., Kan, A.：用于时间序列异常检测的无监督模型选择。国际学习表征会议（2023）3. Hu, W., Xu, J., Zhao, X.：利用维度独立性和重构进行多变量时间序列数据异常检测。2025年第8届国际数据存储与数据工程会议论文集。第15页 3725475 4 22 2025 10 1145 3725472 – . ACM（）。https://doi.org/ . / . . Lubba, C.H., Sethi, S.S., Knaute, P., Schultz, S.R., Fulcher, B.D., Jones, N.S.：Catch22：典型时间序列特征。《数据挖掘与知识发现》**33**(6), 1821–1852（2019）。https://doi.org/ 10.1007/s10618-019-00647-x 5. Paparrizos, J., Boniol, P., Palpanas, T., Tsay, R.S., Elmore, A., Franklin, M.J.：曲面下体积：一种新的时间序列异常检测准确率评估指标。VLDB基金会论文集 **15**3551830 6 11 2774 2787 2022 10 14778 3551793 ( ), – ( ). https://doi.org/ . / . . Schmidl, S., Naumann, F., Papenbrock, T.：AutoTSAD：面向时间序列数据的无监督整体异常检测。VLDB基金会论文集**17**(11), 2987–3002（2024）。https://doi.org/10.14778/ 3681954.3681978 7. Schmidl, S., Wenig, P., Papenbrock, T.：时间序列异常检测：全面评估。VLDB基金会论文集**15**(9), 1779–1797（2022）。https://doi.org/10.14778/3538598. 3538602 8. Tong, Y., She, J., Ding, B., Wang, L., Chen, L.：空间众包中的在线移动微任务分配。2016年IEEE第32届国际数据工程会议。第49–60页。IEEE（2016）9. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.：TimesNet：面向通用时间序列分析的时间二维变化建模。国际学习表征会议（2023）

10. Xu, J., Wu, H., Wang, J., Long, M.：Anomaly Transformer：利用关联差异进行时间序列异常检测。国际学习表征会议论文集（2022）11. Yang, Y., Zhang, C., Zhou, T., Wen, Q., Sun, L.：DCdetector：用于时间序列异常检测的双重注意力对比表征学习。第29届ACM SIGKDD知识发现与数据挖掘大会论文集。第3033–3045页。ACM（2023）。https://doi.org/10.1145/3580305. 3599295