arXiv:2602.03026v1 [cs.AI] 3 Feb 2026

基于多智能体系统的时间序列视觉推理

Weilin Ruan[1] Yuxuan Liang*, [1]中国香港科技大学（广州），
中国 rwlinno@gmail.com, yuxliang@outlook.com

摘要

时间序列分析是许多现实应用的基础，但现有的时间序列专用方法和基于预训练大模型的方法，在融合直观的视觉推理、通过自适应工具使用实现跨任务泛化方面仍存在局限。为解决这些问题，我们提出了**MAS4TS**，这是一种面向通用时间序列任务的工具驱动型多智能体系统。该系统基于**分析器–推理器–执行器**范式构建，在统一框架中整合了智能体通信、视觉推理与潜在空间重建。MAS4TS首先利用视觉语言模型，结合结构化先验，对时间序列图进行视觉推理，以提取时间结构，随后在潜在空间中重建预测轨迹。三个专用智能体通过共享记忆和门控通信进行协作，同时由路由器选择特定任务工具链执行。多个基准测试上的大量实验表明，MAS4TS在广泛的时间序列任务中取得了最先进的性能，同时表现出强大的泛化能力和高效的推理能力。

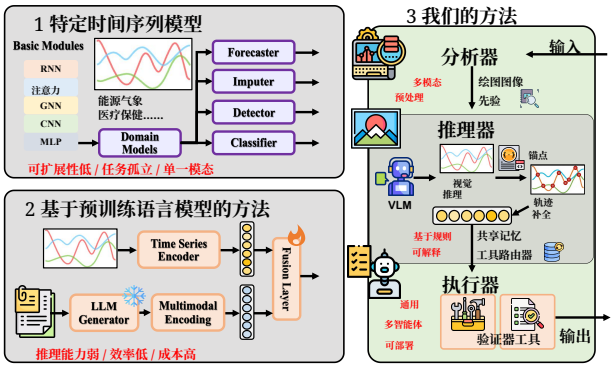

图 *1*。 MAS4TS 的动机。现有方法（左）难以支持直观推理和任务自适应执行，而 MAS4TS（右）将时间序列分析组织成一个统一范式，以协作方式进行分析、推理和执行。

1 引言

时间序列分析是众多现实世界应用的基础，包括能源系统（Deb 等，2017）、交通运输（Zheng 和 Huang，2020）以及其他关键领域（Idrees 等，2019；Karevan 和 Suykens，2020；Schneider 和 Dickinson，1974）。预测、分类、插补和异常检测等核心任务（Lim 和Zohren，2021）在支持这些场景中的数据驱动决策方面发挥着至关重要的作用。

近期，预训练大语言模型（LLMs）取得的进展展现了其在模式理解和一般推理方面的强大能力(Brownet al., 2020; Touvron 等人，2023）。受此进展启发，越来越多的研究开始探索如何利用预训练模型进行时间序列分析（Jin 等人，2024a；Liu 等人，2024a；Zhou 等人，2023），旨在提升泛化能力和上下文建模能力，超越传统神经网络架构（Wu等人，2023b；Zhou 等人，2021；Nie 等人，2023a）。并行地，智能体方法和多智能体系统（MAS）（Ravuru 等人，2024；Gu 等人，2025；Chang 等人，2025a；Wu 等人，2024；Hong 等人，2023；Qian 等人，2023；Huang 等人，2025）正逐渐成为一种实用范式，可用于工具编排、任务分解和协同执行。

尽管取得了这些进展，但多智能体范式在通用时间序列分析中的应用仍鲜有探索。现有方法大多依赖数值或嵌入空间中的隐式历史模式匹配，缺乏能够捕捉潜在时间结构的显式、可解释推理机制。即便是近期的视觉增强时间序列方法（Chen 等人，2024；Zhong 等人，2025；Ruan 等人，2025）也常将视觉表征视为辅助特征，而没有充分发挥现代视觉语言模型（VLM）在时间序列图上进行直接推理的能力，而这些图能够自然地编码趋势、状态转变和结构转变。此外，当前方法通常针对固定推理流程下的孤立任务设计，对自适应工具编排以及在异构时间序列工作负载中的协同执行支持有限。这些局限可归纳为两个概念性缺口：

能力。这些局限可归纳为两个概念性缺口：

• **从表示学习到结构推理**。现有方法在数值空间或潜在嵌入空间中运行，形态模式在其中仍然是隐式且相互纠缠的。近期的视觉增强方法（例如 TimeVLM、VisionTS）将时间序列转换为视觉形式，主要用于表示学习，却未能利用视觉推理能力，从图表中提取显式的结构知识，而人类分析师会自然地借助这类图表来理解时序。我们能否从学习表示转向对时间序列可视化进行直观的结构推理？

• **从任务特定模型转向工具驱动执行**。当前范式将预测、分类、插补和异常检测视为彼此独立的问题，即使它们共享同一个骨干网络，通常仍需要任务特定的头部和各自独立的流程。与利用工具编排进行自适应问题求解的多智能体系统不同，时间序列方法仍遵循固定模型后进行单任务微调的范式。这使预训练模型无法充当通用且能适应任务的时间序列求解器，并导致成本随规模呈倍数增长。我们能否通过共享结构推理统一多种时间序列任务，同时支持针对特定任务的自适应工具选择？

通用时间序列任务从根本上要求理解时间序列图中显式呈现、而在数值嵌入中隐含的形态语义。这表明，时间序列智能的瓶颈在于结构推理，而非表示学习，从而推动范式转变：通过视觉推理统一各类任务，同时实现自适应的工具驱动执行。

基于这一洞见，我们提出了**MAS4TS**，首个面向通用时间序列分析、基于**分析器–推理器–执行器** 范式构建的工具驱动多智能体系统。MAS4TS 将时间序列分析分解为相互补充的角色：分析器 执行统计分析和数据预处理；推理器 融合视觉推理与数值推理，以提取高层次的时序先验并推导潜在轨迹约束；执行器 调用适当的工具和模型，生成特定任务的输出。具体而言，MAS4TS 使用视觉语言模型直接对时间序列图执行视觉推理，以获得结构化锚点，随后利用这些锚点引导潜在空间中的轨迹重构。通过共享记忆和智能体间通信，专用智能体在统一框架内协作，以解决多样化的时间序列任务。我们的主要贡献总结如下：

• 我们提出了**MAS4TS**，这是首个面向通用时间序列分析的工具驱动多智能体系统，确立了一种分析器–推理器–执行器 范式，用于任务自适应的时间序列建模。

• 我们提出了一种直观的推理方法，利用视觉语言模型从图表中提取视觉锚点和先验知识，然后将其转换为用于潜在轨迹重构的数值约束。

• 我们开展了广泛的实验，结果表明，在预测、分类、插补和异常检测任务中，MAS4TS均表现出优越的性能。

2 相关工作

2.1 时间序列分析

经典时间序列模型（Box 等人，2015年；Cleveland 等人，1990年）在处理简单或线性模式时仍然有效。随着越来越专业化的架构不断发展，深度学习已成为现代时间序列分析的主流范式。循环模型（Medsker 等人，2001年；Hochreiter 和 Schmidhuber，1997b；Cho 等人，2014年）能够捕捉短期时间依赖性，但在长序列上容易受到梯度衰减的影响；卷积模型（Bai 等人，2018；Van Den Oord 等人，2016年；Wu等人，2023b）则能够高效提取局部与多尺度模式，并展现出较强的计算优势。基于Transformer的模型（Zhou 等人；2021；Wu 等人，2021；Liu 等人，2022b；Zhou 等人，2022；Nie 等人，2023a）凭借灵活的感受野和强大的建模能力，已成为主流，其创新包括自相关注意力、频域分解和补丁级表示。在大型预测基准测试中，线性架构（Zeng 等人，2023；Xu 等，2023；Zhou 等人，2025）展现出出人意料的竞争力，重新引发了人们对归纳偏置和架构简洁性的关注。

2.2 面向时间序列的预训练大模型

受大语言模型成功的启发，近期研究开始探索时间序列基础模型（Liang 等人，2024；Jin 等人，2024b）。基于大语言模型的方法（Jin 等人，2024a；Liu 等人，2024a；Zhou 等人，2023；Chang 等人，2025b）将时间序列与文本嵌入对齐，利用预训练语言模型实现跨域泛化。然而，这些方法面临固有的模态差距和计算开销。视觉增强方法将时间序列转换为图像，并使用预训练视觉模型（Chen 等人，2024；Zhong 等人，2025；Shen 等人，2025；Ruan 等人，2025；Lyu 等人，2025），但主要将视觉表示视为表示学习的辅助特征。我们的工作有所不同：我们利用视觉语言模型对时间序列图进行直接视觉推理，提取明确的结构模式，以指导下游任务的执行。

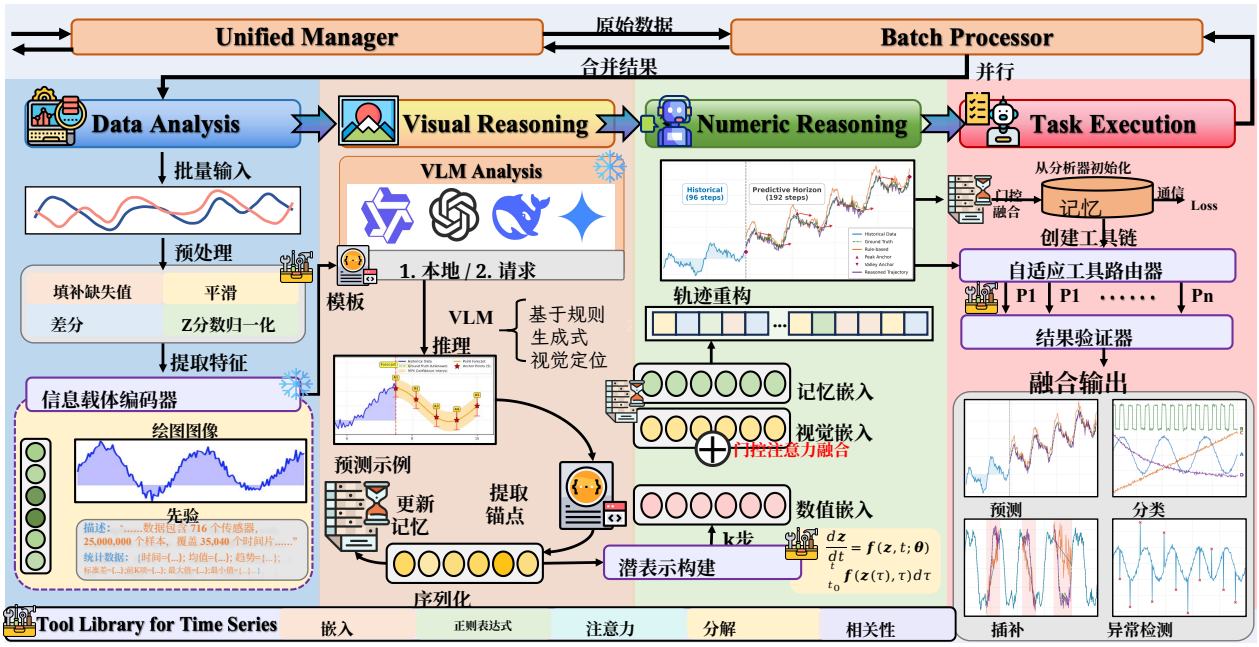

图2. MAS4TS框架概览，遵循分析器-推理器-执行器范式。推理智能体依次执行两个阶段：视觉推理（基于视觉语言模型的锚点提取）和数值推理（潜在轨迹重建）。统一的工具库支撑整个工作流，并主要集成于执行器。

2.3 多智能体系统

多智能体系统（MAS）在复杂推理、规划和工具编排方面展现出强大的能力（Wu 等人，2024；Hong 等人，2023；Qian 等人，2023）。通过为智能体分配专门角色并实现结构化通信，多智能体系统能够分解并处理超出单个模型能力范围的任务。工具增强方法（Schick 等人，2023；Qin 等人，2023）则通过外部函数调用进一步扩展智能体的能力。近期研究探索了针对特定时间序列任务的智能体方法，例如预测（Huang 等人，2025）和异常检测（Gu 等人，2025；Ravuru 等人，2024），但这些方法针对的是孤立的问题，而非提供统一的分析框架。我们的工作填补了这一空白，提出一种工具驱动的多智能体框架，将视觉推理融入通用时间序列分析。

3 方法

如图 2所示，MAS4TS 采用规划—执行式多智能体架构（Huang 等人，2024），其中管理器会将时间序列任务动态拆解为由核心智能体执行的专门阶段。随后，处理器执行任务感知的自适应规划，并支持智能体并行执行，以提高推理效率。对于每个并行批次，工作流由三个智能体组成：❶ 分析器，用于提取统计特征 $\mathbf{F}_{\text{stat}}$ 以及任务感知的绘图图像表征；❷ 推理器，它利用视觉语言模型生成锚点 $\mathcal{A}$（视觉推理阶段），并在潜在空间中重建序列（数值推理阶段）；以及 ❸ 执行器，它通过自适应工具使用生成 $\hat{}$ 并验证最终预测 $\mathbf{Y}$。

3.1 问题形式化

令$\mathbf{X} = \{x_1, \ldots, x_L\} \in \mathbb{R}^{L \times D}$表示输入的多变量时间序列，其中 L为观测序列的长度，D 为变量（特征）的数量。我们考虑四类通用任务：❶预测，即在长度为 H的预测视野上预测未来值$\mathbf{Y} = \{x_{L+1}, \ldots, x_{L+H}\} \in \mathbb{R}^{H \times D}$；❷分类，即为整个序列分配类别标签 $y \in \{1, \ldots, C\}$，其中 C为类别数；❸插补，即估计缺失项 $\{\hat{x}_{i,j} : (i,j) \in \Omega\}$，其中Ω表示缺失时间-变量对的索引集；以及❹异常检测，即在观测窗口内识别异常时间戳$\mathbf{s} \subset \{1, \ldots, L\}$。MAS4TS提供统一的智能体驱动框架，并在这些任务之间共享通用的感知、推理和记忆组件。

3.2 多智能体协作框架

时间序列分析需要异质推理，而这难以在单一模型中统一，因此我们设计了专用智能体，让它们通过共享记忆协作，在保留任务特定表示的同时实现选择性知识迁移。

共享内存机制。我们定义了一组用于批量数据处理的智能体 $\{\text{Agent}^{(i)}\}_{i=1}^{N}$，其中在我们的实例化中，$N$=3 对应于分析器、推理器和执行器。每个智能体都具有：(i) 一个嵌入函数 $\text{Embed}^{(i)}(\cdot)$，(ii) 一个局部内存视图 $\mathcal{M}^{(i)}$，以及 (iii) 一个通信模块 $\text{Com}(\cdot)$。跨智能体协调通过共享内存矩阵$\mathbf{M} \in \mathbb{R}^{L\times d_m}$实现，该矩阵通过门控融合整合各智能体的知识。给定输入时间序列$\mathbf{X}$ 和当前共享记忆$\mathbf{M}$，每个智能体计算：

$$\mathbf{h}^{(i)}, \alpha_i = \text{Agent}^{(i)}\big(\text{Embed}^{(i)}(\mathbf{X}), \mathcal{M}^{(i)}\big), \quad (1)$$

$$\text{Com}(\mathbf{h}^{(i)}|\mathcal{M}^{(i)}) = \mathbf{h}^{(i)} + \text{LN}\Big(\text{Enc}_\ell(\mathbf{h}^{(i)}, \mathcal{M}^{(i)})\Big), \quad (2)$$

$$\mathbf{M} \leftarrow \mathbf{M} \oplus \alpha_i \cdot \text{Com}(\mathbf{h}^{(i)}|\mathcal{M}^{(i)}), \quad (3)$$

其中$\mathbf{h}^{(i)}$ 表示智能体特定的隐藏状态，$\alpha_i \in[0,1]$ 是控制更新幅度的置信度分数，$\text{LN}(\cdot)$表示针对以智能体视角为条件的$\ell = 1,\ldots, L_{enc}$编码器层的层归一化。门控机制 $\oplus$能够实现迭代式内存细化，同时降低噪声智能体输出的影响。

信息载体编码器。智能体通过结构化信息载体 $\{\mathcal{I}, \mathcal{E}, \mathcal{P}\}$ 进行通信，这些载体首先由**分析器**生成，具体如下：

- $\mathcal{I}$：通过基于规则的算子 $\mathcal{I} = \mathcal{R}(\{(t,x_{t,d})\}_{t=1}^{L})$直接从原始时间序列渲染的绘图图像；
- $\mathcal{E}$：来自工具集成的时间序列预处理与编码的归一化嵌入；
- $\mathcal{P}$：结构化先验，包括统计特征、语义描述，以及来自通信的共享记忆。

这些载体构成了紧凑而富有表现力的接口，用于智能体间通信和下游推理。

3.3 潜在视觉推理

推理器旨在弥合直观的视觉模式识别与时间序列分析所需的精确数值推断之间的差距。我们不直接拟合历史轨迹，而是利用视觉推理识别稀疏但信息丰富的结构线索，随后通过连续时间潜在动力学对其加以约束。

基于视觉语言模型的锚定。时间序列中的许多显著时间结构，包括趋势、状态转变和边界行为，往往更容易从视觉表示中识别，而不是从原始一维序列中识别。视觉语言模型（VLM）可直接对绘图图像进行高层模式识别。形式上，给定绘图图像 $\mathcal{I}$ 和结构化先验 $\mathcal{P}$，我们向 VLM M 如下：

$$\mathcal{A} = \text{VLM}\Big(\text{Enc}_{\text{vis}}(\mathcal{I}),\ \text{Enc}_{\text{text}}(\text{Prompt}(\mathcal{P}))\Big) \quad (4)$$

$$= \{(t_k, v_k, \tau_k)\}_{k=1}^{K}, \quad (5)$$

其中，提示（见附录D）指定了所需的输出模式。每个元组由一个关键时间索引 $t_k \in \{1,\ldots, L\}$、一个估计值 $v_k \in \mathbb{R}$以及一个指示符 τ_k（默认值：$\{-1, 0, 1\}$）组成，用于表示局部时间行为（例如，下降／稳定／上升）。这些稀疏锚点编码了通过视觉推理提取的高层结构知识，作为引导潜在空间重建的语义约束。

潜表示构建。视觉锚点虽然稀疏且离散，但下游时间序列任务需要稠密且时间连贯的表示。为弥合这一差距，我们引入一个潜表示构建模块，将锚点视为引导潜在空间演化的条件信号，从而重建连续潜在轨迹。形式上，我们定义连续时间潜在状态$\mathbf{z}(t)$，并将其动力学参数化为：

$$\frac{d\mathbf{z}(t)}{dt} = f_\phi\big(\mathbf{z}(t), t \mid \mathcal{A}, \mathcal{P}\big), \quad \mathbf{z}(t_0) = \mathbf{z}_0, \quad (6)$$

其中，f_ϕ 是一个以锚点集合 $\mathcal{A}$ 和先验 $\mathcal{P}$为条件的可学习动力学函数，初始状态$\mathbf{z}_0$ 则由嵌入 $\mathcal{E}$获得。我们采用有限步潜在重构算子 $\mathcal{S}(\cdot)$来获取潜在状态$\{\mathbf{z}_t\}_{t=1}^{L+H}$，覆盖观测窗口和预测范围：

$$\mathbf{u}^{(k)} = \mathcal{S}^{(k)}\big(\cdots \mathcal{S}^{(1)}(\mathbf{u}^{(0)}, \mathcal{A}, \mathcal{P})\big), \quad \mathbf{z}_t = [\mathbf{u}^{(k)}]_t, \quad (7)$$

其中，$\mathbf{u}^{(i)} \in \mathbb{R}^{(L+H)\times D}$ 表示第 i个求解器步骤之后的序列级潜在表示，$\mathbf{u}^{(0)} = \text{Proj}(\mathcal{E})$以数据嵌入初始化，并通过通信进行融合，而$\mathbf{z}_t$ 是$\mathbf{u}^{(k)}$的第 t行。

在实践中，我们使用 $\mathcal{S}(\cdot)$ 的有限步（$k \le 20$）显式潜在常微分方程求解器，因为锚点条件会大幅限制可行解空间，并降低潜在轨迹的有效曲率。直观地说，这与连续潜在建模领域的先前工作（Dormand 和 Prince，1980；Rubanova 等人，2019）一致，锚点条件下的潜在动力学充当一种软边界正则化，使重建轨迹偏向语义上有意义的时间结构，同时保留局部变化的灵活性。

通过数据嵌入进行潜在融合。重建的潜在轨迹提供高层结构指导，但并不取代原始数据表示。相反，它与数据嵌入进行融合，通过注意力融合形成统一的潜在表示：

$$\mathbf{Z} = \text{Attn}(\mathcal{E} \ \oplus\ \text{Proj}\Big(\mathbf{u}^{(k)}\Big)), \quad (8)$$

其中，$\oplus$ 表示门控融合算子。由此得到的潜在表示 $\mathbf{Z}$ 具有稠密、可微且语义扎根的特性，可在统一潜在框架内充当免训练视觉推理与数据驱动数值推断之间的桥梁。

3.4 工具集成执行与优化

执行器通过工具集成执行与结果验证，将潜在表示转换为特定任务的输出，同时仅优化协调关键组件，并保持推理模块无需训练。MAS4TS 不依赖单体预测器，而是动态组合工具链，应用专用时间序列算子来细化中间结果。

时间序列工具库。在执行时，自适应路由器从预定义的工具库 $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_M\}$ 中选择工具链，其中每个工具实现一种专门操作（例如趋势外推、分解、平滑、统计预测、约束投影和残差校正）。工具路由会根据任务类型 κ、结构化先验 $\mathcal{P}$ 以及前序阶段维护的共享记忆 $\mathbf{M}$进行调整。形式上，执行器生成针对工具的路由策略 π_Θ，并据此构建工具序列 $\mathbf{m} = (m_1, \ldots, m_J)$。最终输出通过顺序复合获得：

$$\pi_\Theta(m \mid \kappa, \mathcal{P}, \mathbf{M}) = \mathrm{Softmax}(g_\psi(\kappa, \mathcal{P}, \mathbf{M})), \quad (9)$$

$$\hat{\mathbf{Y}} = \mathcal{T}_{m_J} \circ \cdots \circ \mathcal{T}_{m_1}(\mathbf{Z}), \quad (10)$$

其中，g_ψ 是一个门控控制器网络，长度为 J 的工具序列由策略π_Θ 动态确定，并通过 $\circ$ 操作连接。这种设计无需针对特定任务重新训练单体骨干，即可通过即插即用的专家算子增强原始预测。

结果验证器。 为确保数值有效性和任务一致性，MAS4TS采用轻量级验证器执行以下操作：(i) 格式与形状校正，(ii) 约束校验，以及(iii) 任务特定归一化。如果候选输出未通过验证，执行器会重新计算，或回退到保守的基础工具，以获得有效预测。给定候选输出$\mathbf{Y}$，验证器将生成最终预测：

$$\mathbf{Y}^* = \mathrm{Verifier}(\hat{\mathbf{Y}} \mid \mathcal{A}, \mathcal{P}), \quad (11)$$

其中，视觉锚点 $\mathcal{A}$ 和结构化先验 $\mathcal{P}$ 作为外部一致性检查。对于连续输出，违反锚点所隐含边界的情况会通过约束感知投影进行软校正；对于离散或分类任务，验证器会确保模式一致性，并剔除逻辑上无效的预测。这使高层推理与严格的约束满足相分离，在不牺牲灵活性的同时提升鲁棒性。

优化策略。 MAS4TS 支持端到端可微执行，同时仅选择性地训练协调关键组件。静态预处理、视觉语言模型提示与推理，以及 $\mathcal{T}$ 中选定的工具均保持无需训练，作为固定的推理与执行基元。优化侧重于控制信息流与协调的可学习组件，包括：(i) 嵌入与通信函数，(ii) 共享记忆表示，以及 (iii) 门控与路由参数 Θ。我们优化一个任务相关目标：

$$\mathcal{L} = \mathcal{L}_{\text{task}}(\mathbf{Y}^*, \mathbf{Y}) + \lambda \cdot \mathcal{L}_{\text{com}}(\mathbf{M}), \quad (12)$$

其中，$\mathcal{L}_{\text{task}}$ 对应下游目标（例如，预测任务中的$\mathcal{L}_{\text{MSE}}$），而 $\mathcal{L}_{\text{com}}$ 则对记忆更新进行正则化，以促进稳定的智能体间通信。

MAS4TS将多智能体协作、直观的视觉推理和工具增强执行整合到统一框架中。通过明确区分结构推理、连续潜在重建和工具集成执行，MAS4TS在保持数值一致性和训练效率的同时，能够灵活适应多种时间序列任务。

4 实验

我们进行了大量实验，以评估 MAS4TS 在四项基础时间序列任务上的表现：预测、分类、插补和异常检测。我们在 21 个广泛使用的数据集上进行基准测试，并与 20 多种具有竞争力的基线方法进行比较，其中包括特定任务的时间序列模型和近期的基于语言模型的方法。为确保比较公平，我们遵循各项任务的标准协议（数据划分、输入/输出长度和评估指标），并在不同方法间使用相同的数据划分和评估设置。实验细节见附录A。

4.1 预测

设置。 对于长期预测，我们在 7 个广泛采用的基准测试上进行评估：ETT（4 个子集：ETTh1、ETTh2、ETTm1、ETTm2）、天气、太阳能和汇率。遵循既定协议（Wu 等人，2023b；Liu 等人，2024b），我们采用固定的 96 长度回看窗口，并对预测范围进行评估 $H \in \{96, 192, 336, 720\}$。

结果。 如表 1所示，MAS4TS 在所有基准测试中均取得了具有竞争力的表现。在 Weather 数据集上，MAS4TS 的平均均方误差为 0.232，优于 MAFS（0.252）和 TimeVLM（0.262），分别提升 7.9% 和 11.5%。汇率数据集中的时间依赖关系呈现出伴随状态转变的非平稳模式，而 MAS4TS 的平均均方误差达到 0.343，这表明视觉锚定能够有效捕捉突发的趋势变化，而这些变化对纯数值方法构成了挑战。

表1. 长期预测任务的汇总结果。我们报告了平均值，即四种预测长度下的平均指标，{96, 192, 336, 720} 针对每个数据集。完整结果见表9。**红色粗体**：最佳结果；蓝色：次佳结果。

模型	MAS4TS (我们的方法)		MAFS 2025		TimeVLM 2025		UniTime 2024a		iTransformer 2024b		PatchTST 2023b		Crossformer 2023		TiDE 2023		TimesNet 2023a		DLinear 2023		SCINet 2022a		FEDformer 2022		Stationary 2022c		Autoformer 2021	
指标	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差
天气	**0.232**	**0.270**	0.252	0.275	0.262	0.281	0.253	0.276	0.258	0.278	0.265	0.285	0.264	0.320	0.271	0.320	0.259	0.287	0.265	0.315	0.292	0.363	0.309	0.360	0.288	0.314	0.338	0.382
太阳能	**0.212**	0.276	0.234	0.265	0.259	0.293	0.270	0.315	0.233	**0.262**	0.287	0.333	0.406	0.442	0.347	0.417	0.403	0.374	0.330	0.401	0.282	0.375	0.328	0.383	0.350	0.390	0.586	0.557
汇率	**0.343**	**0.397**	0.396	0.417	0.355	0.399	0.465	0.462	0.360	0.403	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414	0.750	0.626	0.519	0.429	0.461	0.454	0.613	0.539
ETTh1	**0.440**	**0.438**	0.450	0.440	0.457	0.443	0.442	0.448	0.454	0.447	0.516	0.484	0.529	0.522	0.541	0.507	0.458	0.450	0.461	0.457	0.747	0.647	0.498	0.484	0.570	0.537	0.496	0.487
ETTh2	**0.379**	**0.402**	0.380	0.403	0.384	0.405	0.379	0.403	0.383	0.407	0.391	0.411	0.942	0.684	0.611	0.550	0.414	0.427	0.563	0.519	0.954	0.723	0.437	0.449	0.526	0.516	0.450	0.459
ETTm1	**0.367**	**0.381**	0.397	0.402	0.388	0.398	0.385	0.399	0.407	0.410	0.406	0.407	0.513	0.495	0.419	0.419	0.400	0.406	0.404	0.408	0.485	0.481	0.448	0.452	0.481	0.456	0.588	0.517
ETTm2	**0.278**	**0.326**	0.288	0.330	0.279	0.329	0.293	0.334	0.288	0.332	0.290	0.334	0.757	0.610	0.358	0.404	0.291	0.333	0.354	0.402	0.954	0.723	0.305	0.349	0.306	0.347	0.327	0.371

表2. 8个UEA多变量时间序列数据集上的平均分类准确率（%）。**红色加粗**：最佳结果；蓝色：次佳结果。所有数据集的完整结果见表10。

模型	DTW	XGBoost	长短期记忆网络	LSTNet	LSSL	TCN	Trans.	Re.	In.	Pyra.	Auto.	FED.	ETS.	iTransformer.	DLinear	LightTS	TiDE	**MAS4TS**
	(1994年)	(2016年)	(1997a)	(2018a)	(2022)	(2019)	(2017)	(2020)	(2021)	(2022b)	(2021)	(2022)	(2022)	(2024b)	(2023)	(2023)	(2023)	(我们的方法)
平均准确率	62.78	61.48	51.72	66.40	65.30	67.26	67.31	66.93	59.93	65.58	66.00	65.32	65.83	65.16	64.85	64.47	64.16	**68.25**

ETT基准进一步验证了我们框架的稳健性：MAS4TS 在全部四个子集中始终名列前茅，在 ETTh2 上的表现尤为出色（均方误差为 0.379）；其中复杂的多尺度模式既需要细粒度的数值推理，也需要高层次的结构理解。

4.2 分类

实验设置。 我们在UEA档案库中的8个多变量数据集上进行评估（Bagnall等人，2018）：乙醇浓度（化学）、人脸检测（视频）、手写（动作捕捉）、心跳（医学）、日语元音（音频）、Sel-fRegulationSCP1/SCP2（脑电图）以及UWave手势库（手势识别）。这些数据集涵盖多个不同领域，序列长度（从29到1,751）、维度（从3到144）和类别数各不相同。

结果。 表 2给出了分类结果。MAS4TS的平均准确率最高，达到68.25%，分别比iTransformer（65.16%）和DLinear（64.85%）高出4.7%和5.2%。在FaceDetection数据集上，由于该任务需要捕捉视频序列中细微的时间动态，MAS4TS的准确率达到69.38%，而TiDE为65.3%，这表明视觉语言模型提取的视觉特征具有较强的判别能力。MAS4TS在特性各异的数据集上均保持稳定表现，而Informer等专用方法的性能波动则较大。

4.3 插补

实验设置。 我们在随机缺失场景下，针对 ETT 系列、电力、天气和交易所数据集评估插补效果，掩码比例为 $r \in \{12.5\%, 25\%\ 37.5\%, 50\%\}$。按照（Wu 等人，2023b）中的方案，在所有时间步和变量中随机抽取缺失值。

结果。如表3所示，MAS4TS 在所有掩码比例下均实现了最先进的插补性能。在 ETTm2 上，缺失比例为 50% 时，MAS4TS 的均方误差为 0.030，相较于排名第二的 LightTS（0.046）降低了 34.8%。随着掩码比例提高，性能提升幅度也随之增大，展现出其在严重数据损坏情况下的强大鲁棒性。在交易所数据集上，MAS4TS 的平均均方误差为 0.005，优于所有基线方法。这验证了我们的锚点约束补全方法在缺失片段较长时尤为有效：视觉锚点提供了可靠的结构路径点，而潜表示构建则将这些锚点视为边界约束，从而生成平滑且时间连贯的轨迹。尽管在缺失比例较低时性能提升较为有限，但多智能体协同机制确保了稳定的改进，而基础模型的泛化能力也为进一步处理特定领域的缺失模式带来了潜力。

4.4 异常检测

设置。 我们在五个服务器和基础设施监控基准测试上进行评估：SMD（Su 等人，2019）（包含 38 个维度的服务器机器数据集）、MSL 和 SMAP（Hund- man 等人，2018）（来自美国国家航空航天局的航天器遥测数据）、SWaT（Mathur 和 Tippenhauer，2016年）（拥有 51 个传感器的水处理厂），以及 PSM（Abdulaal 等人，2021）（来自 eBay 的服务器指标）。这些数据集包含点异常和上下文异常，且污染率各不相同。

结果。 表4 展示了异常检测结果。

表 3. 插补任务的结果。我们随机屏蔽 12.5%、25%、37.5% 和 50% 的时间点，并报告平均性能。完整结果见表 11。最佳和次佳结果分别以 **红色** 和 蓝色标示。

模型	MAS4TS (我们的方法)		ETSformer (2022)		LightTS (2022)		DLinear (2023)		FEDformer (2022)		Pyraformer (2021a)		Informer (2021)		Reformer (2020)		LSTM (1997年)		TCN (2019)		LSSL (2022)	
指标	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差
ETTm1	0.054	**0.147**	0.120	0.253	0.104	0.218	0.093	0.206	0.062	0.177	0.717	0.570	0.071	0.188	**0.050**	0.154	0.989	0.786	0.516	0.497	0.113	0.254
ETTm2	**0.030**	**0.103**	0.208	0.327	0.046	0.151	0.096	0.208	0.101	0.215	0.465	0.508	0.156	0.292	0.157	0.280	1.027	0.800	0.266	0.407	0.175	0.324
ETTh1	0.129	**0.235**	0.202	0.329	0.284	0.373	0.201	0.306	**0.117**	0.246	0.842	0.682	0.161	0.279	0.122	0.245	1.225	0.873	0.621	0.571	0.424	0.481
ETTh2	**0.067**	**0.166**	0.367	0.436	0.119	0.250	0.142	0.259	0.163	0.279	1.079	0.792	0.337	0.452	0.234	0.352	2.039	1.114	0.431	0.503	0.495	0.475
电力	**0.086**	**0.202**	0.214	0.339	0.131	0.262	0.132	0.260	0.130	0.259	0.297	0.382	0.222	0.328	0.200	0.313	0.277	0.365	0.582	0.597	0.222	0.293
天气	**0.036**	**0.070**	0.076	0.171	0.055	0.117	0.052	0.110	0.099	0.203	0.152	0.235	0.045	0.104	0.038	0.087	0.365	0.434	0.183	0.291	0.045	0.108
汇率	**0.005**	**0.041**	0.128	0.262	0.042	0.142	0.046	0.146	0.241	0.324	0.398	0.532	0.439	0.554	0.302	0.439	0.032	0.123	0.043	0.146	0.040	0.139
第一名次数	5	6	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0

表 4。 以 F1分数（%）衡量的异常检测结果。F1分数越高，性能越好。完整结果见表 12。 **红色粗体**：最佳结果；蓝色：次佳结果。

数据集	Transformer 2017	Reformer 2020	Informer 2021	Autoformer 2021	Pyraformer 2022b	异常检测 T. 2021	Stationary 2022c	ETSformer 2022	DLinear 2023	TiDE 2023	iTransformer 2024b	PatchTST 2023b	**MAS4TS** (我们的方法)
SMD	79.56	75.32	81.65	85.11	83.04	85.49	84.62	83.13	77.10	68.91	71.15	84.62	85.10
MSL	78.68	84.40	84.06	79.05	84.86	83.31	77.50	85.03	84.88	70.18	72.54	78.70	82.06
SMAP	69.70	70.40	69.92	71.12	71.09	71.18	71.09	69.50	69.26	64.00	66.87	68.82	68.58
SWaT	80.37	82.80	81.43	92.74	91.78	83.10	79.88	84.91	87.52	76.73	79.18	85.72	93.14
PSM	76.07	73.61	77.10	93.29	82.08	79.40	97.29	91.76	93.55	92.50	95.17	96.08	95.04
平均 F1 分数	76.88	77.31	78.83	84.26	82.57	80.50	82.08	82.87	82.46	74.46	76.98	82.79	**84.78**

MAS4TS取得了最高的平均F1分数，为84.78%，较ETSformer（82.87%）和iTransformer（76.98%）分别提升了1.5%和9.3%。在SWaT上，MAS4TS的F1分数达到93.14%，且精确率（91.61%）与召回率（94.74%）表现均衡，表明其能够有效区分正常波动与真实异常。具备视觉推理能力的**推理器** 对异常检测尤为有价值：通过直接从时间序列可视化中“观察”异常模式，它能在模型完全收敛之前识别显著偏差，并预先定位可疑区域，为后续数值分析提供指引。视觉锚点编码了从正常模式中学习到的预期时间结构，使模型能够在观测值偏离这些结构先验时进行灵敏检测。此外，**执行器**的验证步骤通过锚点一致性检查过滤假阳性，从而在不牺牲召回率的情况下提高精确率。

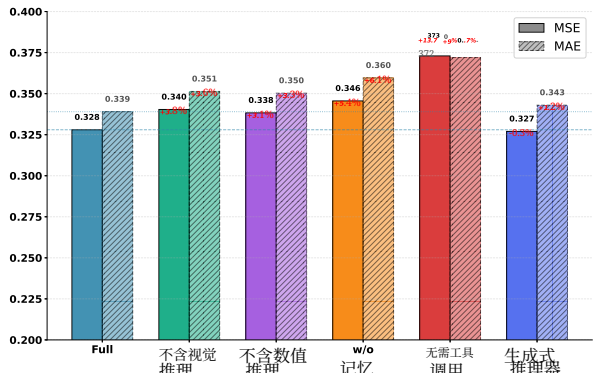

图 3。 预测跨度为 $H = 720$ 时，Weather 数据集上的消融实验结果。每个组件都有助于提升性能，而生成式推理器并未带来进一步改进。

4.5 模型分析

消融研究。 我们开展了系统的消融研究，以量化 MAS4TS 中各关键组件的贡献。所有变体均在 Weather 数据集上进行评估，预测步长为 $H = 720$。图 3展示了完整的 MAS4TS 以及以下变体的结果：(a) 移除视觉推理；(b) 移除数值推理；(c) 移除共享记忆；(d) 移除工具，并在 **执行器**中使用简单预测器；以及 (e)生成式推理器，它用生成模型（*Nano-Banana*）替代默认的基于规则的渲染流程来生成可视化，随后采用相同的锚点提取流程。

消融实验结果揭示了若干重要发现。移除工具调用导致性能下降最为显著，均方误差增加 13.7%（0.328 → 0.373），平均绝对误差增加 9.7%（0.339 → 0.372），这凸显了自适应工具调用与整合对于处理预测场景至关重要。禁用视觉推理、数值推理或共享记忆会在一定程度上增加均方误差（3.1%–5.4%），表明这些组件提供了互补信号，并共同提升了模型的稳健性。生成式推理器 变体未能明显优于默认的指令式锚点提取；不过，这一结果并不排除在更合适的设置下直接基于生成的可视化进行推理的潜力。总体而言，这些发现表明，MAS4TS 能够从多种推理模态的协同运用中获益，其中工具适配在有效利用可用组件方面发挥着核心作用。

超参数研究。我们分析了关键超参数对 ETTm1 在预测时域 $H =$ 下的影响

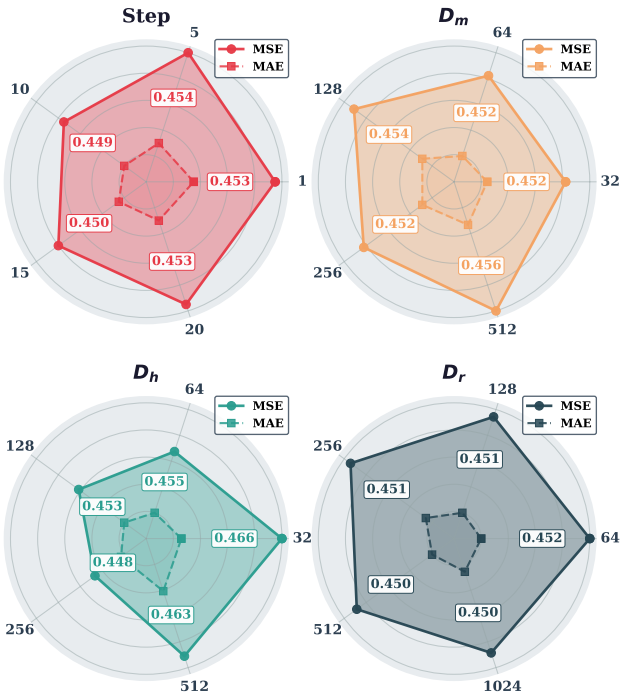

图 4 1 $H=$ 。ETTm（）上的超参数敏感性分析。每个雷达图显示以下设置下的均方误差（实线）和平均绝对误差（虚线）：a) 推理步数；b) 记忆维度 D_m；c) 隐藏维度 D_h；以及 d) 前馈维度 D_r。

720

720，如图4所示。雷达图展示了不同参数设置下均方误差和平均绝对误差的变化。

❶推理步数：随着步数从 1 增加到 20，性能逐步提升，在 10 步时达到 0.449 的最优均方误差，随后随着步数继续增加而略有下降。这表明，10 个积分步为潜在常微分方程求解器提供了足够的数值精度，同时避免了过高的计算开销。 ❷通信记忆的维度 D_m：模型对记忆维度变化表现出显著的鲁棒性，在所有设置（32-512）下，均方误差均窄幅变化于 0.452 至 0.456 之间。出于计算效率的考虑，我们采用 $D_m = 64$ 作为默认值。 ❸隐藏状态的维度 D_h：随着 D_h 从 32（均方误差=0.466）增加到 256（均方误差=0.448），性能持续提升；但在 512 时由于过拟合而下降。最优的 $D_h = 256$在模型容量与泛化能力之间实现了平衡。 ❹前馈维度 D_r：与 D_m类似，模型对 D_r表现出较低的敏感性，在 $D_r \geq 128$下均方误差稳定在约 0.450。我们将 $D_r = 512$设为默认值，在无需付出高额计算成本的情况下取得了略优的性能。

4.6 效率研究

为评估 MAS4TS 的计算效率，我们将其与具有代表性的基于预训练语言模型的时间序列方法在统一的实验设置下进行比较。所有测量均在 ETT 数据集的预测任务上进行，输入长度和预测长度均为 96。我们报告可训练参数量、峰值 GPU 显存消耗、每轮训练时间以及每批次的推理延迟。默认情况下，MAS4TS 使用遵循指令的 VLM 接口（*Qwen3-VL-235B-Instruct*）进行锚点提取，以平衡输出质量和效率。

表 5. MAS4TS 与基于预训练语言模型的时间序列基线方法在 ETTh1 预测任务上的效率对比（L=96, H=96 ）。我们报告可训练参数量、峰值GPU显存、每轮训练时间以及每批次推理延迟。MAS4TS 变体使用*Qwen3-VL-235B-Thinking*（思考）和*Nano-Banana*（生成式），基线方法则使用 GPT-2 进行文本编码。

模型	参数量（M）	显存（GB）	训练（秒）	推理（毫秒）
MAS4TS	**5.38**	**0.05**	**24.55**	**90.49**
- 思考	5.38	0.05	35.98	133.75
- 生成式	5.38	0.05	915.92	2767.01
Time-VLM	151.58	0.72	319.67	1067.14
UniTime†	107.32	5.18	53.49	165.77
TimeLLM†	134.90	15.96	853.75	2296.28

∗ 在 NVIDIA 4090 上以批次大小 32 进行实验。∗ † 表示本地语言模型的部署与推理，而我们通过 EAS API 进行推理。

表5 总结了效率比较结果。尽管 MAS4TS 将多智能体工作流与基础模型相结合，但它仍保持紧凑的骨干网络（5.38M 个参数）和极低的峰值 GPU 显存占用（0.05GB），其规模比基于预训练语言模型的方法（如 Time-VLM、UniTime 和 TimeLLM）小几个数量级。具体而言，这些基线方法的参数量超过100M，且会消耗数 GB 乃至数十 GB 的 GPU 显存；而 MAS4TS 则保持可学习核心轻量，并仅在需要时以小批次、跨样本并行的方式调用视觉语言模型。默认的指令式锚定方式在效率与质量之间实现了最佳权衡，并通过结构化、模式约束接口提取锚点，避免冗长的自由形式解码；相比之下，思考型和生成式变体都会带来更高的推理开销。结果表明，MAS4TS 的实际效率主要源于轻量级协调与结构化工具使用，而非扩展单体架构。

5 结论

我们提出了 MAS4TS，这是一种基于工具驱动的多智能体框架，用于通用时间序列分析，采用分析器–推理器–执行器范式。MAS4TS 将基于视觉语言模型的视觉锚定与锚点引导的潜在轨迹重建相结合，并进一步支持自适应路由和工具链执行，从而在统一协议下完成预测、分类、插补和异常检测。

参考资料

A. Abdulaal、Z. Liu 和 T. Lancewicki。异步多变量时间序列异常检测与定位的实用方法。载于第 *27* 届 *ACM SIGKDD* 知识发现与数据挖掘会议论文集，第 2485–2494 页，2021 年。

A. Bagnall、H. A. Dau、J. Lines、M. Flynn、J. Large、A. Bostrom、P. Southam 和 E. Keogh。《UEA 多变量时间序列分类档案》，2018*00075*，2018*1811*。*arXiv* 预印本 *arXiv:* 。

S. Bai, J. Z. Kolter 和 V. Koltun。通用卷积和循环网络用于序列建模的实证评估。*arXiv* 预印本 *arXiv:1803.01271*，2018。

D. J. Berndt 和 J. Clifford。利用动态时间规整在时间序列中寻找模式。发表于*KDD*研讨会，1994年。

G. E. Box、G. M. Jenkins、G. C. Reinsel 和 G. M. Ljung。时间序列分析：预测与控制。约翰·威立父子出版公司，2015年。

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhari- wal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell 等。语言模型是少样本学习者。载于神经信息处理系统进展，第 1877–1901 页，2020。

D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, 和 C. S. Jensen。Lightts：采用自适应集成蒸馏的轻量级时间序列分类。*ACM* 数据管理会议论文集，1(2):1–27，2023。

C. Chang, S.-D. Cao, W. Yang, H. Tsung, H. 等 Wang。基于大语言模型的时间序列推理与智能体系统综述。*arXiv* 预印本 *arXiv:2509.11575*，2025a。

C. Chang、W.-Y. Wang、W.-C. Peng 和 T.-F. Chen。Ll m4ts：将预训练大语言模型作为数据高效的时间序列预测器进行对齐。*ACM* 智能系统与技术汇刊，16(3):1–20，2025b。

M. Chen, L. Shen, Z. Li, X. J. Wang, J. Sun, 和 C. Liu。Visionts：视觉掩码自编码器是免费的零样本时间序列预测器，2024。网址 https://arxiv.org/abs/ 2408.17253。

T. Chen 和 C. Guestrin。XGBoost：一种可扩展的树提升系统。知识发现与数据挖掘会议，2016年。

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk 与 Y. Bengio。使用循环神经网络编码器-解码器学习短语表示以进行统计机器翻译。*arXiv*预印本 *arXiv:1406.1078*，2014年。

R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning 等。Stl：一种季节性-趋势分解。*J. off. Stat*，6 73，1990 1 3( )：–。

A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, 和 R. Yu。使用 TiDE 进行长期预测：时间序列密集编码器。*arXiv* 预印本 *arXiv:2304.08424*，2023。

C. Deb、F. Zhang、J. Yang、S. E. Lee 和 K. W. Shah。建筑能耗时间序列预测技术综述。《可再生能源与可持续能源评论》，74 924，2017 902：–。

J. R. Dormand 和 P. J. Prince。嵌入式龙格–库塔公式族。《计算与应用数学杂志》，6(1):19–26，1980。J.-Y. Franceschi、A. Dieuleveut 和 M. Jaggi。多变量时间序列的无监督可扩展表示学习。载于 *NeurIPS*，2019。A. Gu、K. Goel 和 C. R´e。利用结构化状态空间高效建模长序列。载于 *ICLR*，2022。Y. Gu、Y. Xiong、J. Mace、Y. Jiang、Y. Hu、B. Kasikci 和 P. Cheng。Argos：通过大语言模型自主生成规则的智能体时间序列异常检测。*arXiv* 预印本 *arXiv:2501.14170*，2025。

Y. He 和 J. Zhao。用于时间序列异常检测的时间卷积网络。载于《物理学杂志：会议系列》，第 1213 卷，页码 042050。IOP 出版社，2019。

S. Hochreiter 和 J. Schmidhuber。长短期记忆。神经计算。，1997a。

S. Hochreiter 和 J. Schmidhuber。长短期记忆。神经计算，9(8):1735–1780，1997b。

S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin 等。Metagpt：面向多智能体协作框架的元编程。载于第十二届国际学习表征会议，2023。

Q. Huang, Z. Zhou, Y. Li, K. Yang, B. Wang, and Y. Wang。众智一心：通过子任务专业化与智能体间协作进行时间序列预测。见第三十九届神经信息处理系统年会，2025。

X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, B. Lian, Y. Wang, 和 E. Chen。《大语言模型智能体规划理解：综述》。*arXiv* 预印本 *arXiv:2402.02716*，2024。

K. Hundman, V. Constantinou, C. Laporte, I. Colwell, 和 T. Soder- strom。使用长短期记忆网络和非参数动态阈值检测航天器异常。载于第*24*届 *ACM SIGKDD* 知识发现与数据挖掘国际会议论文集，第387–395页，2018年。

S. M. Idrees, M. A. Alam, 和 P. Agarwal。一种基于时间序列数据的股市波动性预测方法。*IEEE Access*，7：17287–17298，2019年。

M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen。Time-LLM：通过重新编程大语言模型进行时间序列预测。载于国际学习表征会议（*ICLR*），2024a。

M. Jin, Y. Zhang, W. Chen, K. Zhang, Y. Liang, B. Yang, J. Wang, S. Pan, and Q. Wen。大语言模型能告诉我们什么关于时间序列分析的信息。*arXiv* 预印本 *arXiv:2402.02713*，2024b。

Z. Karevan 和 J. A. Suykens。用于时间序列预测的传导式 LSTM：天气预报中的一项应用。神经网络，125：1–9，2020。

N. Kitaev, Ł. Kaiser 和 A. Levskaya。Reformer：高效的 Transformer。在国际学习表征会议，2020。

G. Lai、W.-C. Chang、Y. Yang 和 H. Liu。使用深度神经网络建模长短期时间模式。发表于 *SIGIR*，2018a。

G. Lai, W.-C. Chang, Y. Yang, 和 H. Liu. 利用深度神经网络建模长短期时间模式。在第*41*届国际 *ACM SIGIR*信息检索研究与发展会议，第95–104页，2018b。

S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, 和 X. Yan. 增强局部性并打破 Transformer 在时间序列预测中的记忆瓶颈。神经信息处理系统进展，32，2019。

Y. Liang、H. Wen、Y. Nie、Y. Jiang、M. Jin、D. Song、S. Pan和Q. Wen。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大会论文集，第6555–6565页，2024。B. Lim和S. Zohren。基于深度学习的时间序列预测：综述。英国皇家学会哲学汇刊*A*辑：数学、物理与工程科学，379(2194)，2021。M. Liu、A. Zeng、M. Chen、Z. Xu、Q. Lai、L. Ma和Q. Xu。SCINet：基于样本卷积与交互的时间序列建模与预测。神经信息处理系统进展，35:5816–5828，2022a。S. Liu、H. Yu、C. Liao、J. Li、W. Lin、A. X. Liu和S. Dustdar。Pyraformer：面向长程时间序列建模与预测的低复杂度金字塔注意力。载于国际学习表征会议，2022b。X. Liu、J. Hu、Y. Li、S. Diao、Y. Liang、B. Hooi和R. Zimmermann。UniTime：面向跨领域时间序列预测的语言赋能的统一模型。载于*2024*年*ACM Web*大会论文集，第4095–4106页，2024a。Y. Liu、H. Wu、J. Wang和M. Long。非平稳Transformer：探索时间序列预测中的平稳性。神经信息处理系统进展，35:9881–9893，2022c。Y. Liu、T. Hu、H. Zhang、H. Wu、S. Wang、L. Ma和M. Long。iTransformer：逆向Transformer对时间序列预测有效。载于第十二届国际学习表征会议，2024b。S. Lyu、S. Zhong、W. Ruan、Q. Liu、Q. Wen、H. Xiong和Y. Liang。OccamVTS：将视觉模型蒸馏为1个URL https://arxiv.org/abs/2508.01727。A. P. Mathur和N. O. Tippenhauer。SWaT：面向工业控制系统安全研究与培训的水处理测试平台。载于*2016*年信息物理系统智能水网络国际研讨会（*CySWater*），第31–36页。IEEE，2016年。L. R. Medsker、L. Jain等。循环神经网络。设计与应用，5(64-67):2，2001。Y. Nie、N. H. Nguyen、P. Sinthong和J. Kalagnanam。一个时间序列值64个词：基于Transformer的长期预测。载于国际学习表征会议，2023a。

Y. Nie、N. H. Nguyen、P. Sinthong 与 J. Kalagnanam。一条时间序列值64个词：基于Transformer的长期预测。见国际学习表征会议，2023b。C. Qian、W. Liu、H. Liu、N. Chen、Y. Dang、J. Li、C. Yang、W. Chen、Y. Su、X. Cong 等。Chatdev：面向软件开发的通信代理。*arXiv*预印本 *arXiv:2307.07924*，2023。Y. Qin、S. Liang、Y. Ye、K. Zhu、L. Yan、Y. Lu、Y. Lin、X. Cong、X. Tang、B. Qian 等。ToolLLM：帮助大语言模型掌握 16000+ 真实世界 API。*arXiv*预印本 *arXiv:2307.16789*，2023。C. Ravuru、S. S. Sakhinana 与 V. Runkana。面向时间序列分析的智能体检索增强生成。*arXiv*预印本 *arXiv:2408.14484*，2024。W. Ruan、S. Zhong、H. Wen 与 Y. Liang。基于潜在扩散模型的视觉增强时间序列预测。*arXiv*预印本 *arXiv:2502.14887*，2025。Y. Rubanova、R. T. Chen 与 D. K. Duvenaud。面向不规则采样时间序列的潜在常微分方程。神经信息处理系统进展，32，2019。T. Schick、J. Dwivedi-Yu、R. Dess`ı、R. Raileanu、M. Lomeli、E. Hambro、L. Zettlemoyer、N. Cancedda 与 T. Scialom。Toolformer：语言模型可以自学使用工具。神经信息处理系统进展，36:68539–68551，2023。S. H. Schneider 与 R. E. Dickinson。气候建模。地球物理学评论，12(3):447–493，1974。C. Shen、W. Yu、Z. Zhao、D. Song、W. Cheng、H. Chen 与 J. Ni。面向长期时间序列预测的多模态视图增强大型视觉模型。*arXiv*预印本 *arXiv:2505.24003*，2025。Y. Su、Y. Zhao、C. Niu、R. Liu、W. Sun 与 D. Pei。基于随机循环神经网络的多变量时间序列鲁棒异常检测。见第*25*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集，第2828–2837页，2019。H. Touvron、T. Lavril、G. Izacard、X. Martinet、M.-A. Lachaux、T. Lacroix、B. Rozi`ere、N. Goyal、E. Hambro、F. Azhar 等。Llama：开放且高效的基础语言模型。*arXiv*预印本 *arXiv:2302.13971*，2023。A. Van Den Oord、S. Dieleman、H. Zen、K. Simonyan、O. Vinyals、A. Graves、N. Kalchbrenner、A. Senior、K. Kavukcuoglu 等。WaveNet：面向原始音频的生成模型。*arXiv*预印本 *arXiv:1609.03499*，12:1，2016年。A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、Ł. Kaiser 与 I. Polosukhin。注意力就是你所需要的一切。见神经信息处理系统进展，第5998–6008页，2017。G. Woo、C. Liu、D. Sahoo、A. Kumar 与 S. Hoi。ETSformer：面向时间序列预测的指数平滑 Transformer。见*arXiv*预印本 *arXiv:2202.01381*，2022。H. Wu、J. Xu、J. Wang 与 M. Long。Autoformer：用于长期序列预测的带自相关的分解 Transformer。神经信息处理系统进展，34:22419–22430，2021。

H. Wu、T. Hu、Y. Liu、H. Zhou、J. Wang 与 M. Long。TimesNet：面向通用时间序列分析的时间二维变化建模。见国际学习表征会议，2023a。H. Wu、T. Hu、Y. Liu、H. Zhou、J. Wang 与 M. Long。TimesNet：面向通用时间序列分析的时间二维变化建模。见国际学习表征会议，2023b。Q. Wu、G. Bansal、J. Zhang、Y. Wu、B. Li、E. Zhu、L. Jiang、X. Zhang、S. Zhang、J. Liu 等。AutoGen：通过多智能体对话实现下一代大语言模型应用。见第一届语言建模会议，2024。J. Xu、H. Wu、J. Wang 与 M. Long。异常检测 Transformer：基于关联差异的时间序列异常检测。*arXiv*预印本 *arXiv:2110.02642*，2021。Z. Xu、A. Zeng 与 Q. Xu。FITS：用 $10k$ 参数建模时间序列。*arXiv*预印本 *arXiv:2307.03756*，2023。A. Zeng、M. Chen、L. Zhang 与 Q. Xu。Transformer对时间序列预测有效吗？见人工智能促进协会会议论文集，第11121–11128页，2023。Y. Zhang 与 J. Yan。Crossformer：利用跨维度依赖进行多变量时间序列预测的Transformer。*ICLR*，2023。J. Zheng 与 M. Huang。基于深度学习的时间序列分析交通流量预测。*IEEE Access*，8 82570，2020 82562: – 。S. Zhong、W. Ruan、M. Jin、H. Li、Q. Wen 与 Y. Liang。Time-VLM：探索多模态视觉语言模型以增强时间序列预测。*arXiv*预印本 *arXiv:2502.04395*，2025。H. Zhou、S. Zhang、J. Peng、S. Zhang、J. Li、H. Xiong 与 W. Zhang。Informer：超越高效Transformer的长序列时间序列预测。见人工智能促进协会会议论文集，第11106–11115页，2021。T. Zhou、Z. Ma、Q. Wen、X. Wang、L. Sun 与 R. Jin。FEDformer：面向长期序列预测的频率增强分解式 Transformer。见国际机器学习会议，第27268–27286页，2022。T. Zhou、P. Niu、X. Wang、L. Sun 与 R. Jin。One Fits All：借助预训练语言模型增强通用时间序列分析。见神经信息处理系统进展，2023。Z. Zhou、G. Lyu、Y. Huang、Z. Wang、Z. Jia 与 Z. Yang。SD-former：用于多变量时间序列长期预测的带谱滤波器和动态注意力的Transformer。Z. Zhou、J. Hu、Q. Wen、J. T. Kwok 与 Y. Liang。面向深度时间序列预测的多阶小波导数变换。*arXiv*预印本 *arXiv:2505.11781*，2025。

A 实验详情

A.1 数据集详情

表*6.* 数据集描述。数据集规模按（训练集、验证集、测试集）排列。

任务	数据集	Dim	序列长度	数据集大小	信息（频率）
长期预测	ETTm1, ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	电力（15分钟）
	ETTh1、ETTh2	7	{96, 192, 336, 720}	（8545，2881，2881）	电力（每小时）
	天气	21	{96, 192, 336, 720}	（36792，5271，10540）	天气（10分钟）
	太阳能	137	{96, 192, 336, 720}	(36345, 5065, 10321)	太阳能（10分钟）
	汇率	8	{96, 192, 336, 720}	(5120, 665, 1422)	汇率（每日）
插补	ETTm1, ETTm2	7	96	(34465, 11521, 11521)	电力（15分钟）
	ETTh1, ETTh2	7	96	(8545, 2881, 2881)	电力（每小时）
	电力	321	96	(18317, 2633, 5261)	电力（每小时）
	天气	21	96	(36792, 5271, 10540)	天气（10分钟）
	汇率	8	96	(5120, 665, 1422)	汇率（每日）
分类 (UEA)	乙醇浓度	3	1751	(261, 0, 263)	酒精行业
	人脸检测	144	62	(5890, 0, 3524)	人脸（250Hz）
	手写	3	152	(150, 0, 850)	手写
	心跳	61	405	(204, 0, 205)	心跳
	日语元音	12	29	(270, 0, 370)	语音
	自我调节SCP1	6	896	(268, 0, 293)	健康（256赫兹）
	自我调节SCP2	7	1152	(200, 0, 180)	健康（256赫兹）
	UWave手势库	3	315	(120, 0, 320)	手势
异常检测	SMD	38	100	(566724, 141681, 708420)	服务器机器
	MSL	55	100	(44653, 11664, 73729)	航天器
	SMAP	25	100	(108146, 27037, 427617)	航天器
	SWaT	51	100	(396000, 99000, 449919)	基础设施
	PSM	25	100	(105984, 26497, 87841)	服务器机器

如表 6所示，基准数据集涵盖了多样的领域和时间粒度：

• **长期预测**：7 个数据集，涵盖电力变压器温度（ETT系列，含 4 个子集）、电力消耗（电力）、交通流量（Traffic）、气象指标（天气）、太阳能发电（太阳能）和货币汇率（汇率）。

• **插补**：采用相同的预测数据集，人工引入随机缺失模式，掩码比例设为 12.5%、25%、37.5% 和 50%，遵循 Wu 等人，2023b 所确立的协议。

• **分类**：从 UEA 多变量时间序列档案中选取 8 个具有代表性的数据集（Bagnall 等人，2018），涵盖化学传感（乙醇浓度）、视频分析（人脸检测）、动作捕捉（手写、UWave手势库）、医学诊断（心跳）、音频识别（日语元音）和脑机接口（自我调节SCP1、自我调节SCP2）。

• **异常检测**：5 个真实 世界数据集，涵盖服务器设备（SMD、PSM）、航天器遥测系统（MSL、SMAP）以及水处理 r 基础设施（S WaT）。

A.2 基线方法

我们使用官方代码库和/或原论文中报告的设置复现了所有基线方法。为确保公平比较，在适用的情况下，我们保持神经网络模型之间的输入嵌入和最终投影接口一致，并重点比较其核心序列建模或特定任务机制。需要注意的是，部分基线方法仅在特定任务上进行评估。

（1）时间序列专用模型。

• **基于 Transformer 的时间序列模型**:

– iTransformer（Liu 等人，2024b）：在变量之间执行自注意力（反向标记化），以更好地建模多变量依赖关系。– PatchTST（Nie 等人，2023b）：采用分块和通道独立性，缩短序列长度并稳定长时域学习。– Crossformer（Zhang 和 Yan，2023）：通过两阶段注意力设计（维度内／维度间）建模跨维度依赖关系。– 非平稳 Transformer（Liu 等人，2022c）（报告为 Stationary）：通过序列平稳化和去平稳化注意力缓解分布偏移。– Autoformer（Wu 等人，2021）：以自相关和渐进式序列分解取代逐点注意力，用于长期预测。– FEDformer（Zhou 等人，2022）：在谱域中进行频率增强分解和注意力计算，以实现高效的长序列建模。– ETSformer（Woo 等人，2022）：引入指数平滑（ETS 风格）组件，明确捕捉水平、趋势和季节性。– Informer（Zhou 等人，2021）：采用 ProbSparse 注意力，降低长序列预测的注意力计算成本。– Reformer（Kitaev 等人，2020）：使用 LSH 注意力，以较低的记忆和计算成本近似 softmax 注意力。– Pyraformer（Liu 等人，2022b）：在多个时间分辨率上构建金字塔注意力，以建模长程结构。

• **CNN/MLP 风格时间序列模型**:

– TimesNet（Wu等人，2023b）：在多重周期性的引导下将一维序列重塑为二维表示，并应用二维卷积块。– DLinear（Zeng 等人，2023）：将序列分解为趋势和残差部分，并用轻量线性层进行预测。– LightTS（Campos 等人，2023）：利用高效的 MLP 混合以及区间/连续采样来处理长上下文。– TiDE（Das 等人，2023）：通过稠密残差块对带协变量的回看窗口进行编码，并用逐时间步时序解码器解码未来值。– SCINet（Liu 等人，2022a）：递归地拆分并交互子序列，以捕获层级时序依赖关系。– 时间卷积网络（He 和 Zhao，2019）：使用扩张因果卷积在无循环的情况下建模长感受野。

• **RNN/状态空间风格模型**:

– 长短期记忆网络 (Hochreiter 和 Schmidhuber, 1997b)：采用循环门控机制来学习长期/短期时序依赖关系。– *LSTNet* (Lai 等人, 2018b)：结合 CNN 特征提取与 RNN 时序建模，并利用跳跃连接捕捉周期模式。– *LSSL* (Gu 等人, 2022)：将 HiPPO 初始化的状态空间模型重新参数化为对角加低秩形式，并通过高效的柯西核卷积实现近线性时间的长序列建模。。

(2) 基于预训练语言模型的基线。

• *MAFS* (Huang 等人, 2025)：采用多智能体框架，由每个智能体专门负责一个预测子任务，并通过智能体间通信和投票聚合器生成最终预测。（注：我们将其列于此处，是因为它采用了先预训练再微调范式，不过它是在子任务上预训练 iTransformer，而非利用大语言模型。）

• *Time-VLM* (Zhong 等人, 2025)：通过将时间序列转换为图像，同时根据统计特征生成上下文文本提示。冻结的视觉语言模型对这两种模态进行编码，并通过跨模态注意力将其与时间记忆特征融合，用于预测。

• *UniTime* (Liu 等人, 2024a)：采用带有领域指令的预训练语言模型进行领域识别并缓解领域混淆，同时应用掩码来平衡不同领域的收敛速度。Language-TS Transformer 将来自不同输入空间的时间序列对齐到语言模型的共同潜在空间，以实现跨域泛化。

（3）经典基线方法及其他。

• *Transformer*（Vaswani等人，2017）：作为通用序列基线的经典编码器-解码器注意力架构。

• *DTW*（Berndt和Clifford，1994年）：使用动态时间规整距离（通常搭配1近邻），作为时间序列分类的强大经典基线方法。

• *XGBoost*（Chen 和 Guestrin，2016年）：在构造的时间特征/统计数据上训练的梯度提升树。

对于异常检测，我们还将其与 异常检测 *Transformer*（Xu等，2021）进行比较。该方法专为此任务设计，通过时间序列级关联与先验关联之间的关联差异来识别异常。除了任务特定模型，我们还考虑了在实验结果中仍极具竞争力的经典方法。例如，在分类任务中，*DTW*、*XGBoost*和原版 *Transformer* 均展现出强劲的性能，因此也被纳入作为额外的基线方法。

A.3 评估指标

对于各项指标，我们针对每项任务采用不同的评估标准：

长期预测和插补。 我们使用均方误差（MSE）和平均绝对误差（MAE）：

$$\text{MSE} = \frac{1}{H \cdot C} \sum_{i=1}^{H} \sum_{j=1}^{C} (\mathbf{X}_{i,j} - \widehat{\mathbf{X}}_{i,j})^2, \qquad \text{MAE} = \frac{1}{H \cdot C} \sum_{i=1}^{H} \sum_{j=1}^{C} |\mathbf{X}_{i,j} - \widehat{\mathbf{X}}_{i,j}|,$$

其中，$\mathbf{X}$ $\widehat{\mathbf{X}}$ $\in \mathbb{R}^{H \times C}$ 表示真实值和预测值，其包含 H 个时间步和 C 个特征维度。值越低表示性能越好。

分类。 我们采用准确率（ACC）作为主要评估指标：

$$\text{ACC} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \times 100\%.$$

准确率越高，表示分类性能越好。

异常检测。 我们遵循点调整协议（Xu等，2021），采用精确率（P）、召回率（R）和F1分数（F1）：

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

其中，TP、FP 和 FN 分别表示真正例、假正例和假负例。F1分数综合衡量精确率和召回率，值越高表示检测性能越好。

A.4 模型与智能体配置

本节全面概述 MAS4TS 中的模型超参数、智能体专属配置和共享内存机制。该框架采用模块化设计，每个智能体都维护专门的参数集，并通过统一记忆接口进行协作。

训练配置。表7 总结了任务特定训练配置。所有实验均使用 AdamW优化器，并采用默认动量参数 $(\beta_1, \beta_2) = (0。9, 0。999)$，权重衰减为 10^{-2}。我们采用耐心值为 3 轮的提前停止机制，以防止过拟合，并使用余弦退火进行学习率调度。默认启用混合精度训练（FP16），以加速计算，同时保持数值稳定性。

表7。任务特定训练配置（适用于 MAS4TS）。嵌入维度 d_{model} 和网络深度保持一致以确保不同任务间的公平比较，同时针对各项任务调整学习率和批次大小，以实现最优收敛。

Task	架构				训练			
	d_{model}	层数	补丁	步幅	LR	Loss	批次	训练轮数
长期预测	64	2	16	8	10^{-4}	MSE	32	10
插补	64	2	16	8	10^{-3}	MSE	16	10
分类	64	2	16	8	10^{-3}	CE	16	30
异常检测	64	3	16	8	10^{-4}	MSE	128	10

LR：采用余弦退火衰减的初始学习率。CE：交叉熵损失。

消融标志。 MAS4TS通过可配置标志支持细粒度消融研究，这些标志可启用或禁用各个组件：

• 启用视觉推理器：切换基于视觉语言模型的锚点提取（默认：True）
• 启用数值推理器：切换神经常微分方程序列重建（默认：True）

• 启用共享记忆：切换跨智能体记忆共享（默认：True）

• 启用门控注意力：切换门控或简单多层感知机通信（默认：True）

• 启用工具：切换工具增强执行或简单多层感知机头（默认：True）

• 完成策略：常微分方程求解器替代方案：{'ode', 'linear', 'quadratic', 'repeat'}

表 *8.* MAS4TS 模型架构参数

参数	值	描述
全局配置		
d_模型	64	所有智能体的嵌入维度
d_记忆	64	共享内存维度
补丁长度	16	分词的补丁长度
步幅	8	补丁间步长
数据分析智能体		
前 k 个特征	10	用于协方差选择的前 k 个特征
特征方法	协方差	特征选择标准
视觉推理智能体		
视觉语言模型_	Qwen3-VL-235B-Instruct	视觉语言模型骨干网络
最大锚点	20	每个样本的最大锚点数
置信度阈值	0.7	锚点接纳的最小置信度
数值推理智能体		
隐藏维度	128	常微分方程动力学的隐藏维度
常微分方程求解器_	RK4	常微分方程积分方法
常微分方程步长_	0.05	积分步长
任务执行器智能体		
e层数	2	任务工具中的编码器层
n个头部	4	注意力头
丢弃	0.1	丢弃率

A.5 实现细节

我们在表 6 和 7中提供了数据集描述和实验配置。所有实验均在一台 NVIDIA GeForce RTX 4090 GPU（48GB 版本）上进行，并使用固定的随机种子以确保结果可复现。MAS4TS 框架基于 PyTorch 2.9 实现。1+cu128 和 CUDA 12.8。对于视觉语言模型组件，我们通过阿里云平台的 EAS API 服务使用 *Qwen2-VL-235B-Instruct* 模型。

B 完整结果

本附录提供四项评估任务的完整实验结果：长期预测、时间序列分类、插补和异常检测。

B.1 长期预测

表 9 展示了所有数据集和预测范围上的完整预测结果 $H \in \{96, 192, 336, 720\}$。按照以往研究的做法（Wu 等人，2023a），所有方法的输入序列长度均固定为 $L = 96$。与任务特定模型（如 PatchTST、iTransformer）和基于预训练语言模型的方法（如 UniTime、TimeVLM）相比，MAS4TS 始终取得具有竞争力或更优的性能。

B.2 分类

表10 展示了 UEA 多元时间序列档案中 8 个代表性数据集的分类准确率（Bagnall等人，2018）。这些数据集涵盖多个应用领域，包括化学传感、视频分析、动作捕捉、医疗诊断和音频识别。

表*9*。长期预测任务的完整结果。我们比较了多种具有竞争力的模型在不同预测长度下的表现。所有数据集的输入序列长度均设为 96。平均值为四种预测长度的平均值，即 $\{96, 192, 336, 720\}$。**红色粗体**：最佳结果；蓝色下划线：次佳结果。

模型		**MAS4TS** (我们的方法)		MAFS 2025		TimeVLM 2025		UniTime 2024a		iTransformer 2024b		PatchTST 2023b		Crossformer 2023		TiDE 2023		TimesNet 2023a		DLinear 2023		SCINet 2022a		FEDformer 2022		Stationary 2022c		Autoformer 2021	
指标		均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差
Weather	96	**0.147**	**0.197**	0.166	0.208	0.181	0.220	0.171	0.214	0.174	0.214	0.186	0.227	0.195	0.271	0.202	0.261	0.172	0.220	0.195	0.252	0.221	0.306	0.217	0.296	0.173	0.223	0.266	0.336
	192	**0.192**	**0.244**	0.218	0.253	0.227	0.260	0.217	0.254	0.221	0.254	0.234	0.265	0.209	0.277	0.242	0.298	0.219	0.261	0.237	0.295	0.261	0.340	0.276	0.336	0.245	0.285	0.307	0.367
	336	**0.245**	**0.286**	0.274	0.295	0.281	0.298	0.274	0.293	0.278	0.296	0.284	0.301	0.273	0.332	0.287	0.335	0.280	0.306	0.282	0.331	0.309	0.378	0.339	0.380	0.321	0.338	0.359	0.395
	720	**0.328**	**0.339**	0.351	0.345	0.358	0.348	0.351	0.343	0.358	0.347	0.356	0.349	0.379	0.401	0.351	0.386	0.365	0.359	0.345	0.382	0.377	0.427	0.403	0.428	0.414	0.410	0.419	0.428
	Avg	**0.228**	**0.267**	0.252	0.275	0.262	0.281	0.253	0.276	0.258	0.278	0.265	0.285	0.264	0.320	0.271	0.320	0.259	0.287	0.265	0.315	0.292	0.363	0.309	0.360	0.288	0.314	0.338	0.382
Solar-Energy	96	**0.192**	0.259	0.198	**0.237**	0.220	0.269	0.239	0.301	0.203	0.237	0.265	0.323	0.232	0.302	0.312	0.399	0.373	0.358	0.290	0.378	0.237	0.344	0.286	0.341	0.321	0.380	0.456	0.446
	192	**0.207**	0.271	0.231	0.263	0.257	0.293	0.269	0.316	0.233	**0.261**	0.288	0.332	0.371	0.410	0.339	0.416	0.397	0.376	0.320	0.398	0.280	0.380	0.291	0.337	0.346	0.369	0.588	0.561
	336	**0.219**	0.283	0.250	0.278	0.271	0.303	0.288	0.324	0.248	**0.273**	0.301	0.339	0.495	0.515	0.368	0.430	0.420	0.380	0.353	0.415	0.304	0.389	0.354	0.416	0.357	0.387	0.595	0.588
	720	**0.231**	0.293	0.255	0.281	0.287	0.307	0.285	0.320	0.249	**0.275**	0.295	0.336	0.526	0.542	0.370	0.425	0.420	0.381	0.357	0.413	0.308	0.388	0.380	0.437	0.375	0.424	0.733	0.633
	Avg	**0.212**	0.276	0.234	0.265	0.259	0.293	0.270	0.315	0.233	**0.262**	0.287	0.333	0.406	0.442	0.347	0.417	0.403	0.374	0.330	0.401	0.282	0.375	0.328	0.383	0.350	0.390	0.586	0.557
Exchange	96	0.088	0.208	0.084	0.204	**0.082**	**0.198**	0.125	0.245	0.086	0.206	0.088	0.205	0.256	0.367	0.094	0.218	0.107	0.234	0.088	0.218	0.267	0.396	0.148	0.278	0.111	0.237	0.197	0.323
	192	0.181	0.303	0.180	0.304	**0.175**	**0.296**	0.236	0.343	0.177	0.299	0.176	0.299	0.470	0.509	0.184	0.307	0.226	0.344	0.176	0.315	0.351	0.459	0.271	0.315	0.219	0.335	0.300	0.369
	336	**0.293**	**0.396**	0.316	0.406	0.324	0.412	0.420	0.470	0.331	0.417	0.301	0.397	1.268	0.883	0.349	0.431	0.367	0.448	0.313	0.427	1.324	0.853	0.460	0.427	0.421	0.476	0.509	0.524
	720	**0.810**	**0.682**	1.002	0.755	0.840	0.689	1.078	0.791	0.847	0.691	0.901	0.714	1.767	1.068	0.852	0.698	0.964	0.746	0.839	0.695	1.058	0.797	1.195	0.695	1.092	0.769	1.447	0.941
	Avg	**0.343**	**0.397**	0.396	0.417	0.355	0.399	0.465	0.462	0.360	0.403	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414	0.750	0.626	0.519	0.429	0.461	0.454	0.613	0.539
ETTh1	96	0.378	0.399	0.389	0.403	**0.376**	**0.395**	0.397	0.418	0.386	0.405	0.460	0.447	0.423	0.448	0.479	0.464	0.384	0.402	0.397	0.412	0.654	0.599	0.395	0.424	0.513	0.491	0.449	0.459
	192	**0.419**	0.428	0.450	0.438	0.426	**0.424**	0.434	0.439	0.441	0.512	0.477	0.429	0.471	0.474	0.525	0.492	0.436	0.429	0.446	0.441	0.719	0.631	0.469	0.470	0.534	0.504	0.500	0.482
	336	**0.459**	**0.445**	0.475	0.447	0.469	0.449	0.468	0.457	0.487	0.458	0.546	0.496	0.570	0.546	0.565	0.515	0.491	0.469	0.489	0.467	0.778	0.659	0.530	0.499	0.588	0.535	0.521	0.496
	720	0.503	0.501	0.487	**0.472**	0.559	0.504	**0.469**	0.477	0.503	0.491	0.544	0.517	0.653	0.621	0.594	0.558	0.521	0.500	0.513	0.510	0.836	0.699	0.598	0.544	0.643	0.616	0.514	0.512
	Avg	**0.440**	**0.438**	0.450	0.440	0.457	0.443	0.442	0.448	0.454	0.447	0.516	0.484	0.529	0.522	0.541	0.507	0.458	0.450	0.461	0.457	0.747	0.647	0.498	0.484	0.570	0.537	0.496	0.487
ETTh2	96	0.296	0.345	0.298	0.346	**0.290**	**0.339**	0.296	0.345	0.297	0.349	0.308	0.355	0.745	0.584	0.400	0.440	0.340	0.374	0.340	0.394	0.707	0.621	0.358	0.397	0.476	0.458	0.346	0.388
	192	0.386	0.399	0.380	0.398	**0.374**	**0.389**	0.374	0.394	0.380	0.400	0.393	0.405	0.877	0.656	0.528	0.509	0.402	0.414	0.482	0.479	0.860	0.689	0.429	0.439	0.512	0.493	0.456	0.452
	336	**0.407**	**0.422**	0.413	0.427	0.433	0.440	0.415	0.427	0.428	0.432	0.427	0.436	1.043	0.731	0.643	0.571	0.452	0.452	0.591	0.541	1.000	0.744	0.496	0.487	0.552	0.551	0.482	0.486
	720	0.427	**0.442**	0.428	0.444	0.438	0.452	**0.425**	0.444	0.427	0.445	0.436	0.450	1.104	0.763	0.874	0.679	0.462	0.468	0.839	0.661	1.249	0.838	0.463	0.474	0.562	0.560	0.515	0.511
	Avg	**0.379**	**0.402**	0.380	0.403	0.384	0.405	0.379	0.403	0.383	0.407	0.391	0.411	0.942	0.684	0.611	0.550	0.414	0.427	0.563	0.519	0.954	0.723	0.437	0.449	0.526	0.516	0.450	0.459
ETTm1	96	**0.296**	**0.341**	0.329	0.362	0.326	0.365	0.322	0.363	0.334	0.368	0.352	0.374	0.404	0.426	0.364	0.387	0.338	0.375	0.346	0.374	0.418	0.438	0.379	0.419	0.386	0.398	0.505	0.475
	192	**0.341**	**0.366**	0.370	0.385	0.368	0.384	0.366	0.387	0.390	0.393	0.374	0.387	0.450	0.451	0.398	0.404	0.374	0.387	0.382	0.391	0.439	0.450	0.426	0.441	0.459	0.444	0.553	0.496
	336	**0.385**	**0.390**	0.409	0.410	0.394	0.405	0.398	0.407	0.426	0.420	0.421	0.414	0.532	0.515	0.428	0.425	0.410	0.411	0.415	0.415	0.490	0.485	0.445	0.459	0.495	0.464	0.621	0.537
	720	**0.445**	**0.426**	0.478	0.451	0.462	0.440	0.454	0.440	0.491	0.459	0.462	0.449	0.666	0.589	0.487	0.461	0.478	0.450	0.473	0.451	0.595	0.550	0.543	0.490	0.585	0.516	0.671	0.561
	Avg	**0.367**	**0.381**	0.397	0.402	0.388	0.398	0.385	0.399	0.407	0.410	0.406	0.407	0.513	0.495	0.419	0.419	0.400	0.406	0.404	0.408	0.485	0.481	0.448	0.452	0.481	0.456	0.588	0.517
ETTm2	96	**0.175**	**0.260**	0.180	0.264	0.176	0.261	0.183	0.266	0.180	0.264	0.183	0.270	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.293	0.286	0.377	0.203	0.287	0.192	0.274	0.255	0.339
	192	**0.237**	**0.302**	0.245	0.304	0.240	0.302	0.251	0.310	0.250	0.309	0.255	0.314	0.414	0.492	0.290	0.364	0.249	0.309	0.284	0.361	0.399	0.445	0.269	0.328	0.280	0.339	0.281	0.340
	336	0.303	0.343	0.316	0.351	**0.299**	**0.339**	0.319	0.351	0.311	0.348	0.309	0.347	0.597	0.542	0.377	0.422	0.321	0.351	0.382	0.429	0.637	0.591	0.325	0.366	0.334	0.361	0.339	0.372
	720	**0.396**	**0.400**	0.409	0.403	0.400	0.416	0.420	0.410	0.412	0.407	0.412	0.404	1.730	1.042	0.558	0.524	0.408	0.403	0.558	0.525	0.960	0.735	0.421	0.415	0.417	0.413	0.433	0.432
	Avg	**0.278**	**0.326**	0.288	0.330	0.279	0.329	0.293	0.334	0.288	0.332	0.290	0.334	0.757	0.610	0.358	0.404	0.291	0.333	0.354	0.402	0.954	0.723	0.305	0.349	0.306	0.347	0.327	0.371
1^{st}计数27		22		0	2	6	7	2	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表*10*。 分类任务的完整结果。 $*$. 在模型名称中表示 $*$前身的名称。我们报告分类准确率（%）作为结果。标准差在0.1%以内。红色粗体：最佳结果；蓝色下划线：次佳结果。

方法	DTW	XGBoost	长短期记忆网络	LSTNet	LSSL	TCN	Trans.	Re.	In.	Pyra.	Auto.	FED.	ETS.	iTransformer	DLinear	LightTS	TiDE	**MAS4TS**
数据集	(1994年)	(2016年)	(1997a)	(2018a)	(2022)	(2019)	(2017)	(2020)	(2021)	(2022b)	(2021)	(2022)	(2022)	(2024b)	(2023)	(2023)	(2023)	(我们的方法)
乙醇浓度	32.3	43.7	32.3	39.9	31.1	28.9	32.7	31.9	31.6	30.8	31.6	28.1	31.2	28.1	32.6	29.7	27.1	33.46
人脸检测	52.9	63.3	57.7	65.7	66.7	52.8	67.3	68.6	67.0	65.7	68.4	66.0	66.3	66.3	68.0	67.5	65.3	69.38
手写	28.6	15.8	15.2	25.8	24.6	53.3	32.0	27.4	32.8	29.4	36.7	28.0	32.5	24.2	27.0	26.1	23.2	33.64
心跳	71.7	73.2	72.2	77.1	72.7	75.6	76.1	77.1	80.5	75.6	74.6	73.7	71.2	75.6	75.1	75.1	74.6	78.04
日语元音	94.9	86.5	79.7	98.1	98.4	98.9	98.7	97.8	98.9	98.4	96.2	98.4	95.9	96.6	96.2	96.2	95.6	97.56
自我调节SCP1	77.7	84.6	68.9	84.0	90.8	84.6	92.2	90.4	90.1	88.1	84.0	88.7	89.6	90.2	87.3	89.8	89.2	91.81
自我调节SCP2	53.9	48.9	46.6	52.8	52.2	55.6	53.9	56.7	53.3	53.3	50.6	54.4	55.0	54.4	50.5	51.1	53.4	55.55
UWave手势库	90.3	75.9	41.2	87.8	85.9	88.4	85.6	85.6	85.6	83.4	85.9	85.3	85.0	85.9	82.1	80.3	84.9	86.56
平均准确率	62.78	61.48	51.72	66.40	65.30	67.26	67.31	66.93	59.93	65.58	66.00	65.32	65.83	65.16	64.85	64.47	64.16	**68.25**

表*11*。插补任务的完整结果。我们随机遮蔽12.5%、25%、37.5%和50%的时间点，以比较模型在不同缺失程度下的性能。∗中的Transformer表示∗前身的名称。

模型		MAS4TS (我们的方法)		ETS. (2022)		LightTS* (2022)		DLinear* (2023)		FED. (2022)		Pyra. (2021a)		In. (2021)		Re. (2020)		LSTM (1997年)		TCN (2019)		LSSL (2022)	
掩码比例		均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差
ETTm1	12.5%	0.046	0.137	0.067	0.188	0.075	0.180	0.058	0.162	0.035	0.135	0.670	0.541	0.047	0.155	**0.032**	**0.126**	0.974	0.780	0.510	0.493	0.101	0.231
	25%	0.049	**0.143**	0.096	0.229	0.093	0.206	0.080	0.193	0.052	0.166	0.689	0.553	0.063	0.180	**0.042**	0.146	1.032	0.807	0.518	0.500	0.106	0.235
	37.5%	0.054	**0.150**	0.133	0.271	0.113	0.231	0.103	0.219	0.069	0.191	0.737	0.581	0.079	0.200	**0.052**	0.158	0.999	0.792	0.516	0.499	0.116	0.246
	50%	0.061	**0.159**	0.186	0.323	0.134	0.255	0.132	0.248	0.089	0.218	0.770	0.605	0.093	0.218	**0.063**	0.173	0.952	0.763	0.519	0.496	0.129	0.260
	Avg	0.052	**0.147**	0.120	0.253	0.104	0.218	0.093	0.206	0.062	0.177	0.717	0.570	0.071	0.188	**0.050**	0.154	0.989	0.786	0.516	0.497	0.113	0.254
ETTm2	12.5%	**0.026**	**0.095**	0.108	0.239	0.034	0.127	0.062	0.166	0.056	0.159	0.394	0.470	0.133	0.270	0.108	0.228	1.013	0.805	0.307	0.441	0.150	0.298
	25%	**0.029**	**0.100**	0.164	0.294	0.042	0.143	0.085	0.196	0.080	0.195	0.421	0.482	0.135	0.272	0.136	0.262	1.039	0.814	0.263	0.402	0.159	0.306
	37.5%	**0.031**	**0.105**	0.237	0.356	0.051	0.159	0.106	0.222	0.110	0.231	0.478	0.521	0.155	0.293	0.175	0.300	0.917	0.744	0.250	0.396	0.180	0.321
	50%	**0.034**	**0.111**	0.323	0.421	0.059	0.174	0.131	0.247	0.156	0.276	0.568	0.560	0.200	0.333	0.211	0.329	1.140	0.835	0.246	0.389	0.210	0.353
	Avg	**0.030**	**0.103**	0.208	0.327	0.046	0.151	0.096	0.208	0.101	0.215	0.465	0.508	0.156	0.292	0.157	0.280	1.027	0.800	0.266	0.407	0.175	0.324
ETTh1	12.5%	0.104	0.211	0.126	0.263	0.240	0.345	0.151	0.267	**0.070**	**0.190**	0.857	0.609	0.114	0.234	0.074	0.194	1.265	0.896	0.599	0.554	0.422	0.461
	25%	0.119	**0.227**	0.169	0.304	0.265	0.364	0.180	0.292	0.106	0.236	0.829	0.672	0.140	0.262	**0.102**	0.227	1.262	0.883	0.610	0.567	0.412	0.456
	37.5%	0.137	**0.243**	0.220	0.347	0.296	0.382	0.215	0.318	**0.124**	0.258	0.830	0.675	0.174	0.293	0.135	0.261	1.200	0.867	0.628	0.577	0.421	0.461
	50%	**0.158**	**0.261**	0.293	0.402	0.334	0.404	0.257	0.347	0.165	0.299	0.854	0.691	0.215	0.325	0.179	0.298	1.174	0.849	0.648	0.587	0.443	0.473
	Avg	0.129	**0.235**	0.202	0.329	0.284	0.373	0.201	0.306	**0.117**	0.246	0.842	0.682	0.161	0.279	0.122	0.245	1.225	0.873	0.621	0.571	0.424	0.481
ETTh2	12.5%	**0.062**	**0.161**	0.187	0.319	0.101	0.231	0.100	0.216	0.095	0.212	0.976	0.754	0.305	0.431	0.163	0.289	2.060	1.120	0.410	0.494	0.521	0.555
	25%	**0.063**	**0.160**	0.279	0.390	0.115	0.246	0.127	0.247	0.137	0.258	1.037	0.774	0.322	0.444	0.206	0.331	2.007	1.105	0.419	0.490	0.487	0.535
	37.5%	**0.068**	**0.168**	0.400	0.465	0.126	0.257	0.158	0.276	0.187	0.304	1.107	0.800	0.353	0.462	0.252	0.370	2.033	1.111	0.429	0.498	0.487	0.529
	50%	**0.074**	**0.176**	0.602	0.572	0.136	0.268	0.183	0.299	0.232	0.341	1.193	0.838	0.369	0.472	0.316	0.419	2.054	1.119	0.467	0.529	0.484	0.523
	Avg	**0.067**	**0.166**	0.367	0.436	0.119	0.250	0.142	0.259	0.163	0.279	1.079	0.792	0.337	0.452	0.234	0.352	2.039	1.114	0.431	0.503	0.495	0.475
Electricity	12.5%	**0.070**	**0.188**	0.196	0.321	0.102	0.229	0.092	0.214	0.107	0.237	0.297	0.383	0.218	0.326	0.190	0.308	0.277	0.366	0.621	0.620	0.217	0.341
	25%	**0.079**	**0.199**	0.207	0.332	0.121	0.252	0.118	0.247	0.120	0.251	0.294	0.380	0.219	0.326	0.197	0.312	0.281	0.369	0.559	0.585	0.219	0.341
	37.5%	**0.090**	**0.201**	0.219	0.344	0.141	0.273	0.144	0.276	0.136	0.266	0.296	0.381	0.222	0.328	0.203	0.315	0.275	0.364	0.567	0.588	0.223	0.343
	50%	**0.107**	**0.221**	0.235	0.357	0.160	0.293	0.175	0.305	0.158	0.284	0.299	0.383	0.228	0.331	0.210	0.319	0.273	0.361	0.581	0.597	0.229	0.347
	Avg	**0.086**	**0.202**	0.214	0.339	0.131	0.262	0.132	0.260	0.130	0.259	0.297	0.382	0.222	0.328	0.200	0.313	0.277	0.365	0.582	0.597	0.222	0.293
Weather	12.5%	0.035	0.090	0.057	0.141	0.047	0.101	0.039	0.084	0.041	0.107	0.140	0.220	0.037	0.093	**0.031**	**0.076**	0.296	0.379	0.176	0.287	0.036	0.095
	25%	**0.035**	**0.062**	0.065	0.155	0.052	0.111	0.048	0.103	0.064	0.163	0.147	0.229	0.042	0.100	0.035	0.082	0.327	0.409	0.187	0.293	0.042	0.104
	37.5%	**0.037**	**0.065**	0.081	0.180	0.058	0.121	0.057	0.117	0.107	0.229	0.156	0.240	0.049	0.111	0.040	0.091	0.406	0.463	0.172	0.281	0.047	0.112
	50%	**0.036**	**0.063**	0.102	0.207	0.065	0.133	0.066	0.134	0.183	0.312	0.164	0.249	0.053	0.114	0.046	0.099	0.431	0.483	0.195	0.303	0.054	0.123
	Avg	**0.036**	**0.070**	0.076	0.171	0.055	0.117	0.052	0.110	0.099	0.203	0.152	0.235	0.045	0.104	0.038	0.087	0.365	0.434	0.183	0.291	0.045	0.108
Exchange	12.5%	**0.004**	**0.038**	0.043	0.151	0.040	0.137	0.023	0.106	0.112	0.224	0.235	0.407	0.345	0.483	0.280	0.411	0.017	0.090	0.025	0.112	0.019	0.098
	25%	**0.004**	**0.039**	0.104	0.240	0.043	0.143	0.037	0.134	0.184	0.288	0.412	0.551	0.443	0.561	0.253	0.383	0.030	0.122	0.037	0.136	0.034	0.130
	37.5%	**0.005**	**0.042**	0.149	0.296	0.045	0.148	0.053	0.159	0.281	0.355	0.447	0.579	0.393	0.522	0.316	0.472	0.045	0.150	0.050	0.159	0.046	0.153
	50%	**0.005**	**0.046**	0.215	0.363	0.039	0.140	0.071	0.185	0.387	0.431	0.499	0.593	0.575	0.649	0.358	0.489	0.036	0.130	0.061	0.176	0.062	0.176
	Avg	**0.005**	**0.041**	0.128	0.262	0.042	0.142	0.046	0.146	0.241	0.324	0.398	0.532	0.439	0.554	0.302	0.439	0.032	0.123	0.043	0.146	0.040	0.139
第一名次数		25	32	0	0	0	0	0	0	3	1	0	0	0	0	7	2	0	0	0	0	0	0

B.3 插补

表 11 展示了不同缺失率下的插补结果 $r \in \{12.5\%、25\%37。5\%，50\%\}$。遵循 TimesNet 中的实验协议（Wu 等人，2023b），我们随机遮蔽测试序列中的值，并使用均方误差（MSE）和平均绝对误差（MAE）指标评估重建质量。

B.4 异常检测

表12 报告了在五个基准数据集上的异常检测结果，采用精确率（P）、召回率（R）和F1分数（F1）指标。遵循标准做法（Xu等，2021），我们采用点调整协议：若异常片段内的任意一点被识别，则该检测被视为正确。

表*12.* 异常检测任务的完整结果。P、R和F1分别表示精确率、召回率和F1分数（%）。F1分数是精确率与召回率的调和平均数。P、R和F1的值越高，表示性能越好。

模型	SMD			MSL			SMAP			SWaT			PSM			平均F1分数
指标	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
长短期记忆网络（1997年） 1997b	78.52	65.47	71.41	78.04	86.22	81.93	91.06	57.49	70.48	78.06	91.72	84.34	69.24	99.53	81.67	77.97
Transformer 2017	83.58	76.13	79.56	71.57	87.37	78.68	89.37	57.12	69.70	68.84	96.53	80.37	62.75	96.56	76.07	76.88
LogTrans 2019	83.46	70.13	76.21	73.05	87.37	79.57	89.15	57.59	69.97	68.67	97.32	80.52	63.06	98.00	76.74	76.60
时间卷积网络 2019	84.06	79.07	81.49	75.11	82.44	78.60	86.90	59.23	70.45	76.59	95.71	85.09	54.59	99.77	70.57	77.24
Reformer 2020	82.58	69.24	75.32	85.51	83.31	84.40	90.91	57.44	70.40	72.50	96.53	82.80	59.93	95.38	73.61	77.31
Informer 2021	86.60	77.23	81.65	81.77	86.48	84.06	90.11	57.13	69.92	70.29	96.75	81.43	64.27	96.33	77.10	78.83
Autoformer2021 88.06	82.35	85.11	77.27	80.92	79.05	90.40	58.62	71.12	89.85	95.81	92.74	99.08	88.15	93.29	84.26	
Pyraformer 2022b	85.61	80.61	83.04	83.81	85.93	84.86	92.54	57.71	71.09	87.92	96.00	91.78	71.67	96.02	82.08	82.57
AnomalyTrans. 2021	88.91	82.23	85.49	79.61	87.37	83.31	91.85	58.11	71.18	72.51	97.32	83.10	68.35	94.72	79.40	80.50
Stationary 2022c	88.33	81.21	84.62	68.55	89.14	77.50	89.37	59.02	71.09	68.03	96.75	79.88	97.82	96.76	97.29	82.08
LSSL 2022	78.51	65.32	71.31	77.55	88.18	82.53	89.43	53.43	66.90	79.05	93.72	85.76	66.02	92.93	77.20	76.74
ETSformer 2022	87.44	79.23	83.13	85.13	84.93	85.03	92.25	55.75	69.50	90.02	80.36	84.91	99.31	85.28	91.76	82.87
DLinear 2023	83.62	71.52	77.10	84.34	85.42	84.88	92.32	55.41	69.26	80.91	95.30	87.52	98.28	89.26	93.55	82.46
TiDE 2023	76.00	63.00	68.91	84.00	60.00	70.18	88.00	50.00	64.00	98.00	63.00	76.73	93.00	92.00	92.50	74.46
PatchTST 2023b	87.26	82.14	84.62	88.34	70.96	78.70	90.64	55.46	68.82	91.10	80.94	85.72	98.84	93.47	96.08	82.79
iTransformer 2024b	78.45	65.10	71.15	86.15	62.65	72.54	90.67	52.96	66.87	99.96	65.55	79.18	95.65	94.69	95.17	76.98
MAS4TS（我们的方法）	86.76	83.51	85.10	90.65	74.96	82.06	89.87	55.44	68.58	91.61	94.74	93.14	91.87	98.44	95.04	**84.78**

C 工具库与自适应选择

本附录介绍了工具库的实现细节 $\mathcal{T}$，该工具库在第 3.4 节中进行了介绍。我们将描述统一调用接口、基本构建模块、特定任务的工具架构以及自适应路由机制。具体而言，我们将阐明：（i）路由器所考虑的 候选链空间；（ii）推理阶段的 搜索/选择过程；以及（iii）用于学习路由策略的 训练目标，从而确保工具驱动执行过程能够被复现和审计。

C.1 统一工具接口

MAS4TS 中的所有工具都提供标准化的调用接口，与正文中定义的路由策略 π_Θ 兼容。路由器会评估候选工具链，并返回置信度最高的选择。例如，若周期性信号较强，它可能会选择[TimesBlock、FFT] 而非 [PatchEmbed、Transformer]；若存在稀疏锚点，则可能会在工具链前添加 [ODE重构]。

语义自适应路由。 路由器维护一个可学习策略网络 g_ψ，根据三个输入为每条候选工具链评分：任务类型 τ、推理智能体提取的结构化先验 $\mathcal{P}$（例如检测到的趋势、锚点密度和周期性强度），以及共享记忆状态 $\mathbf{M}$的压缩摘要。

候选工具链空间。 对于每项任务 τ，我们维护一个候选工具链注册表 $\mathcal{C}_\tau = \{\text{Chain}_1、\ldots\ldots、\text{Chain}_{|\mathcal{C}_\tau|}\}$。每条工具链都是由若干工具模块组成的短序列，这些模块选自任务专用工具集（第 C.3–C.6 节）和少量共享预处理/后处理算子（例如归一化和掩码）。为保持路由效率并避免组合爆炸，我们将工具链长度限制在一个较小的整数范围内：

$$K \in [K_{\min}, K_{\max}], \tag{13}$$

使用 $K_{\min} = 1$ 和 $K_{\max} = 3$ 在我们的实验中。[1] 对于一项任务，我们不会 枚举所有可能的组合；相反，$\mathcal{C}_\tau$ 由一组经过精心筛选的有效且数值稳定的链组成，这些链符合模块输入输出模式（第 C.7节）。

推理时的搜索与选择。给定输入上下文（$\mathbf{X}$ $\mathcal{P}$，$\mathbf{M}$）和任务 τ，路由器会计算候选链的得分向量：

$$\mathbf{s} = g_\psi\big(\tau, \mathcal{P}, \text{Pool}(\mathbf{M})\big) \in \mathbb{R}^{|\mathcal{C}_\tau|}, \tag{14}$$

随后通过 softmax 得到 $\pi_\Theta(\text{Chain}_i \mid \tau,\ \mathcal{P},\ \mathbf{M})$（与正文中的公式 (9) 一致）。然后我们选择以下任一方式：(i) 选择 top-1 链（贪心），或 (ii) 选择一小组 top-k 集合进行集成：

$$\mathcal{I}_k = \text{TopK}(\mathbf{s}, k), \qquad k \in \{1, 2, 3\}. \tag{15}$$

除非另有说明，当最高置信度高于阈值（0.6）时，我们采用贪心 top-1 选择；低于该阈值时，则切换为 top-k 集成（附录 C.7）。

C.2 基础构建模块

在介绍特定任务工具之前，我们先介绍构成工具库的基本组件。这些模块可供不同任务共享，并能以不同配置组合使用。以下列出了一些关键组成部分，但并非全部。我们还将继续更新

[1]这些界限经过设定，既涵盖“归一化 → 骨干网络 → 头部”等常见模式，也允许包含一个或两个可选模块（例如分解、常微分方程重构）。

该工具库。

C.2.1 分块嵌入

分块嵌入模块将连续时间序列转换为离散词元，以便进行基于 Transformer 的处理：

$$\mathbf{P} = \text{Linear}(\text{Unfold}(\text{Pad}(\mathbf{X}))) + \mathbf{E}_{pos}, \tag{16}$$

其中，Unfold(·) 提取长度为 p、步幅为 s 的重叠分块，而 $\mathbf{E}_{pos}$ 表示可学习的位置嵌入。默认超参数为补丁长度= 16、步幅= 8和 $d = 64$。该模块能够捕捉逐点词元化无法表示的局部语义模式，从而显著提升时间模式学习效果。

C.2.2 多头自注意力

我们的注意力机制遵循标准的 Transformer 公式：

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{17}$$

通过多头扩展，该 将查询、键和值投影到 h 个子空间中，使模型能够关注来自不同表示空间的信息 s.

C.2.3 TIMESBLOCK

对于需要捕获周期模式的任务（例如分类），我们采用一个专用模块，该模块：(i) 使用快速傅里叶变换振幅分析检测前 k 个周期，(ii) 根据检测到的周期将一维序列重塑为二维表示，(iii) 应用具有多个卷积核尺寸的二维 Inception 风格卷积，以及 (iv) 使用学习得到的周期权重聚合结果：

$$\text{Period}(\mathbf{X}) = \text{FFT}^{-1}\left(\text{TopK}\left(|\text{FFT}(\mathbf{X})|, k\right)\right). \tag{18}$$

C.2.4 归一化策略

我们提供了多种针对非平稳时间序列优化的归一化策略：

• **RevIN** （可逆实例归一化）：对每个实例分别进行归一化，并保留统计数据以便反归一化，这对于在预测中保留原始尺度至关重要。

• **通道独立归一化**：分别对每个特征通道进行归一化，以避免产生虚假的跨通道相关性。
• **稳健归一化**：使用百分位数而非均值/标准差，从而在异常检测中能够抵御异常值的影响。

C.2.5 投影头

针对特定任务的投影头将潜在表示映射到输出空间：

• **FlattenHead**：展平补丁特征并投影至输出维度，同时支持通道独立模式和共享模式。
• **MLPHead**：两层多层感知机，激活函数可配置（ReLU、GELU）。
• **ResidualHead**：包含跳跃连接，以确保梯度流稳定。

C.3 预测工具

表13 总结了可用的预测工具。自适应路由器根据先验推断出的数据特征，在这些工具中进行选择 $\mathcal{P}$。

表13. 预测工具概览

Tool	架构	关键组件	选择条件
预测工具	Patch-Transformer	PatchEmbed + 编码器 + 展平头	默认；通用型
分解工具	趋势-季节	移动平均 + 投影	检测到明显趋势
NBeats模型	残差堆栈	基展开 + 双重残差	需要可解释性

C.3.1 补丁变换器预测工具

这是主要的预测工具，结合了分块嵌入与变换器编码：

算法 1 补丁变换器预测

要求： 输入 $\mathbf{X} \in \mathbb{R}^{B \times L \times D}$，预测时域 H

$\mathbf{X}_{norm}, \mu, \sigma \leftarrow \text{RevIN}(\mathbf{X})$ {实例归一化}
精确率 $\leftarrow \text{PatchEmbed}(\mathbf{X}_{norm})$ $\{\mathbb{R}^{B \cdot D \times N_p \times d}\}$
$\mathbf{H} \leftarrow \text{TransformerEncoder}(\mathbf{P})$ {多层注意力 + 前馈神经网络}
$\mathbf{Y}_{norm} \leftarrow$ 展平头$(\mathbf{H})$ {通道独立投影}
$\mathbf{Y} \leftarrow \text{RevIN}^{-1}(\mathbf{Y}_{norm}, \mu, \sigma)$ {反归一化}
返回 $\mathbf{Y} \in \mathbb{R}^{B \times H \times D}$

通道独立性: 每 个特征通道均独立处理，这在通道具有异质性时能提升鲁棒性 s
统计特性，并支持 高效的并行计算。

C.3.2 分解工具

检测到明显趋势（斜率 > 0.01）时，路由器会在前面添加分解：

$$\mathbf{X}_{\text{trend}} = \text{AvgPool}_k(\mathbf{X}), \quad \mathbf{X}_{\text{seasonal}} = \mathbf{X} - \mathbf{X}_{\text{trend}}, \tag{19}$$

其中， k 是移动平均核的大小。随后，主预测工具会处理季节性成分。

C.4 分类工具

表14. 分类工具概览

Tool	架构	关键组件	选择条件
分类工具	基于 TimesBlock	一维卷积 +快速傅里叶变换 周期 + 2D 卷积	默认；周期模式
时间卷积网络	空洞卷积	多尺度一维卷积 + 全局池化	短序列（$L \leq 32$）
InceptionTime	多分支卷积神经网络	Inception模块 + 残差连接	需要多尺度特征

C.4.1 TIMESBLOCK分类工具

主要的分类工具利用基于快速傅里叶变换的周期发现：

1. **数据嵌入**：使用一维卷积（而非线性层）捕捉局部模式，并结合位置编码。

2. **TimesBlock 堆栈**：多个模块通过快速傅里叶变换检测周期，将数据重塑为二维形式，应用 Inception 风格二维卷积，然后再重塑回原形。

3. **填充掩码**：对于变长序列至关重要——在展平之前将填充部分置零 .

4. **全序列投影**：展平 $[\text{seq len} \times d]$，并将其投影为类别逻辑值 .

C.4.2 时间卷积网络工具

对于较短的序列，路由器会选择采用扩张因果卷积的时间卷积网络：

$$(\mathbf{X} *_r f)(t) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{X}_{t - r \cdot i}, \tag{20}$$

其中， r 是随深度呈指数增长的膨胀因子，使模型能够通过 $O(\log L)$ 层高效捕获长程依赖。

表*15*。插补工具概述

Tool	架构	关键组件	选择条件
插补工具	Patch-Transformer	PatchEmbed + 编码器 + 重建	默认；通用
SAITS	自注意力	对角掩码注意力 + ORT	稀疏缺失模式
BRITS	双向循环神经网络	前向 + 后向 GRU + 衰减	连续缺失块

C.5 插补工具

C.5.1 PATCH-TRANSFORMER插补工具

调整基于patch的架构以进行重建：

算法 2 Patch-Transformer插补

要求： 输入 $\mathbf{X}$，包含缺失值；二值掩码 $\mathbf{M}$

$\mathbf{X}_{masked} \leftarrow \mathbf{X} \odot (1 - \mathbf{M})$ {将缺失位置置零}
$\mathbf{X}_{norm}$，μ，$\sigma \leftarrow$ 归一化$(\mathbf{X}_{masked})$
精确率 $\leftarrow$PatchEmbed$(\mathbf{X}_{norm})$
$\mathbf{H} \leftarrow$ TransformerEncoder(**精确率**)
$\hat{\mathbf{X}}_{norm} \leftarrow$ 重建头$(\mathbf{H})$ {与输入长度相同}
$\hat{\mathbf{X}} \leftarrow$ 反归一化$(\hat{\mathbf{X}}_{norm}, \mu, \sigma)$
$\mathbf{Y} \leftarrow \mathbf{X} \odot (1 - \mathbf{M}) + \hat{\mathbf{X}} \odot \mathbf{M}$ {保留已知值}
返回 Y

C.6 异常检测工具

表*16.* 异常检测工具概览

Tool	架构	关键组件	选择条件
异常检测工具	多尺度补丁	多尺度补丁嵌入 + 重建	默认；一般异常
VAE异常检测器	VAE	编码器 + 重参数化 + 解码器	关注点异常

C.6.1 多尺度异常检测工具

主要的异常检测工具采用多尺度重建：

1. **实例归一化**：使用可学习仿射参数保留异常的相对幅度（不同于会将其移除的 RevIN）。2. **多尺度处理**：细尺度（$p = 4$）用于点异常；粗尺度（$p = 16$）用于片段异常。

3. **多尺度融合**：拼接所有尺度的重建结果，并通过多层感知机进行融合。

异常分数结合重构误差与基于注意力的加权：

$$s_t = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2 \cdot \alpha_t, \tag{21}$$

其中，α_t 是用于突出显示可疑区域的注意力权重。

C.6.2 变分自编码器异常检测器

通过重构似然进行概率检测：

$$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi^2(\mathbf{x})), \tag{22}$$

$$s = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \beta \cdot D_{KL}(q_\phi \| p), \tag{23}$$

其中，异常分数由重构误差和KL散度组成。

C.7 工具链式调用

如第 3.4 节所述，MAS4TS 将多个工具组合成工具链，其中每个工具的输出都作为下一个工具的输入。工具链的正式定义如下：

$$\hat{\mathbf{Y}} = \mathcal{T}_{m_K} \circ \mathcal{T}_{m_{K-1}} \circ \cdots \circ \mathcal{T}_{m_1}(\mathbf{Z}), \tag{24}$$

w其中，$\circ$表示函数复合，$\{m_1, \ldots, m_K\}$是由路由策略 π_Θ选定的工具索引 .

输入/输出模式与有效性。 每个工具 $\mathcal{T}_m$ 都公开 (i) 一个输入模式（张量形状、掩码字段和可选的辅助上下文），以及 (ii) 一个与工具链中下一个工具兼容的输出模式。工具注册表会在组合时强制确保模式一致性；无效组合会从 $\mathcal{C}_\tau$ (第 C.1节)中排除。这种设计使工具链可审计，并能防止路由过程中出现未被察觉的形状/类型不匹配。

默认链。 表 17展示了路由置信度超过 0.6 时的默认工具链。方括号表示根据先验条件纳入的组件 $\mathcal{P}$。

表17。 按任务划分的默认工具链

Task	**工具链**
预测	RevIN $\rightarrow$ [分解] $\rightarrow$PatchTransformer $\rightarrow$ RevIN^{-1}
分类	归一化 $\rightarrow$ TimesBlock $\rightarrow$全局池化 $\rightarrow$ 分类头
插补	掩码编码 $\rightarrow$ 归一化 $\rightarrow$ PatchTransformer $\rightarrow$ 重建
异常检测	实例归一化 $\rightarrow$ 多尺度编码器 $\rightarrow$ 重建 $\rightarrow$ 分数计算

E集成链。 当路由置信度较低（< 0.6）时，可以对多个工具链进行集成 :

$$\hat{\mathbf{Y}} = \sum_{i=1}^{K} w_i \cdot \text{Chain}_i(\mathbf{Z}), \quad w_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)}, \tag{25}$$

其中，s_i 是置信度 分数，由 g_ψ 针对链 i计算得出。这在模糊或分布外输入情况下提供了稳健性 ts.

关于确定性的实践说明。为降低评估期间的方差，当 $c \geq 0.6$时，我们会关闭链式推理中的随机采样，并采用确定性的 top-1 选择；否则则采用确定性的 top-k 集成（第C.1节）。在随机种子固定的情况下，这能确保不同运行之间的路由行为保持稳定。

D 视觉语言模型提示模板

本附录记录了**推理器**智能体在视觉推理中使用的提示模板，相关内容见第3.3节。每个提示都经过设计，旨在引导视觉语言模型生成结构化且针对特定任务的输出，以便下游数值推理直接使用。

D.1 提示设计原则

我们的提示遵循三项关键原则：

1. **结构化输出**：所有提示均要求以预定义模式提供 JSON 格式响应，以确保确定性解析，并与下游模块无缝集成。

2. **统计锚定**：提供输入统计量（最小值、最大值、均值、标准差）和语义上下文，将视觉推理锚定于定量现实，从而减少超范围幻觉。

3. **任务特定锚点**：不同任务需要不同类型的锚点——预测锚点用于定义未来轨迹形状，异常锚点用于对比正常与异常模式，分类锚点用于捕捉判别性特征，而插补锚点则用于引导局部重构。

锚点数量范围 {最小锚点数}-{最大锚点数} 可由用户配置。根据经验，我们建议：预测（8–15 个）、异常检测（8–12 个）、分类（5–8 个）和插补（5–7 个）；预测时域越长，通常越适合使用更密集的锚点。

D.2 预测提示

系统提示（预测）

你是一位资深时间序列预测专家。请分析时间序列图，并以严格的 JSON 格式给出预测结果。请始终根据模式一致性提供置信度分数（0.00–1.00）。

用户提示模板（预测）

分析这张用于预测任务的时间序列图（预测 {pred len} 个未来步）。输入序列：{seq len} 个步（t=0 至 {seq len-1}），最后一个值={last value}历史统计数据：范围=[{history min}, {history max}]，标准差={history std}要求：

1. 仅输出预测窗口的 {min anchors}-{max anchors} 个关键锚点（峰值/谷值/拐点）。2. 锚点类型：'start'、'peak'、'valley'、'inflection'、'end'。3. 所有值必须处于合理范围内。4. 置信度分数应反映模式连续性。

输出 JSON：

{ "置信度": 0.85, "锚点": [{"t": 96, "v": 0.342, "类型": "起始"}, {"t": 120, "v": 0.456, "类型": "峰值"}, ...]}

预测提示明确将锚点限制在预测窗口（时间步 $t \geq L$）内，而不是概括历史模式。最后值字段确保预测边界处的连续性，而历史标准差有助于校准合理的波动幅度。随后，这些锚点会输入潜在常微分方程解码器，以进行稠密轨迹重建。

D.3 分类提示

系统提示（分类）

你是一位资深的时间序列分类专家。请分析时间序列图并对序列模式进行分类。请以严格的 JSON 格式提供置信度分数。

用户提示模板（分类）

分析这幅时间序列图以进行分类（输入长度={序列长度}）。输入统计信息：范围=[{历史最小值}, {历史最大值}], 均值={历史均值}要求：

1. 识别主导模式类型（周期性、趋势性、Stationary 等） 2. 关键锚点：{最小值锚点}-{最大值锚点} 个确定类别特征的点 3. 置信度分数反映模式的独特性

输出 JSON：

{ "confidence": 0.78, "pattern_type": "周期性", "key_anchors": [{"t": 12, "v": 0.89, "type": "周期_峰值"}, {"t": 36, "v": -0.45, "type": " 周期 _ 谷值 "}]}

该提示侧重于刻画不同类别的判别性时间特征。pattern type 字段提供高层语义指导，可与学习得到的数值特征融合。

D.4 插补提示

系统提示（插补）

你是一位资深的时间序列插补专家。请分析时间序列图，并对输入序列中的缺失值进行插补。请提供包含置信度分数的 JSON。

用户提示模板（插补）

分析这张用于插补的时间序列图（输入长度={序列长度}）。缺失值：标记在位置 {缺失位置}
输入统计信息：范围=[{历史最小值}, {历史最大值}], 均值={历史均值}要求：

1. 仅插补标记的缺失位置 2. 说明推理方式：“插值”或“外推” 3. 关键锚点：{最小值锚点}-{最大值锚点} 用于指导插补的关键点

输出 JSON：

{ "置信度": 0.88, "插补_值": [{"t": 25, "v": 0.34, "原因": "插值"},],"关键_锚点": [{"t": 20, "v": 0.28, "类型": "观测值"},]}

插补提示会明确标记缺失位置，使视觉语言模型将注意力集中在相关缺口上。原因字段用于区分插值（两侧均有观测值作为界限）和外推（基于单侧上下文延伸），从而为潜在常微分方程提供重建置信度信息。观测边界处的关键锚点提供硬约束，确保结果与已知数据点保持一致。

D.5 异常检测提示

系统提示（异常检测）

您是一位资深的时间序列异常检测专家。请分析时间序列图，并找出输入序列中的异常时间步。请提供包含置信度分数的严格 JSON。

用户提示模板（异常检测）

分析此时间序列图以进行异常检测（输入长度={序列长度}）。
输入统计信息：范围=[{历史最小值}, {历史最大值}], 均值={历史均值}, 标准差={历史标准差}
要求：

1. 仅关注输入序列（t=0 至 {序列长度-1}）
2. 异常分数：0.0-1.0（数值越高 = 越异常）
3. 对于连续异常，仅报告其中最显著的一个
4. 关键锚点：{最小锚点}-{最大锚点} 个点，包括正常模式作为参考

输出 JSON：

```
{
  "置信度": 0.82,
  "异常_评分": [
    {"t": 45, "评分": 0.91, "原因": "尖峰"},
    {"t": 78, "score": 0.85, "reason": "level_shift"}
  ],
  "key_锚点": [
    {"t": 10, "v": 0.23, "type": "normal"},
    {"t": 45, "v": 1.82, "type": "异常"}
  ]
}
```

异常检测提示要求同时提供异常分数和参考锚点。这种双输出结构使基于重建的检测流程能够区分异常模式与正常时间动态。原因字段（尖峰、水平偏移等）为下游分析提供了可解释性。对于连续出现的异常，仅要求指出其中最显著的一项，以避免冗余检测。

E 多智能体系统的合理性

尽管预测、插补、异常检测和分类在输出与目标上各不相同，但它们共享一个通用工作流程：(i) **感知** 显著的时间结构，(ii) **推理** 全局形状约束和局部不规则性，(iii) **执行** 决策，同时满足有效性约束（例如与观测值一致、有界且平滑）。这与智能体视角自然契合，并促使我们将时间序列分析建模为分析—推理—执行流程。

单一的整体模型通常会将这些步骤纠缠在一个带有固定头部的潜在表示中，因此难以 (a) 根据异质时间序列的特征和任务要求调整执行路径，以及 (b) 以可审计、可控的方式整合领域工具（例如分解、稳健统计、约束投影）。多智能体设计提供了一种有原则的方法，可将工作流模块化为不同的专门角色，同时通过受控的共享内存接口实现协作。具体而言，MAS4TS 遵循**分析器–推理器–执行器（ARE）**范式：分析器生成有依据的统计先验和基于图表的接口，推理器将感知到的结构转化为明确的锚点和轨迹约束，而执行器则选择、组合并验证特定任务工具链。

这种分工支持自适应计算（针对不同情形采用不同的工具组合），提高了可解释性（锚点、约束和验证器检查均可检查），并通过将学习重点放在协调关键组件（例如路由和记忆门控）上，而非重新训练特定任务的骨干网络，从而减轻训练负担。从这个意义上说，多智能体系统的优势不仅在于并行化，更在于能够通过统一的推理与执行协议，并借助明确的中间产物，整合多种时间序列任务。

F 局限性与未来工作

MAS4TS 继承了其底层基础模型（尤其是用于锚点提取的视觉语言模型）的能力与失效模式，这可能会影响其在分布偏移下的鲁棒性。此外，在复杂的多变量和长上下文场景中，智能体工作流可能会引入冗余上下文和通信开销，从而降低工具驱动执行带来的效率提升。

我们将继续扩展 MAS4TS，使其成为不断演进的时间序列智能体系统，具体包括：(i) 丰富工具库，并改进工具路由，以应对更严苛的现实约束；(ii) 探索智能体**技能**和基础模型推理策略，以便以更少的冗余、更强的泛化能力更好地处理复杂上下文。