基于机器学习的小样本空气质量预测

李佳璇

辽宁师范大学 辽宁 大连 116029

【摘 要】：空气污染精准预测对公共健康预警和环境政策制定具有重要意义，但细颗粒物（PM2.5）浓度预测面临历史数据匮乏和区域间气象-排放模式差异显著的双重挑战，传统深度学习方法在小样本场景下泛化能力严重不足。针对这一问题，本文提出了一种基于模型无关元学习（MAML）的全连接神经网络预测模型（MAML-ANN），旨在学习跨城市的通用知识，实现对新目标城市仅用极少量观测样本即可快速、精准适配。在方法设计上，首先构建了一个维度为 23 的输入特征向量，系统融合了气象特征、多尺度滞后特征、时序差分、周期性时间编码及城市嵌入信息，其中城市嵌入层用于捕获不同城市的固有静态属性。模型主体采用三层全连接网络作为基学习器，参数量控制在 1.5 万以内，与 MAML 的小样本场景高度匹配。元学习器负责在多个源域城市上执行双层优化：内循环通过单步梯度下降在支持集上快速生成任务专属参数，外循环则在查询集上优化元损失以更新全局初始化参数。模型集成了权重衰减、早停、梯度裁剪及学习率调度等多重正则化策略，有效抑制了小样本下的过拟合风险。实验在多个城市空气质量数据集上验证了所提模型的有效性。结果表明，MAML-ANN 在仅有 5 至 50 个训练样本的极端小样本条件下，仍能取得优于传统预训练-微调及独立训练方案的预测精度，且计算复杂度适中。该模型为数据稀缺区域的高精度空气质量预测提供了一种高效、可行的解决方案，其元学习框架亦可推广至其他环境监测时序预测任务。空气污染精准预测对公共健康预警和环境政策制定具有重要意义，但细颗粒物（PM2.5）浓度预测面临历史数据匮乏和区域间气象-排放模式差异显著的双重挑战，传统深度学习方法在小样本场景下泛化能力严重不足。针对这一问题，本文提出了一种基于模型无关元学习(MAML)的全连接神经网络预测模型(MAML-ANN)，旨在学习跨城市的通用知识，实现对新目标城市仅用极少量观测样本即可快速、精准适配。

【关键词】：小样本学习；模型无关元学习；空气质量预测；跨城市知识迁移

Small-Sample Air Quality Prediction Based on Machine Learning

Li Jiaxuan

School of Mathematics,Liaoning Normal University.Dalian 116029

【Abstract】:Accurate air pollution forecasting is of critical importance for public health early warning systems and environmental policy formulation. However, the prediction of fine particulate matter (PM2.5) concentrations faces two fundamental challenges: the scarcity of historical observations in many regions and the substantial heterogeneity in meteorological conditions and emission patterns across different cities. Conventional deep learning approaches suffer from severe generalization degradation under few-shot scenarios. To address these issues, this paper proposes a Model-Agnostic Meta-Learning-based Multilayer Perceptron (MAML-ANN) model, which aims to learn transferable cross-city knowledge and enable rapid, accurate adaptation to a new target city with only a limited number of observed samples.In terms of methodological design, we first construct a 23-dimensional input feature vector that systematically integrates meteorological features, multi-scale lagged features, temporal differences, periodic temporal encodings, and city-specific embeddings. The city embedding layer is designed to capture the inherent static attributes of different cities. The base learner is implemented as a three-layer fully connected network with approximately 14.27 thousand parameters, which is well-suited to the few-shot setting of MAML. The meta-learner performs bi-level optimization across multiple source-domain cities: the inner loop conducts single-step gradient descent on the support set to rapidly generate task-specific parameters, while the outer loop optimizes the meta-loss on the query set to update the globally shared initialization parameters. Multiple regularization strategies,

including weight decay, early stopping, gradient clipping, and learning rate scheduling, are incorporated to effectively mitigate overfitting risks under small-sample conditions. The proposed model is validated on air quality datasets from multiple cities. Experimental results demonstrate that under extreme few-shot conditions with only 5 to 50 training samples, MAML-ANN consistently achieves superior prediction accuracy compared to conventional pre-training-fine-tuning approaches and independently trained models, while maintaining moderate computational complexity. The proposed framework offers an eff icient and feasible solution for high-precision air quality forecasting in data-scarce regions, and the meta-learning paradigm can potentially be extended to other time-series forecasting tasks in environmental monitoring.

【Keywords】:Few-shot learning;Model-Agnostic Meta-Learning (MAML);Air quality prediction;Cross city knowledge transfer

1 问题建模

本节从任务构建、数据划分和特征编码三个层面对小样本空气质量预测问题进行形式化建模，为后续 MAML-ANN 模型[1]的训练与评估奠定数学基础。

1.1 任务构建策略

在 MAML 框架中，模型通过在多个任务上的元训练来学习通用的初始化参数。每个任务 T_i 被设计为一个独立的“城市-时段”空气质量预测子问题。具体而言，设共有 M 个城市或监测站点的数据，每个城市 c 的数据包含长度为 T_c 的连续时间序列 $D_c = \{(\mathrm{u}_t, y_t)\}_{t=1}^{T_c}$，其中 $\mathrm{u}_t \in$ 耽^d 为第 3.1.2 节定义的多维输入特征向量，$y_t \in$ 耽为预测目标(如PM2.5 浓度)。

本文采用跨城市任务划分策略：将每个城市的数据视为一个独立任务，即 T_i 对应于城市 i 的空气质量预测问题。与基于时间滑窗的任务划分相比，跨城市划分具有以下优势：其一，不同城市的气象条件、污染源结构和地理特征存在本质差异，这种天然的任务异质性更有利于 MAML 学习通用特征表示；其二，跨城市划分避免了同一城市时序数据中样本间的强相关性，使得任务间的独立性假设更加合理；其三，该策略直接对应小样本预测的应用场景——模型在多个数据相对充足的城市上完成元训练后，快速适配到仅有少量观测数据的新建监测站。

设所有城市构成任务集合 $\boldsymbol{T} = \{T_1, T_2, \ldots, T_M\}$。在元训练阶段，从 $\boldsymbol{T}$ 中采样一批任务 $\{T_i\}_{i=1}^{B}$，其中 B 为批次大小（batch size）。每个任务 T_i 的数据划分为支持集 S_i 和查询集 Q_i，其划分方式详见 3.1.2 节。

1.2 支持集与查询集的划分机制

MAML 的训练依赖于每个任务内部支持集与查询集的划分。支持集 S_i 用于内循环的任务级适配——即利用少量样本更新模型参数以获得任务专属参数 θ'_i；查询集 Q_i 用于外循环的元损失计算——评估适配后的模型在未见样本上的表现，并据此更新元参数 θ。

对于时间序列预测问题，支持集和查询集的划分不能采用随机抽样，因为时序样本之间存在严格的前后依赖关系。若随机划分，会出现未来信息“泄漏”到训练集中的问题，导致模型评估失真。本文采用时序前向划分策略：将每个城市的时间序列按时间顺序划分为前后两段，前段作为支持集，后段作为查询集。

设城市 i 的时间序列长度为 T_i，划分比例为 $\rho \in (0, 1)$，则：

$$S_i = \{(\mathrm{u}_t, y_t) \mid t = 1, 2, \ldots, \lfloor \rho T_i \rfloor\} \tag{1}$$

$$Q_i = \{(\mathrm{u}_t, y_t) \mid t = \lfloor \rho T_i \rfloor + 1, \ldots, T_i\} \tag{2}$$

其中 $\lfloor \cdot \rfloor$ 为向下取整函数。支持集的大小记为 $N_S = |S_i|$，查询集的大小记为 $N_Q = |Q_i|$。在小样本场景下，支持集的样本数量应严格控制——典型设置为 $N_S \in \{5, 10, 20, 50\}$，以模拟目标域只有极少量历史观测数据的实际情况。

需要说明的是，上述划分是在元训练阶段每个任务内部进行的。在元测试阶段，对于目标城市，仅使用其少量支持集样本进行适配，然后在整个测试周期上评估预测性能。

1.3 时序特征编码模块

空气质量时间序列具有复杂的时序依赖性和周期性模式，直接以原始数值输入全连接网络难以有效捕捉这些特征。为此，本文设计了一套面向小样本场景的时序特征编码方案，对时间戳信息和时序上下文进行结构化编码。

（1）周期性编码（Periodic Encoding）

空气质量受日周期和季节周期的显著影响——早晚交通高峰时段污染物浓度升高，冬季供暖期间PM2.5浓度明显高于夏季。为将这种周期性信息显式地输入模型，本文采用三角循环编码（Cyclic Encoding）对时间特征进行映射。

对于小时特征 $h \in \{0, 1, \ldots, 23\}$，编码为二维向量：

$$h_{\sin} = \sin\left(\frac{2\pi h}{24}\right) \quad (3)$$

$$h_{\cos} = \cos\left(\frac{2\pi h}{24}\right) \quad (4)$$

类似地，对于星期特征 $d \in \{0, 1, \ldots, 6\}$，编码为：

$$d_{\sin} = \sin\left(\frac{2\pi d}{7}\right) \quad (5)$$

$$d_{\cos} = \cos\left(\frac{2\pi d}{7}\right) \quad (6)$$

三角循环编码相比直接使用整数表示或独热编码的优势在于：其一，将时间点映射到单位圆上的连续空间，保留了时间点之间的循环距离关系（如 23 时与 0 时的距离天然较近）；其二，编码维度远低于独热编码，有利于缓解小样本下的维度灾难问题。

（2）滞后特征构造（Lag Feature Construction）

由于小样本场景下可用样本数量有限，直接构建复杂的循环神经网络可能面临严重的过拟合。本文采用滞后特征（lag features）的方式，将时序上下文以增广特征的形式融入输入向量，从而以全连接网络的简洁结构捕获时序依赖性[2]。

设滞后步长为 $L = \{1, 2, 3, 7, 14\}$，分别对应前 1 小时、前 2 小时、前 3 小时、前 7 天（同一时刻）、前14 天（同一时刻）的历史观测值。对于时刻 t 的样本，构造滞后特征向量：

$$z_t = [y_{t-1}, y_{t-2}, y_{t-3}, y_{t-168}, y_{t-336}] \in \mathbb{R}^5$$

其中 y_{t-168} 和 y_{t-336} 分别表示7 天前和 14 天前同一时刻的污染物浓度（以小时为单位，$24 \times 7 = 168$，$24 \times 14 = 336$）。选择这些特定滞后步长兼顾了短期依赖（1-3 小时）和长期周期依赖（7/14 天），能够以极低的参数成本为模型提供丰富的时序上下文信息。

（3）差分特征（Diferential Features）

除绝对浓度值外，浓度的变化率同样蕴含重要信息——浓度快速上升往往预示污染事件的临近。本文构造一阶差分特征：

$$\Delta y_{t-1} = y_{t-1} - y_{t-2}$$

表示前一小时浓度的变化量。该特征以极简的形式为模型提供了“趋势”信息，在小样本条件下是一种高效的先验知识注入方式。

1.4 完整输入特征向量

综合前述编码方案，时刻 t 的完整输入特征向量 $u_t \in \mathbb{R}^d$ 由以下五部分拼接构成：

$$u_t = \text{Concat}\left(x_{\text{气象}}, \underbrace{x^{\text{lag}}}_{\text{滞后特征}}, \underbrace{x^{\text{dif}}_t}_{\text{差分特征}}, \underbrace{x^{\text{time}}_t}_{\text{时间编码}}, \underbrace{x_{\text{静态}}}_{\text{静态}}\right)$$

各部分的具体构成如表 3.1 所示。

Table 1 表 3.1：输入特征向量构成

特征类别	具体变量	维度
气象特征	温度、湿度、风速、气压、降水量	5
滞后特征	y_{t-1}，y_{t-2}，y_{t-3}，y_{t-168}，y_{t-336}	5
差分特征	Δy_{t-1}	1
时间编码	小时 sin / cos、星期 sin / cos	4
静态特征	城市/站点编码（独热或嵌入）	M（或嵌入维度）

其中城市/站点编码用于区分不同城市的固有特征（如海拔、气候类型、工业结构等），在元训练阶段帮助模型学习城市间的共性与差异。实际应用中，若城市数量 M 较大，可采用嵌入层（Embedding Layer）将城市编码映射为低维稠密向量，以避免独热编码带来的维度爆炸。本文采用嵌入维度 $d_{\text{emb}} = 8$，将城市编码转化为8 维连续向量。

综上所述，完整输入特征向量的维度为 $d = 5 + 5 + 1 + 4 + 8 = 23$。这一维度处于适中水平，使得3 层全连接网络的参数量控制在可接受范围内，与 MAML 的小样本场景相匹配。第 3.2 节将在此基础上进一步阐述 MAML-ANN 模型的完整架构设计。

2 模型架构设计

本节在前述问题建模的基础上，详细阐述 MAML-ANN 模型的完整架构设计。模型由三个核心模块组成：输入编码层负责将原始特征映射为适合神经网处理的形式；基学习器（Base Learner）采用全连接神经网络作为任务级适配的主体；MAML 元学习器（Meta Learner）负责学习跨任务的通用初始化参数。三个模块协同工作，构成了完整的小样本空气质量预测模型。

2.1 总体架构概述

本文提出的 MAML-ANN 模型采用“元训练-元测试”两阶段架构，如图 3.1 所示。在元训练阶段，模型在多个源域城市上执行多任务元学习，通过内循环和外循环的双层优化策略，获得一组通用的初始化参数 Θ^*；在元测试阶段，对于仅有少量历史数据的目标城市，从 Θ^* 出发在支持集上进行少量梯度更新，即可快速适配该城市的预测任务。

模型的数学形式可表述为：设基学习器为 f_θ，其参数为 θ。MAML 元学习器的目标是学习一组最优初始参数 θ^*，使得对于任意任务 $T_i \sim p(T)$，经过 K 步内循环更新后得到的任务专属参数 θ_i' 在查询集上取得最小损失：

$$\theta^* = \arg\min_\theta \mathbb{E}_{T_i \sim p(T)} \left[L_{\mathcal{T}_i^{\text{query}}}\left(f_{\theta_i'}\right) \right], \quad \text{s.t.} \quad \theta_i' = \theta - \alpha \nabla_\theta L_{\mathcal{T}_i}^{\text{support}}(f_\theta) \quad (7)$$

其中 α 为内循环学习率，L^{support} 和 L^{query} 分

别为支持集和查询集上的损失函数。

2.2 输入编码层设计

输入编码层负责将原始输入特征$u_t \in$ 耽^{23}（参见 3.1.4 节）转换为神经网络的输入向量。编码层包含以下三个子模块：

（1）特征标准化模块

不同特征的量纲和数值范围差异较大（如温度在-20° C 到 40° C 之间，PM2.5 浓度在 0 到 500 $\mu g/m^3$之间），直接输入网络会导致训练不稳定。本文采用 Z-score 标准化方法，将每个特征维度转换为均值为0、标准差为 1 的标准分布：

$$\tilde{x}^{(j)} = \frac{x^{(j)} - \mu_j}{\sigma_j + \epsilon} \tag{8}$$

其中 μ_j 和 σ_j 分别为第 j 个特征在训练集上的均值和标准差，$\epsilon = 10—^8$ 为防止除零的小常数。标准化参数在元训练阶段从所有源域数据中计算得到，并在元测试阶段直接应用于目标域数据，确保数据预处理的一致性。

（2）城市嵌入模块

对于城市/站点编码特征，本文采用嵌入层（Embedding Layer）将其映射为稠密向量。嵌入层本质上是一个可训练的查找表 $E \in$ 耽$^{M \times d_{\text{emb}}}$，其中 M 为城市总数，$d_{\text{emb}} = 8$ 为嵌入维度。对于城市 c，其嵌入向量为：

$$\mathbf{e}_c = E[c,:] \in 耽^8 \tag{9}$$

嵌入层与网络主体联合训练，在元学习过程中自动学习城市间的相似性结构——气象和污染特征相似的城市在嵌入空间中距离更近，从而促进跨城市知识迁移。

（3）特征拼接与正则化

将标准化后的气象特征、滞后特征、差分特征、时间编码与城市嵌入向量沿特征维度拼接，得到编码层的最终输出：

$$\mathbf{x}_t = \text{Concat}\left(\tilde{\mathbf{x}}_t^{\text{meteor}}, \tilde{\mathbf{x}}_t^{\text{lag}}, \tilde{\mathbf{x}}_t^{\text{diff}}, \tilde{\mathbf{x}}_t^{\text{time}}, \mathbf{e}_c\right) \in \mathbb{R}^{23} \tag{10}$$

编码层输出维度为 23，与基学习器的输入维度相匹配。在编码层末端，本文采用批次归一化（Batch Normalization, BN）对输出进行进一步规范化，以加速训练收敛并增强模型对小样本扰动的鲁棒性。

2.3 基学习器网络结构

基学习器（Base Learner）是模型中进行任务级预测的主体网络，采用全连接神经网络（Multilayer Perceptron, MLP）架构。选择 MLP 而非 LSTM 等时序模型的原因在于：其一，MLP 参数量较小，与MAML 的小样本场景相匹配，能够有效避免过拟合；其二，3.1.3 节构造的滞后特征已充分编码了时序依赖关系，MLP 足以捕获这些特征中的非线性映射。

基学习器的网络结构如表 3.2 所示：

Table 2 表 3.2：基学习器网络结构

层序号	层类型	输入维度	输出维度	激活函数
1	全连接层	23	64	ReLU
2	Dropout 层	64	64	_
3	全连接层	64	64	ReLU
4	Dropout 层	64	64	_
5	全连接层	64	64	ReLU
6	Dropout 层	64	64	_
7	全连接层	64	1	Linear

网络包含 3 个隐藏层，每层 64 个神经元，逐层堆叠以提取高层次特征表示。隐藏层采用 ReLU 激活函数，其数学表达式为：

$$\text{ReLU}(z) = \max(0, z) \tag{11}$$

ReLU 函数具有计算简单、梯度非饱和等优点，能够有效缓解深度网络中的梯度消失问题。输出层为线性层，直接输出 PM2.5 浓度的连续预测值，不使用激活函数。

Dropout 层的配置：在每个隐藏层后均接入 Dropout 层，丢弃率设置为$p = 0.3$。Dropout 层在训练时随机丢弃部分神经元，迫使网络学习冗余特征表示；在测试时所有神经元均参与计算，但输出乘以 (1_p)以保持期望一致。Dropout 的引入对缓解小样本下的过拟合至关重要——在仅有数十个支持集样本的条件下，Dropout 能够显著降低训练误差与测试误差之间的差距。

网络参数量分析：该网络的总参数量为：

$$|\theta| = (23 \times 64 + 64) + 3 \times (64 \times 64 + 64) + (64 \times 1 + 1) = 14,273 \tag{12}$$

其中第一项为输入层到隐藏层 1 的权重大小，第二项为三个隐藏层间的权重大小，第三项为隐藏层 3到输出层的权重大小。不足 1.5 万的参数量在深度模型中属于极轻量级，理论上仅需 $O(10^2)$ 量级的样本即可有效训练，与 MAML 的快速适配需求相匹配。

2.4 MAML 元学习器设计

元学习器是 MAML-ANN 模型的核心组件，负责跨任务学习通用的初始化参数。其设计包含以下关键要素：

（1）元参数的组织形式

元学习器维护的元参数 Θ 与基学习器的网络参数 θ 具有完全相同的结构和维度，即

$$\Theta = \{W^{(1)}, b^{(1)}, \ldots, W^{(4)}, b^{(4)}\}$$

共 14,273 个参数。MAML 的训练目标是优化这些参数，使其成为适用于多任务快速适配的通用初始点。

（2）内循环更新机制

在元训练的每个任务 $\mathcal{T}_i$ 上，元参数 Θ 通过内循环在支持集上更新 K 步，得到任务专属参数 θ_i'。本文采用 $K=1$ 的更新策略，即单步梯度下降，这一设置基于以下考虑：其一，单步更新在计算上最为高效，使元训练能够在有限时间内容纳更多的任务采样；其二，单步更新迫使元参数落在多个任务损失曲面的“最陡下降方向交汇处”，从而最大化初始参数的通用性；其三，Finn 等人的实验表明，对于小样本回归任务，$K=1$ 已能取得与多步更新相当的精度[3]。

单步更新的具体形式为：

$$\theta_i' = \Theta - \alpha \nabla_\Theta \frac{1}{|\mathcal{S}_i|} \sum_{(\mathbf{x},y)\in\mathcal{S}_i} \left(f_\Theta(\mathbf{x}) - y\right)^2 \tag{13}$$

其中 α 为内循环学习率，本文取值为 $\alpha = 0.01$。

（3）外循环元优化

外循环在查询集上计算元损失，并通过二阶梯度（即梯度对梯度求导）更新元参数。元损失定义为所有采样任务在查询集上损失的平均值：

$$\mathcal{L}_{\text{meta}}(\Theta) = \frac{1}{B}\sum_{i=1}^{B} \frac{1}{|\mathcal{Q}_i|} \sum_{(\mathbf{x},y)\in\mathcal{Q}_i} \left(f_{\theta_i'}(\mathbf{x}) - y\right)^2 \tag{14}$$

其中 B 为批次大小。元参数的更新采用 Adam 优化器：

$$\Theta \leftarrow \Theta - \beta \nabla_\Theta \mathcal{L}_{\text{meta}}(\Theta) \tag{15}$$

其中 β 为外循环学习率，本文取值为 $\beta = 0.001$。采用 Adam 替代标准 SGD 的优势在于其自适应学习率调整机制，能够在元训练的不同阶段自动调节步长，提高训练的稳定性和收敛速度。

（4）二阶导数的计算

外循环梯度 $\nabla_\Theta \mathcal{L}_{\text{meta}}(\Theta)$ 涉及对 θ_i^g 的链式求导，而 θ_i^g 本身是 Θ 的函数，因此需要计算二阶导数。具体地，对于单个任务，元损失对 Θ 的梯度为：

$$\nabla_\Theta \mathcal{L}_{\text{meta}} = \nabla_{\theta_i'} \mathcal{L}_{\text{query}} \cdot \left(I - \alpha \nabla^2_\Theta \mathcal{L}_{\text{support}}\right) \tag{16}$$

其中 I 为单位矩阵，$\nabla^2_\Theta \mathcal{L}_{\text{support}}$ 为支持集损失的 Hessian 矩阵。计算完整的 Hessian 矩阵在计算上代价高昂，本文采用一阶近似（First-order Approximation）策略，忽略二阶导数项，将梯度简化为：

$$\nabla_\Theta \mathcal{L}_{\text{meta}} \approx \nabla_{\theta_i'} \mathcal{L}_{\text{query}} \tag{17}$$

这一近似虽然在理论上损失了一定的精度，但在实践中已被证明能够取得与完整二阶 MAML 相当的泛化性能，且计算效率显著提升，尤其适用于参数量适中的回归[4]。

2.5 正则化策略与训练稳定性

为增强模型在小样本场景下的泛化能力并确保训练稳定性，本文在模型设计中集成了以下正则化策略：

（1）权重衰减（L2 正则化）

在元损失函数中引入权重衰减项，对基学习器的所有可训练参数施加 L2 范数惩罚：

$$\mathcal{L}_{\text{meta}}^{\text{reg}} = \mathcal{L}_{\text{meta}} + \lambda \sum_{l=1}^{4} \left(\|\mathbf{W}^{(l)}\|_F^2 + \|\mathbf{b}^{(l)}\|_2^2\right) \tag{18}$$

其中 $\lambda = 1 \times 10^{-4}$ 为正则化系数，$\|\cdot\|_F$ 为 Frobenius 范数。权重衰减通过约束参数幅值来抑制过拟合，其本质是降低了模型对训练样本中噪声的敏感性。

（2）早停机制（Early Stopping）

在元训练过程中，预留一个验证任务集（从源域城市中随机抽取 20% 的城市不参与训练），在每个元迭代轮次后评估模型在验证任务上的元损失。当验证损失连续 20 轮未下降时，提前终止训练并恢复最佳模型参数，防止过度优化导致的泛化性能退化。

（3）梯度裁剪（Gradient Clipping）

为防止梯度爆炸，在外循环更新前对元梯度进行裁剪，将其范数限制在阈值 $\gamma = 1.0$ 以内：

$$\mathbf{g} \leftarrow \begin{cases} \gamma \cdot \mathbf{g} / \|\mathbf{g}\|_2, & \text{if } \|\mathbf{g}\|_2 > \gamma \\ \mathbf{g}, & \text{otherwise} \end{cases} \tag{19}$$

梯度裁剪确保每次更新的步长不会过大，从而维持训练的稳定性。

（4）学习率调度

内外循环的学习率采用指数衰减策略，随着元训练进程逐步降低：

$$\alpha_{\text{iter}} = \alpha_0 \cdot \exp\left(-\frac{\text{iter}}{\tau}\right), \quad \beta_{\text{iter}} = \beta_0 \cdot \exp\left(-\frac{\text{iter}}{\tau}\right) \tag{20}$$

其中 τ 为衰减时间常数。学习率衰减使模型在训练初期快速探索参数空间，在后期精细化调整，提升了收敛质量和最终泛化性能。

综上所述，本节设计的 MAML-ANN 模型通过输入编码层、基学习器和元学习器的协同配合，结合多重正则化策略，形成了完整的小样本空气质量预测解决方案。第 3.3 节将在此基础上给出模型的完整训练算法流程。

3 训练算法

本节在前述模型架构设计的基础上，系统阐述 MAML-ANN 模型的完整训练算法，包括元训练阶段的双层优化流程、元测试阶段的快速适配过程，以及训练中的关键实现细节。

3.1 元训练算法流程

元训练是 MAML-ANN 模型学习的核心阶段，其目标是在多个源域城市上通过双层优化获得一组通用的初始化参数 Θ。算法 3.1 完整描述了元训练流程。

Table 3 算法 3.1：MAML-ANN 元训练算法

输入：源域城市数据集 $\{D_c\}_{c=1}^{M}$，任务批次大小 B，内循环学习率 α，：外循环学习率 β，内循环步数 K，元迭代轮数 N_{meta}， ：权重衰减系数 λ，梯度裁剪阈值 γ，早停耐心值 P
初始化：随机初始化元参数 Θ，嵌入层参数 E

for n = 1 to N_{meta} do
从 M 个城市中随机采样 B 个任务 $\{T_i\}_{i=1}^{B}$
对每个任务 T_i，按比例 ρ 划分支持集 S_i 和查询集 Q_i for each
任务 T_i in parallel do
初始化任务参数 $\theta_i \leftarrow \Theta$
for k = 1 to K do
在支持集上计算损失：$L_{S_i} = \frac{1}{\lvert S_i \rvert}\sum_{(x,y)\in S_i}(f_{\theta_i}(\mathbf{x})-y)^2$
计算梯度并更新：$\theta_i \leftarrow \theta_i - \alpha \nabla_{\theta_i} L_{S_i}$
end for
得到任务专属参数 $\theta'_i = \theta_i$
end for
在查询集上计算元损失：$L_{\text{meta}} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{\lvert Q_i \rvert}\sum_{(x,y)\in Q_i}(f_{\theta_i'}(x)-y)^2 + \lambda\sum_{l}\lVert W^{(l)}\rVert_F^2$
计算元梯度：g $\leftarrow \nabla_\Theta L_{\text{meta}}$
梯度裁剪：g $\leftarrow$ clip(g, γ)
更新元参数：$\Theta \leftarrow \Theta - \beta \cdot$Adam(g, Θ)在验证
任务集上评估元损失 L_{val}
if L_{val} 连续 P 轮未下降 then 早停，终止训练end for

输出：元参数 Θ^*，嵌入层参数 E*

算法 3.1 的核心机制可进一步分解为以下关键步骤：

步骤 1：任务构建与采样（第 1-3 行）

在每一轮元迭代中，首先从 M 个源域城市中随机采样 B 个城市构成一个任务批次。对于每个被采样的城市 c，将其时间序列数据 D_C 按 3.1.2 节所述方式划分为支持集 S_i 和查询集 Q_i。划分比例 ρ 控制着支持集的样本数量——在小样本场景下，ρ 通常取较小值（如 5% − 20%),使得每个任务仅有少量样本用于内循环适配。

任务批处理（batching）的设计意图在于：通过每轮在多个任务上并行计算内循环更新，可以更准确地估计元梯度的方向，降低单个任务噪声对元参数更新的影响。实践中，B 的取值需权衡计算效率与梯度估计精度，本文取 B = 4。

步骤 2：内循环适配（第 4-10 行）

内循环是 MAML 算法区别于传统预训练微调的核心机制。对于任务批次中的每个任务 T_i，模型从元参数 Θ 出发，在支持集 S_i 上执行 K 步梯度下降，获得任务专属参数 θ'_i。本文取 K = 1，其数学表达式为：

$$\theta_i' = \Theta - \alpha\nabla_\Theta \frac{1}{|S_i|}\sum_{(\mathbf{x},y)\in S_i}(f_\Theta(\mathbf{x})-y)^2 \tag{21}$$

需要特别指出的是，公式(21)中的梯度计算在实现时需要冻结元参数 Θ 的梯度历史——即 θ'_i 的计算应保留完整的计算图（computation graph),以便后续反向传播计算对 Θ 的元梯度。

步骤 3：元损失计算与元梯度计算（第 11-13 行）

在获得各任务的任务专属参数 θ'_i 后，模型在查询集 Q_i 上评估适配后模型的表现，计算元损失。元损失由两部分组成：查询集上的平均预测误差和 L2 权重衰减项：

$$\mathcal{L}_{\text{meta}} = \frac{1}{B}\sum_{i=1}^{B}\mathcal{L}_{\mathcal{Q}_i}(f_{\theta_i'}) + \lambda\sum_{l=1}^{4}\|\mathbf{W}^{(l)}\|_F^2 \tag{22}$$

其中 $L_{Q_i}(f_{\theta_i'}) = \frac{1}{|Q_i|}\sum_{(x,y)\in Q_i}(f_{\theta_i'}(\mathbf{x})-y)^2$。

元梯度 $\nabla_\Theta L_{\text{meta}}$ 的传播路径涉及通过内循环更新公式(21)进行反向链式求导。这一计算通过深度学习框架的自动微分机制自动完成——只要在实现时保留了内循环的计算图，框架即可自动追踪从 Θ 到 θ'_i 再到元损失的完整微分路径。

步骤 4：元参数更新（第 14-15 行）

Table 4 算法 3.2：MAML-ANN 元测试算法

输入：训练好的元参数 Θ^*，目标城市数据集 D_{target}， ：支持集样本数 N_S，内循环学习率 α，内循环步数 K
步骤 1：数据划分 从 D_{target} 中取前 N_S 个样本作为支持集 S_{test}剩余样本作为测试集 Q_{test}

步骤 2：快速适配
初始化任务参数 $\theta \leftarrow \Theta^*$
for $k = 1$ to K do
$L_S = \frac{1}{\lvert S_{\text{test}} \rvert} \sum_{(x,y) \in S_{\text{test}}} (f_\theta(x) - y)^2$ $\theta \leftarrow \theta - \alpha \nabla_\theta L_S$
end for
得到适配后参数 θ_{adapted}

步骤 3：预测与评估 对测试集 Q_{test} 进行预测：$\hat{y} = f_{\theta_{\text{adapted}}}(X_{\text{test}})$ 计算性能指标：$\text{RMSE} = \sqrt{\frac{1}{\lvert Q_{\text{test}} \rvert} \sum_t (\hat{y}_t - y_t)^2}$， $R^2 = 1 - \frac{\sum_t (\hat{y}_t - y_t)^2}{\sum_t (y_t - \bar{y})^2}$
输出：预测值 $\hat{y}$，RMSE，R^2

元梯度经过梯度裁剪后，使用 Adam 优化器更新元参数 Θ：

$$\Theta \leftarrow \Theta - \beta \cdot \text{Adam}(\text{clip}(\nabla_\Theta L_{\text{meta}}, \gamma), \Theta) \tag{23}$$

Adam 优化器的动量机制在外循环中发挥着重要作用——它累积了各轮元迭代的梯度历史，能够平滑元参数在任务空间中的更新轨迹，避免因个别任务批次分布偏移而导致的元参数震荡。

步骤 5：早停验证（第 16-17 行）

为避免元训练过程中的过拟合，算法在每个元迭代轮次结束后，在预留的验证任务集上评估元损失。若验证损失连续 $P = 20$ 轮未呈现下降趋势，则触发早停机制，终止训练并恢复验证损失最低的元参数 Θ^*。早停是防止元参数在源域任务集上过度优化而牺牲泛化性能的关键保障。

3.2 元测试算法流程

元测试阶段的目标是验证训练好的元参数 Θ^* 在目标城市上的快速适配与预测能力。与元训练不同，元测试阶段不再更新元参数本身，而是从 Θ^* 出发在目标城市的少量支持集上进行适配，然后在完整的测试集上评估性能。算法 3.2 描述了元测试流程。

元测试流程的核心优势在于其极低的样本需求——在仅有 N_S 个样本（本文实验中 $N_S \in \{5, 10, 20, 50\}$）的条件下，模型仅需 K 步梯度更新即可完成适配，整个过程可在数秒内完成。

3.3 实现细节与超参数设置

为确保模型训练的稳定性和结果的可复现性，本节详细说明训练中的关键实现细节和超参数设置。

（1）参数初始化策略

元参数 Θ 的初始值对MAML 的训练收敛至关重要。本文采用Xavier 均匀初始化（Glorot Initialization),其基本思想是保持各层输入和输出的方差一致，从而避免信号在网络传播过程中的过度放大或衰减。对于第 l 层的权重矩阵 $W^{(l)} \in \mathbb{R}^{n_{\text{in}} \times n_{\text{out}}}$，Xavier 初始化的分布为：

$$\mathbf{W}^{(l)} \sim \mathcal{U}\left(-\sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}, \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}\right) \tag{24}$$

其中 U 为均匀分布。偏置向量 $b^{(l)}$ 初始化为零向量。

嵌入层参数 $E \in \mathbb{R}^{M \times 8}$ 采用均值为 0、标准差为 0.01 的正态分布初始化，以保持嵌入向量在训练初期的适度多样性。

（2）数据加载与批处理

在元训练的每一轮迭代中，从源域城市中采样 $B = 4$ 个任务。由于各城市的时间序列长度可能不同，支持集和查询集的样本数量在任务间存在差异。为实现高效的批处理，本文采用“逐任务前向传播”的策略：对每个任务独立执行内循环的前向和反向传播，累积各任务的元梯度后再进行元参数更新。这一策略避免了因序列长度不一致而需要进行填充（padding）

的麻烦，且与 PyTorch 等框架的自动微分机制天然兼容。

（3）学习率的设置

内外循环学习率的取值直接决定了 MAML 的训练动态。本文采用以下设置：

Table 5 表 3.3：超参数设置汇总

参数	符号	取值
内循环学习率	α	0.01
外循环学习率	β	0.001
学习率衰减因子	τ	500
元迭代轮数	N_{meta}	1000
任务批次大小	B	4
内循环步数	K	1
Dropout 丢弃率	p	0.3
权重衰减系数	λ	1×10^{-4}
梯度裁剪阈值	γ	1.0
早停耐心值	P	20

（4）随机种子的固定

为保证实验的可复现性，本文在数据生成、任务采样、参数初始化等涉及随机性的环节均设置固定随机种子。在 Python/MATLAB 实现中，通过特定指令统一控制随机数生成器。

3.4 计算复杂度分析

MAML-ANN 模型的计算复杂度主要来源于内循环和外循环两个层次：

（1）内循环计算复杂度

对于单个任务，单步内循环包含一次前向传播和一次反向传播。设基学习器的参数量为 $|\theta| = 14,273$，支持集样本数为 N_S，则单任务单步内循环的计算复杂度为 $O(N_S \cdot |\theta|)$。对于 $K = 1$ 和 $B = 4$ 的典型配置，每轮元迭代的内循环计算量为 $O(4 \cdot N_S \cdot 14,273)$。

（2）外循环计算复杂度

外循环需要在前向传播的基础上额外计算对元参数 Θ 的梯度，该梯度通过内循环计算图自动传播，计算量与内循环处于同一量级。

（3）总体训练复杂度

元训练的总计算复杂度为 $O(N_{\text{meta}} \cdot B \cdot K \cdot N_S \cdot |\theta|)$。以 $N_{\text{meta}} = 1000$、$B = 4$、$K = 1$、$N_S \approx 50$、$|\theta| = 14,273$ 计算，总浮点运算次数约为 2.85×10^9 次，在 CPU 环境下可在数小时内完成训练。

综上所述，本章完整定义了 MAML-ANN 模型的训练与测试算法流程，明确了各环节的数学形式和实现细节，为第 4 章的实验验证奠定了方法论基础。

3.5 未来工作展望

基于上述讨论，未来可从以下方向推进研究：

（1）多源数据融合

将卫星遥感 AOD（Aerosol Optical Depth）数据、气象再分析资料（如 ERA5）、交通流量数据等多源异构信息纳入 MAML 框架，丰富特征空间，进一步提升小样本下的预测精度[5]。

（2）图神经网络扩展

将 MAML 与图神经网络（GNN）结合，利用城市间的地理邻接关系和气象连通性构建空间图结构，实现更充分的空间信息共享。图结构可作为任务间关系的先验知识，引导元学习的任务采样和梯度聚合。

（3）在线元学习

研究持续学习（Continual Learning）场景下的 MAML 扩展，使模型能够在连续到达的新城市数据上增量更新元参数，避免周期性重新训练的计算开销，提升模型的持续适应能力。

（4）物理-informed 神经网络

将大气物理化学过程的先验知识（如扩散方程、化学反应动力学）以偏微分方程约束的形式嵌入神经网络，构建物理-informed MAML（PI-MAML),在极端小样本条件下通过物理先验弥补数据不足。

（5）可解释性增强

引入注意力机制定位关键时间步和关键特征，结合 SHAP 值分析，为模型预测提供可解释性依据，增强在环境管理应用中的可信度。

参考文献：

[1] YANG Y, CERMAK J, CHEN X, et al. High-resolution pm10 estimation using satellite data and model-agnostic meta-learning[J/OL]. Remote Sensing, 2024, 16(13): 2498. DOI: 10.3390/rs16132498.

[2] HE R R, CHEN Y Q, TIAN L, et al. Constructing and evaluating predictors for data-driven pm2.5 forecasting models [J/OL]. International Journal of Environmental Research, 2025, 19: 99. DOI: 10.1007/s41742-025-00767-x.

[3] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[C]//Proceedings of

the 34th International Conference on Machine Learning: Vol. 70. JMLR.org, 2017: 1126-1135.
[4] NICHOL A, ACHIAM J, SCHULMAN J. On first-order meta-learning algorithms[A]. 2018.
[5]DEY P, et al. Fewshot-airnet: Temporal 2d-variation modeling for air pollution forecasting[C]//2025 IEEE International Conference on Data Mining (ICDM). Brisbane, Australia, 2025.

作者简介：李佳璇（2002—），女，汉族，辽宁省锦州市，硕士研究生，辽宁师范大学，研究方向为 CAD。