计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(7): 1869−1883, 2026

# 面向少样本学习的增强式跨模态微调框架

丁琳琳　李佳林　卢　晶　李　默　王媛娜
（辽宁大学信息学部　沈阳　110036）
（dinglinlin@lnu.edu.cn）

# Enhanced Cross-Modal Fine-Tuning Framework for Few-Shot Learning

Ding Linlin, Li Jialin, Lu Jing, Li Mo, and Wang Aina
（*Faculty of Information*, *Liaoning University*, *Shenyang* 110036）

**Abstract**　In recent years, large language model (LLM) has shown potential in multivariate time series forecasting. However, existing cross-modal methods based on LLM still face the challenges of insufficient modality alignment and limited generalization capability in few-shot learning and long-term forecasting tasks. To address the above issues, we propose an enhanced cross-modal LLM fine-tuning framework, termed Enhanced CALF, which integrates dynamic attention and hierarchical distillation. Firstly, the framework constructs a dynamic attention cross-modal matching module, introducing adaptive weight generation and alignment prediction mechanisms. This module can dynamically adjust the allocation of attention weights based on data distribution characteristics and inter-modality correlation strengths, thereby improving the precision of cross-modal feature alignment. Secondly, a multi-level knowledge distillation and contrastive learning module is built. By introducing projection mappings at each Transformer layer and combining hierarchical distillation loss with an adaptive temperature contrastive loss, it achieves hierarchical feature transmission covering aspects from local details to global semantics, aiming to enhance the consistency of cross-modal representations. Finally, an adaptive alignment mechanism is designed, which dynamically adjusts the weights of the total loss function through quantitative evaluation of inter-modality alignment scores, thereby optimizing the model training process. Experiments on seven real-world datasets demonstrate that Enhanced CALF outperforms existing baseline models in long-term forecasting and few-shot learning tasks.

**Key words**　large language model (LLM); time series forecasting; knowledge distillation; contrastive learning; attention mechanism

**收稿日期**：2025-10-21；**修回日期**：2026-03-04

**基金项目**：国家科技重大专项课题（2024ZD1700104）；国家自然科学基金项目（52574191, 62072220）；“兴辽英才”青年拔尖项目（XLYC2203003）；辽宁省教育厅科研平台建设项目（LJ232510140001）；辽宁省自然科学基金项目（2022-KF-13-06）；辽宁省教育厅青年项目（JYTQN2023189）；辽宁大学自然科学基金项目（LDZDJC2402）；辽宁省科技计划联合计划项目（a272500011, 2025-BSLH-155, 2025-BSLH-159）；辽宁大学科研启动项目（a282500028）；中央引导地方科技发展资金项目（2026JH6/101100002）；辽宁省高校基本科研项目创新发展项目（LJ242510140015）

This work was supported by the National Science and Technology Major Special Project (2024ZD1700104), the National Natural Science Foundation of China (52574191, 62072220), the Youth Top Notch Project of “Xingliao Talents” (XLYC2203003), the Basic Science Research Project of Liaoning Provincial Department of Education (LJ232510140001), the Natural Science Foundation of Liaoning Province (2022-KF-13-06), the Youth Project of Liaoning Provincial Department of Education (JYTQN2023189), the Natural Science Foundation of Liaoning University (LDZDJC2402), the Liaoning Province Science and Technology Joint Plan Project (a272500011, 2025-BSLH-155, 2025-BSLH-159), the Liaoning University Research Launch Project (a282500028), the Central Government Guides Local Science and Technology Development Fund Project (2026JH6/101100002), and the Liaoning Provincial University Basic Scientific Research Project Innovation and Development Project (LJ242510140015).

**通信作者**：卢晶（lujing@lnu.edu.cn）

**摘 要** 近年来,大语言模型(large language model, LLM)在多元时间序列预测中展现出潜力,但现有基于大语言模型的跨模态方法在少样本学习和长期预测中仍面临模态对齐不充分与泛化能力受限的挑战。为解决上述问题,提出一种集成动态注意力与层次蒸馏的增强式跨模态微调框架(enhanced cross-modal LLM fine-tuning, Enhanced CALF)。该框架首先构建动态注意力跨模态匹配模块,引入自适应权重生成与对齐度预测机制,能够依据数据分布特性与模态间关联强度,动态调整注意力权重分配,以提升跨模态特征对齐的精准度;其次,构建多层次知识蒸馏与对比学习模块,通过在 Transformer 各层级引入投影映射,并结合层级蒸馏损失与自适应温度对比损失,实现层次化的特征传递,涵盖从局部细节到整体语义的各个层面,以提升跨模态表示的一致性;最后,设计自适应对齐机制,通过模态间对齐度分数的量化评估,动态调整总损失函数的权重,从而优化模型训练过程。在 7 个真实数据集上的实验表明,Enhanced CALF 在长期预测与少样本学习任务中优于现有基线模型。

**关键词** 大语言模型;时间序列预测;知识蒸馏;对比学习;注意力机制

**中图法分类号** TP18; TP389.1

**DOI:** 10.7544/issn1000-1239.202550755 **CSTR:** 32373.14.issn1000-1239.202550755

多元时间序列预测(multivariate time series forecasting, MTSF)作为时间序列分析领域的核心任务,凭借挖掘复杂变量间动态关系的能力,在气象预测、交通、金融建模等多个领域得到广泛应用。例如,在气象预测中,通过分析温度、湿度、气压等多维数据,可实现对极端天气事件的精准预测。在交通预测中,结合交通报告文本与车流量时间序列,大语言模型(large language model, LLM)可捕捉路况变化和交通事件影响,提升有限数据条件下的预测性能。在金融市场领域,基于股票价格、交易量、宏观经济指标等序列预测,助力投资者制定科学的投资策略。

近年来,随着大语言模型在自然语言处理领域取得突破性进展,基于大语言模型的跨模态方法逐渐被引入时间序列预测任务,该方法通过创新性地融合文本和时间序列信息,有效地弥补了传统模型在数据表征方面的不足。例如,在金融预测任务中,通过融合财经新闻文本与股票价格时间序列数据,大语言模型能够利用其强大的语义理解能力捕捉市场情绪与政策导向,从而在有限数据条件下提升预测性能。

已有研究提出了基于跨模态大语言模型微调的CALF(cross-modal LLM fine-tuning)框架,旨在通过融合文本与时间序列模态提升预测性能[1]。然而,尽管基于LLM的跨模态方法展现出巨大潜力,现有研究仍面临以下关键挑战:1)文本与时间 token 的分布差异导致模态对齐不充分。自然语言文本具有离散、语义丰富的特点,而时间序列数据呈现连续、数值化特征,二者在数据结构和语义表达上存在本质差异。这种差异使得跨模态融合时,难以建立有效的对齐机制,容易造成信息损失或冗余。2)在少样本学习和长期依赖建模场景下,该方法的泛化能力受限。少样本学习场景下,由于缺乏充足的训练数据,模型难以学习到时间序列的内在规律;而在长期依赖建模中,随着预测时间跨度的增加,模型对历史信息的记忆和利用能力不足,导致预测精度下降。

针对上述挑战,本文提出了一种集成动态注意力与层次蒸馏的增强式跨模态微调框架(enhanced cross-modal LLM fine-tuning, Enhanced CALF)。Enhanced CALF 框架首先构建动态注意力机制,并设计包含门控权重生成单元的跨模态匹配模块,该模块可依据数据分布特性实时计算文本与时间序列模态间的相似度矩阵。当检测到时间序列周期性模式与文本语义存在强关联时,自动提升交叉注意力权重以强化模态交互;若模态内特征一致性较高,则增强自注意力以挖掘局部依赖关系,从而动态调整注意力权重。之后搭建多层次蒸馏框架,在 Transformer 编码器的每一层级部署投影映射模块,通过多层级特征正则化损失和对比损失的联合优化,实现从低级时序特征到高级语义特征的全层级知识迁移。以此在特征表示的不同层级进行知识迁移,通过分层特征对齐减少语义漂移。最后,引入自适应对齐机制,通过量化评估模态间匹配质量来优化损失函数。当对齐度分数表明跨模态特征匹配程度高时,适当降低总损失函数的权重;当匹配程度低时,增加权重,从而增强模型对分布差异的鲁棒性。

本文的主要贡献有 4 个方面:

1)针对文本与时间序列模态间分布差异导致的跨模态特征对齐精准度不足问题,本文构建了动态

注意力跨模态匹配模块，设计了一种基于动态门控的对齐架构。该架构引入自适应权重生成机制，能够依据数据分布特性与模态间关联强度，动态调节注意力权重分配，从而在特征交互过程中实现更精准、更具适应性的模态对齐，有效提升了模型对复杂时序模式与文本语义的匹配精度。

2）针对现有知识蒸馏框架仅能在单一层级进行特征迁移的局限性，构建了多层次知识蒸馏与对比学习模块。该模块通过在 Transformer 各层级部署投影映射操作，并引入层级蒸馏损失与自适应温度对比损失，实现了从低级时序特征到高级语义特征的全层级知识迁移，有效减少语义漂移现象。

3）针对传统损失函数对模态匹配质量变化不敏感、损失权重固定导致的训练适配性不足的问题，本文提出了自适应对齐机制。该机制通过对模态间对齐质量进行量化评估，生成对齐度分数，并据此动态调整总损失函数中各部分的权重分配，使模型能够根据不同模态匹配场景下的分布差异，自适应优化训练过程，增强了模型对异构模态数据的鲁棒性与训练稳定性。

4）在长期预测与少样本学习任务中开展全面实验。结果表明，在长期预测任务中，Enhanced CALF 模型在多个数据集上表现突出，平均 *MSE* 相对下降 7.38%，*MAE* 相对下降 6.10%。在少样本场景下，即便仅使用 5% 的训练数据，该模型依然优于基线模型：例如在 ETTm1 和 ETTh2 数据集中，*MSE* 分别相对降低 6.64% 和 8.74%，*MAE* 分别降低 6.09% 和 5.43%。随着训练数据比例提升至 15%，模型性能得到进一步改善，在 ETTm1 数据集上实现了 *MSE* 相对降低 9.31%、*MAE* 相对降低 8.85%。以上实验结果充分验证了该框架在预测任务中的有效性与泛化能力。

## 1 相关工作

### 1.1 时间序列预测

近年来，深度学习在时间序列预测领域引发了重大变革，一系列致力于提高预测精度的新模型相继出现。其中，Transformer 架构凭借其对时序中复杂关联关系的强大建模能力，已成为该领域的主流方法，在多项预测任务中表现出优异性能。Zhou 等人[2]较早探索了大语言模型在该领域的应用潜力，所提出的框架具有开创性，为此后的研究确立了重要方向。受此影响，后续工作从不同角度对该范式进行了拓展与优化。例如，Chang 等人[3]设计了一种 2 阶段微调方法，将时序片段与额外的时间编码融合后输入预训练大语言模型，以提升对时间维度的识别与建模效果。针对非平稳时序的建模难题，Cao 等人[4]则引入了时序分解与选择性提示相结合的机制，通过分解处理数据非平稳性，并借助可选择的提示动态调整模型，从而增强其适应能力。然而，现有方法通常直接向大语言模型输入原始时序数据，忽略了时序与文本模态在表示结构上的本质区别，这种直接迁移导致次优性能。为此，近期研究开始提出更具针对性的对齐方案。Sun 等人[5]的工作通过对比学习对齐时间序列与大语言模型的嵌入表示，同时集成软提示技术，以增强预训练大语言模型直接理解与处理时序模式的能力。Jin 等人[6]进一步提出了基于文本原型重编程与上下文前缀增强的对齐框架，通过文本原型转换时序输入，并加入动态生成的前缀信息，以更好地适配大语言模型的推理模式。此外，将多元时间序列中的变量依赖建模为图结构，并借助图神经网络进行学习与预测，也是重要的研究方向。其中，注意力机制常用于增强节点间的信息聚合。例如，Ding 等人[7]提出了自适应锚点注意力网络，用于建模大规模稀疏二部图，其自适应注意力机制能有效捕捉稀疏连接中的关键关系，为复杂结构数据的依赖建模提供了新工具。这类方法为多元时间序列的关系视角建模提供了互补路径。

### 1.2 少样本学习

少样本学习致力于在标注数据有限的情况下，实现对未知类别对象的准确辨识，已成为应对数据稀缺挑战的重要研究方向。主要方法可归纳为 2 个方向：元学习方法聚焦于设计能够快速适应新任务的学习框架或参数优化机制[8]；度量学习方法则着力于构建判别性更强的特征嵌入空间[9]。尽管少样本目标检测等延伸任务与少样本学习共享部分核心的方法理念，但后者相对完善的理论体系持续为相关领域提供支撑。以文献 [10] 的方法为例，该方法通过迁移已有知识改进区域建议网络的鲁棒性，从而抑制样本不足导致的模型性能下降；文献 [11] 则借助子空间学习对多视角特征实施降噪处理，以此协调不同视角间的表征差异，达到缩小类间距离的效果；而文献 [12] 提出的类别分布校准策略，通过将基类特征分布对齐至相似的新类，进一步增强了模型的跨类别泛化能力。

### 1.3 知识蒸馏

知识蒸馏（knowledge distillation, KD）[13]作为一种模型压缩技术，基本原理是利用大规模预训练的教

师模型监督一个轻量级学生模型的训练过程。该方法借助输出分布的近似优化，促使教师网络指导学生网络充分训练，因而成为实现大规模深度网络高效迁移与部署的重要技术[14]。作为大规模深度网络轻量化部署与应用的关键技术之一，知识蒸馏在基本保持模型性能的同时，降低了计算与存储资源的需求，有力推动了高性能模型在资源受限的边缘设备上的落地应用[15]。

近年来，为了实现教师模型知识向学生模型的高效传递，已发展出多种蒸馏机制，主要包括基于输出逻辑值的对齐[16]、中间层特征的模仿[17]以及样本间关系的一致化[18]。例如：CLIP-KD[19]研究发现，在对CLIP模型进行蒸馏时，使用简单的均方误差(MSE)损失进行特征模仿即可达到最优效果；TinyCLIP[20]则尝试在教师与学生模型的仿射变换空间内实现跨模态特征对齐。

目前的知识蒸馏方法在各类任务中均表现出良好效果。例如，王双丞等人[21]提出基于跨模块知识蒸馏的3阶段少样本目标检测方法，通过特征分布校准、初始偏差抑制和跨模块蒸馏策略，有效改善了少样本目标检测中新类样本的特征失真与分布偏移，从而提升了检测性能与稳定性。此外，多层蒸馏[22]、跨层蒸馏[23]等策略也被广泛应用于缓解新类和基类间的语义差异问题，提升模型在少样本任务中的泛化能力。

### 1.4 对比学习

在自监督学习中，对比学习通过构造正负样本对来实现判别性表征的学习，已成为该领域的主要范式之一，并有力推动了表征学习的发展。SimCLR[24]与MoCo[25]作为早期经典工作，分别通过数据增强和动量编码器等技术，验证了无监督对比学习在特征提取上的潜力。而在跨模态学习方面，CLIP和ALIGN[26]借助海量图像-文本对进行预训练，建立了能够对齐多模态信息的共享嵌入空间，为后续图文融合任务提供了重要基础。针对跨模态实体链接任务中模态间相似实体不易区分的挑战，王苑铮等人[27]进一步提出自对比学习与难负例对比学习策略，改善了不同模态间相似实体的区分难题。此外，对比学习方法在知识超图谱链接预测中也得到应用，Ding等人[28]提出了一种内外诱导嵌入方法(HyIE)，通过对比学习机制增强超图谱中实体与关系的表示对齐，进一步验证了对比学习在复杂结构化数据建模中的有效性。

综合现有研究，基于大语言模型的跨模态时间序列预测方法虽展现出潜力，但仍面临关键挑战。具体表现为模态对齐多依赖静态机制，难以依据数据分布与模态关联动态调节注意力权重，导致特征匹配不精准；知识蒸馏与特征对齐常局限于单一层级，阻碍了从局部时序模式到全局语义信息的全尺度知识迁移，易造成语义漂移；训练过程中损失函数权重固定，缺乏对模态匹配质量的动态评估与自适应调整，制约了模型在少样本及长期预测任务中的鲁棒性。所以本文提出Enhanced CALF框架来解决上述问题。

## 2 基于大语言模型的增强式跨模态微调框架

本文提出一种集成动态注意力与层次蒸馏的增强式跨模态微调框架——Enhanced CALF，解决了基于大语言模型的时间序列预测中少样本学习和长期依赖建模难题。

Enhanced CALF框架如图1所示。该模型采用双分支架构，包括文本源分支和时间目标分支。文本源分支首先对输入文本进行词嵌入处理，从预训练词向量库中获取词向量表示，再经主嵌入层增强语义编码，并引入位置编码注入序列顺序信息。处理后的特征输入至多层Transformer Encoder中进行深层特征提取，最终由输出头生成文本模态表示。时间目标分支将时间序列通过Embedding层映射为稠密向量，同样添加位置编码以保持时序结构信息，随后该向量序列输入至多层Transformer编码器中提取时序特征，最终通过输出头获得时序模态表示。核心模块中，层次化知识蒸馏模块接收双分支特征，通过投影层将二者映射至同一表示空间，并经全局平均池化获取各层特征表示，进而计算层次化蒸馏损失以促进跨模态特征在不同层级上的一致性。在跨模态注意力交互阶段，文本分支输出作为键($\boldsymbol{K}$)与值($\boldsymbol{V}$)，时间分支输出作为查询($\boldsymbol{Q}$)，通过跨模态多头注意力和多头注意力实现2模态间的特征融合。之后动态权重生成器依据注意力交互后的特征自适应生成权重系数，以精细化调控模态对齐过程。

### 2.1 层次化知识蒸馏与对比学习模块

该模块的主要作用是对齐时间序列和文本这2种不同模态的表示，使它们在多个层级上保持一致，提升模型跨模态表示能力。此模块算法为附录A的算法A1。需要特别指出的是，传统的单一层级对齐方法无法处理这种多尺度特征的复杂性，要么只能捕捉局部细节而忽略宏观趋势，要么只能把握全局

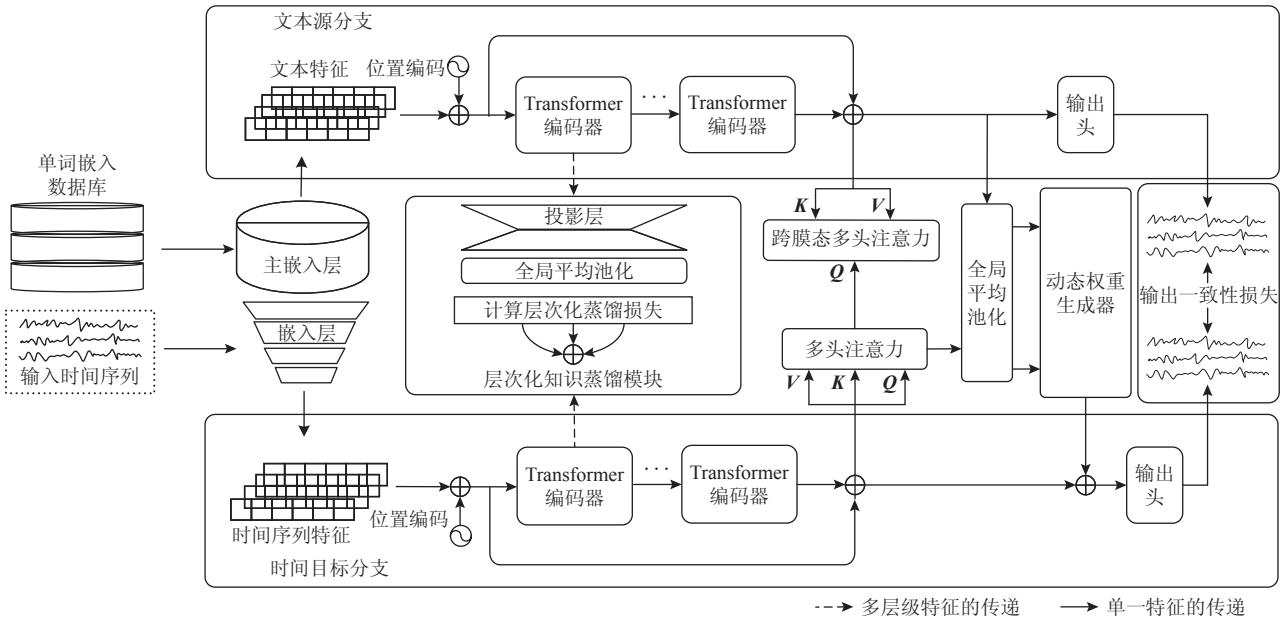



Fig. 1 Enhanced CALF model architecture

图 1 Enhanced CALF 模型架构

轮廓而丢失微观动态。本文提出的层次化知识蒸馏框架，能够协调这种潜在的矛盾。它并不假定所有层级的特征都必须完全一致，而是通过分层级的、加权的方式实现知识迁移。底层网络专注于捕捉短期波动等局部特征，并与文本中的瞬时性描述建立对齐；高层网络则擅长提取长期趋势等全局特征，并与文本中的概括性语义进行匹配，表 1 为关键参数说明。

**Table 1 Key Parameter Description**

**表 1 关键参数说明**

| 符号 | 含义 |
|---|---|
| $\boldsymbol{X}_{\text{time}}, \boldsymbol{X}_{\text{text}}$ | 时间序列和文本特征 |
| $L_{\text{time}}, L_{\text{text}}$ | 时间序列和文本长度 |
| $B$ | 批大小 |
| $\tau_i$ | 层级自适应温度 |
| $\sigma$ | softmax/sigmoid 函数 |
| $\lambda$ | 可学习损失权重 |

通过这种分层解耦的对齐策略，模型能够学习到从局部到全局的跨模态表示谱系。最终的层级加权融合模块则负责动态地整合这些不同尺度的对齐信息，根据具体预测任务的需求平衡局部细节与全局趋势的贡献，从而在面对局部与全局模式矛盾时作出更精准的决策。

层次化知识蒸馏与对比学习模块对来自文本分支和时间分支的不同层级的特征进行系统性的对齐与融合。输入为文本分支的层级特征列表$\{\boldsymbol{X}_{\text{text}}^{(1)}, \boldsymbol{X}_{\text{text}}^{(2)}, \cdots, \boldsymbol{X}_{\text{text}}^{(N)}\}$和时间分支的层级特征列表$\{\boldsymbol{X}_{\text{time}}^{(1)}, \boldsymbol{X}_{\text{time}}^{(2)}, \cdots, \boldsymbol{X}_{\text{time}}^{(N)}\}$，其中，$N$为 Transformer 层数，$\boldsymbol{X}^{(i)} \in \mathbb{R}^{L\times B\times M}$，$L$为序列长度，$B$为批大小，$M$为隐藏层维度。模块通过以下 4 个子操作，最终输出一个融合了多尺度跨模态信息的增强表示$\boldsymbol{F} \in \mathbb{R}^{B\times M}$。

1）层级投影函数

层次化知识蒸馏与对比学习模块通过层级投影函数实现跨模态特征的空间对齐，该函数以第 $i$ 层的原始时间特征$\boldsymbol{X}_{\text{time}}^{(i)}$和文本特征$\boldsymbol{X}_{\text{text}}^{(i)}$作为输入，通过独立的线性变换层$\boldsymbol{W}_{\text{time}}^{(i)}$，$\boldsymbol{W}_{\text{text}}^{(i)}$和偏置向量$\boldsymbol{b}_{\text{time}}^{(i)}$，$\boldsymbol{b}_{\text{text}}^{(i)}$将其映射至统一的表示空间，最终输出投影后的时间特征$\hat{\boldsymbol{T}}^{(i)}$和文本特征$\hat{\boldsymbol{X}}^{(i)}$，为后续层级对齐操作建立统一的特征基础。公式为

$$\hat{\boldsymbol{T}}^{(i)} = \boldsymbol{W}_{\text{time}}^{(i)} \cdot \boldsymbol{X}_{\text{time}}^{(i)} + \boldsymbol{b}_{\text{time}}^{(i)}, \tag{1}$$

$$\hat{\boldsymbol{X}}^{(i)} = \boldsymbol{W}_{\text{text}}^{(i)} \cdot \boldsymbol{X}_{\text{text}}^{(i)} + \boldsymbol{b}_{\text{text}}^{(i)}, \tag{2}$$

其中$i$表示层级索引。

2）层级蒸馏损失

层级蒸馏损失以前一层输出的投影特征$\hat{\boldsymbol{T}}^{(i)}$和$\hat{\boldsymbol{X}}^{(i)}$作为输入，通过计算跨模态特征分布差异来约束层级对齐效果。

首先对投影后的时间特征和文本特征分别沿序列维度进行平均池化，得到层级全局表示：

$$\bar{\boldsymbol{T}}^{(i)} = \frac{1}{L_{\text{time}}} \sum_{j=1}^{L_{\text{time}}} \hat{\boldsymbol{T}}_j^{(i)} \text{ 和 } \bar{\boldsymbol{X}}^{(i)} = \frac{1}{L_{\text{text}}} \sum_{k=1}^{L_{\text{text}}} \hat{\boldsymbol{X}}_k^{(i)}, \tag{3}$$

其中$i$表示层级索引。$\hat{\boldsymbol{T}}_j^{(i)}$是第$i$层级中时间特征经过

投影后的第$j$个元素；$\hat{\boldsymbol{X}}_k^{(i)}$是第$i$层级中文本特征经过投影后的第$k$个元素。随后通过计算2个全局表示之间的均方误差得到该层级的蒸馏损失。公式为

$$L_{\text{distill}}^{(i)}=\frac{1}{d}\left\|\frac{1}{L_{\text{time}}}\sum_{j=1}^{L_{\text{time}}}\hat{\boldsymbol{T}}_j^{(i)}-\frac{1}{L_{\text{text}}}\sum_{k=1}^{L_{\text{text}}}\hat{\boldsymbol{X}}_k^{(i)}\right\|_2^2, \tag{4}$$

其中$\frac{1}{L_{\text{time}}}\sum_{j=1}^{L_{\text{time}}}\hat{\boldsymbol{T}}_j^{(i)}$表示第$i$层级时间特征投影后的均值，$\frac{1}{L_{\text{text}}}\sum_{k=1}^{L_{\text{text}}}\hat{\boldsymbol{X}}_k^{(i)}$表示第$i$层级文本特征投影后的均值。这2个均值相减后取二范数(即欧几里得距离的平方)，衡量了时间和文本特征在该层级投影后均值的差异程度。再除以$d$，得到第$i$层级的蒸馏损失$L_{\text{distill}}^{(i)}$。最终输出的层级蒸馏损失将作为总损失函数的重要组成部分，用于约束该层级跨模态特征的对齐质量。

3)层级对比损失

层级对比损失以前一层输出的平均池化特征$\bar{\boldsymbol{T}}^{(i)}$和$\bar{\boldsymbol{X}}^{(i)}$作为输入，通过对比学习机制增强跨模态表示的判别性。

在批次数据中将同一实例的时间特征$\bar{\boldsymbol{T}}_b^{(i)}$和文本特征$\bar{\boldsymbol{X}}_b^{(i)}$构建为正样本对，通过计算InfoNCE损失来最大化正样本对的相似度，同时推远与其他实例的负样本对距离。公式为

$$L_{\text{contrast}}^{(i)}=-\frac{1}{B}\sum_{b=1}^{B}\ln\frac{\exp(sim(\bar{\boldsymbol{T}}_b^{(i)},\bar{\boldsymbol{X}}_b^{(i)})/\tau_i)}{\sum_{k=1}^{B}\exp(sim(\bar{\boldsymbol{T}}_b^{(i)},\bar{\boldsymbol{X}}_k^{(i)})/\tau_i)}, \tag{5}$$

其中$sim(\cdot)$为余弦相似度计算，$\tau_i$是自适应温度参数。最终输出的对比损失将作为总损失函数的重要组成部分，用于提升跨模态特征的判别能力。

4)层级加权融合

层级加权融合模块以所有层级投影并池化后的时间特征表示$\{\bar{\boldsymbol{T}}^{(1)},\bar{\boldsymbol{T}}^{(2)},\cdots,\bar{\boldsymbol{T}}^{(N)}\}$和文本特征表示$\{\bar{\boldsymbol{X}}^{(1)},\bar{\boldsymbol{X}}^{(2)},\cdots,\bar{\boldsymbol{X}}^{(N)}\}$作为输入，通过特征拼接与加权融合实现多尺度信息整合。

首先将各层级的时间与文本全局表示进行拼接，并通过共享的融合网络$f_{\text{fusion}}$(通常为MLP)得到层级融合特征$\boldsymbol{F}^{(i)}=f_{\text{fusion}}(concat(\bar{\boldsymbol{T}}^{(i)},\bar{\boldsymbol{X}}^{(i)}))$，随后基于可学习的层级权重参数$\alpha_i$经$\sigma$($\sigma$指softmax函数)归一化后对各层级特征进行自适应加权，最终通过加权求和$\boldsymbol{F}=\sum_{i=1}^{N}\sigma(\alpha_i)\cdot\boldsymbol{F}^{(i)}$得到从局部细节到全局语义的多尺度跨模态信息的融合特征$\boldsymbol{F}\in\mathbb{R}^{B\times M}$，作为下游预测任务的增强输入。

通过上述操作，层次化知识蒸馏与对比学习模块在多个层级上对时间序列和文本模态特征进行了有效对齐与融合，增强了跨模态表示的一致性和鲁棒性。进一步，为实现模态间的精准匹配，设计了动态注意力跨模态匹配模块。

### 2.2 动态注意力跨模态匹配模块

该模块主要用于实现跨模态特征的有效对齐，通过动态调整注意力权重等方式，能更好地捕捉不同模态间的关键匹配信息。该模块以知识蒸馏模块输出的融合特征和原始分支特征为基础，进行精细化的跨模态交互。输入为时间目标分支的顶层特征$\boldsymbol{T}\in\mathbb{R}^{L_{\text{time}}\times B\times M}$和文本源分支的顶层特征$\boldsymbol{X}\in\mathbb{R}^{L_{\text{text}}\times B\times M}$。模块通过动态权重生成、门控注意力、对齐度预测与特征融合等一系列操作，最终输出一个经过跨模态信息增强的时间序列特征$\boldsymbol{T}_{\text{enhanced}}\in\mathbb{R}^{L_{\text{time}}\times B\times M}$，以及反映本轮交互质量的对齐度分数。此模块算法为附录A的算法A2和算法A3。

1)自适应注意力权重

自适应权重生成机制以时间特征$\boldsymbol{T}$与文本特征$\boldsymbol{X}$作为输入，通过模态交互实现动态权重分配。首先对2种模态特征分别进行序列维度的平均池化，得到全局表征$\bar{\boldsymbol{T}}=Mean(\boldsymbol{T})$和$\bar{\boldsymbol{X}}=Mean(\boldsymbol{X})$，随后将二者拼接后送入权重生成网络$MLP$与$\sigma$($\sigma$指softmax函数)组合，通过计算

$$[\omega_{\text{self}},\omega_{\text{cross}}]=\sigma(MLP(concat(\bar{\boldsymbol{T}},\bar{\boldsymbol{X}}))), \tag{6}$$

动态生成一对自适应权重标量，分别量化自注意力特征与交叉注意力特征在最终融合中的相对重要性，为后续的特征融合提供依据。

该机制在增强模型表达能力的基础上，进一步从优化过程层面提升了训练的稳定性与收敛性。具体而言，在传统静态权重分配策略中，预设的权重若与当前批次数据的模态关联性不匹配，易导致梯度更新产生剧烈波动，进而引发训练不稳定的问题。而本机制通过基于实例特征动态生成权重，使模型能够平滑适应不同数据样本的模态特性变化。这种以数据为中心的权重分配策略，有效缓冲了批次间数据分布差异带来的冲击，使得损失曲面更加平滑，梯度更新更为稳定。与此同时，该机制为模型提供了一条学习权重分配的高效路径，模型无需通过漫长训练隐式地探索固定权重配置，而是显式地学习如何分配权重的元策略。这不仅大幅降低了优化难度，加速了收敛进程，更使得模型在训练初期即可快速锁定各模态的相对重要性，避免因初始权重配置不当而陷入次优优化路径，从而提升收敛至更优解的可能性。

2)门控注意力机制

门控注意力机制以时间特征$\boldsymbol{T}$作为$\boldsymbol{Q}$、文本特征$\boldsymbol{X}$作为$\boldsymbol{K}$和$\boldsymbol{V}$进行跨模态交互，通过引入门控单元实现注意力分数的自适应调节。

首先计算原始注意力分数$\boldsymbol{A}_{\text{raw}}=\dfrac{\boldsymbol{T}\boldsymbol{X}^{\mathrm{T}}}{\sqrt{d_{\boldsymbol{K}}}}$，同时通过对$\boldsymbol{Q}$沿特征维度均值化并应用$\sigma$($\sigma$指 sigmoid 函数)生成门控值$\boldsymbol{G}=\sigma\left(\dfrac{1}{d_{\text{time}}}\sum\boldsymbol{T}\right)$，随后将门控值与原始注意力分数进行逐元素相乘并经$\sigma$($\sigma$指 softmax 函数)归一化得到最终注意力权重

$$\boldsymbol{A}=\sigma(\boldsymbol{A}_{\text{raw}}\odot\boldsymbol{G})\,, \tag{7}$$

其中⊙为逐元素乘法。

利用门控机制调节后的注意力权重$\boldsymbol{A}$，对文本特征$\boldsymbol{X}$进行加权求和，从而提取出与时间序列最相关的文本信息。$\boldsymbol{V}_{\text{cross}}=\boldsymbol{A}\cdot\boldsymbol{X}$，最终得到的$\boldsymbol{V}_{\text{cross}}$是跨模态交互的输出特征。

3)对齐度预测

对齐度预测以交叉注意力交互后的特征$\boldsymbol{V}_{\text{cross}}$作为输入，通过量化评估机制衡量跨模态对齐质量。

首先对跨模态特征$\boldsymbol{V}_{\text{cross}}$沿序列维度进行平均池化获得全局表征，随后通过一个由全连接层与激活函数构成的预测网络 $MLP$+$\sigma$($\sigma$指 sigmoid 函数)进行非线性变换，通过计算

$$\alpha=\sigma(\boldsymbol{W}_2\cdot ReLU(\boldsymbol{W}_1\cdot Mean(\boldsymbol{V}_{\text{cross}})+\boldsymbol{b}_1)+\boldsymbol{b}_2)\,, \tag{8}$$

得到范围在 0~1 的对齐度分数$\alpha\in\mathbb{R}^{B\times1}$，该分数将作为损失函数的重要调节因子，用于根据模态间实际匹配程度动态优化训练过程。

4)动态特征融合

动态特征融合模块以原始时间特征$\boldsymbol{T}'$、交叉注意力输出$\boldsymbol{V}_{\text{cross}}$以及自适应权重$\omega_{\text{self}}$，$\omega_{\text{cross}}$作为输入，通过加权融合机制实现跨模态信息整合。

将自注意力路径的特征与交叉注意力路径的特征，按照学习到的权重系数进行线性加权组合，并通过层归一化稳定训练过程，计算公式为

$$\boldsymbol{T}_{\text{enhanced}}=LayerNorm(\omega_{\text{self}}\cdot\boldsymbol{T}'+\omega_{\text{cross}}\cdot\boldsymbol{V}_{\text{cross}})\,。 \tag{9}$$

最终输出增强后的时间序列特征$\boldsymbol{T}_{\text{enhanced}}\in\mathbb{R}^{L_{\text{time}}\times B\times M}$，该特征在保留时间序列自身特性的同时，有效融入了与之高度相关的文本语义信息。

动态注意力跨模态匹配模块借助上述多种注意力相关机制，在跨模态特征的对称和融合上发挥了重要作用，提升了特征交互效果。但为了让模型实现更高效的优化，还需要构建合适的整体模型目标函数来进行训练。

### 2.3 整体模型目标函数

1)总损失函数

总损失函数通过整合特征正则化、输出一致性、知识蒸馏和对比学习等多任务的误差，综合衡量模型在各方面的表现，引导模型优化训练，实现各项任务指标的平衡，进而提升整体性能。公式为

$$L_{\text{total}}=L_{\text{sup}}+\lambda_{\text{f}}L_{\text{feature}}+\lambda_{\text{o}}L_{\text{output}}+\lambda_{\text{d}}L_{\text{distill}}+\lambda_{\text{c}}L_{\text{contrast}}\,。 \tag{10}$$

2)对齐度修正

对齐度修正根据对齐度分数动态调整总损失函数的权重。匹配程度高时降低权重，减少学习强度；匹配程度低时增加权重，加大学习强度。这种动态调整使模型能依据实际匹配程度灵活优化训练。公式为

$$L_{\text{final}}=L_{\text{total}}\cdot(1+E[\alpha])\,。 \tag{11}$$

整体目标函数通过总损失函数和对齐度修正，全面且灵活地指导了模型的训练过程。它整合了模型在各个关键环节的表现评估，使模型能够朝着更优的方向进行参数调整和优化。

### 2.4 与基础 CALF 模型的对比增强部分

Enhanced CALF 在基础 CALF 框架的基础上进行了系统性增强。Enhanced CALF 算法为附录 A 的算法 A4。首先，针对基础 CALF 模型中静态跨模态匹配模块的局限性，将它扩展为动态注意力跨模态匹配模块。该模块新增了自适应权重生成机制、门控注意力单元以及对齐度预测机制，实现了依据数据分布特性和模态相关性动态调整的模态交互，从而有效提升了模态对齐的质量。其次，在特征对齐方面，引入了层次化知识蒸馏与对比学习模块。该模块通过在各 Transformer 层级部署投影映射，并结合层级蒸馏损失与自适应温度对比损失，有效解决了基础 CALF 模型中单一层级对齐所导致的特征漂移问题，实现了更全面的跨模态特征融合。在损失函数设计中，新增了自适应对齐机制。该机制基于对齐度分数对模态匹配质量进行量化评估，并依据此分数动态调整总损失权重，从而使模型能够更好地适应不同模态匹配场景下的分布差异。最后，整体目标函数在基础 CALF 框架的基础上，新增了层级蒸馏损失与层级对比损失，通过多层级特征对齐与对比约束，进一步强化跨模态表示的一致性与判别性。

## 3 实　　验

为了验证 Enhanced CALF 的有效性，设计了全面

的实验，包括长期时间序列预测和少样本学习任务，并分别从预测精度、跨模态对齐能力和计算效率方面与当前最先进的基线方法进行比较。

### 3.1 实验设置

#### 3.1.1 数据集

本文采用7个真实数据集，涵盖不同领域背景，包括变压器温度(ETT)数据集(含ETTh1, ETTh2, ETTm1, ETTm2这4个子集)、天气数据集、电力数据集和交通数据集。

#### 3.1.2 评价指标

在实验中，采用以下评价指标对模型性能进行评估：

1)平均绝对误差(mean absolute error, *MAE*)

*MAE*是预测值与真实值差值的绝对值的平均值。计算公式为

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|\ , \tag{12}$$

其中$n$为样本数量，$y_i$为真实值，$\hat{y}_i$为预测值。

*MAE*可以直接反映预测值与真实值的平均偏离程度，对异常值的敏感度较低，结果更符合对误差的直观理解。

2)均方误差(mean square error, *MSE*)

*MSE*是预测值与真实值差值的平方的平均值，计算公式为

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\ 。 \tag{13}$$

*MSE*通过对误差项进行平方处理，放大较大误差的影响，对预测结果中的极端偏差更为敏感，常用于衡量模型的整体预测精度和稳定性。

在长期预测与少样本学习任务中，本文选取*MSE*与*MAE*作为核心性能评价指标。前者侧重评估模型对整体误差的控制能力，后者则更直观地反映预测值的平均偏离程度，二者结合可全面衡量模型在不同场景下的预测性能。

#### 3.1.3 模型参数设置

在Enhanced CALF的实现中，使用预训练GPT-2的前6层Transformer作为主干，并用Adam优化器训练。选择GPT-2作为主干是因为它能够有效捕捉文本语义和上下文信息，提供丰富的语义特征，同时Transformer架构能够高效处理序列数据并捕捉长期依赖关系。预训练与微调相结合的策略使模型能够快速适应特定领域任务，提升在少样本学习场景下的泛化能力和预测精度。此外，由于基础CALF模型同样采用了GPT-2作为主干，Enhanced CALF延续这一选择更有利于进行公平对比实验，确保性能提升源于所提出的增强模块而非主干网络差异。在损失函数设计方面，通过可学习权重参数动态平衡各损失项，包括特征正则化损失、输出一致性损失、层级蒸馏损失与层级对比损失，并通过对齐度预测机制自适应调整总损失权重，以提升模型对不同数据分布的适应性。

### 3.2 实验结果

#### 3.2.1 长期预测

为了公平地比较，输入时间序列长度$T$被固定为96，采用4个不同的预测时间范围$H\in\{96, 192, 336, 720\}$。为全面评估模型性能，本实验采用*MSE*与*MAE*作为评价指标。

基于长期预测实验结果，如表2、图2与图3所示，Enhanced CALF模型在多个真实数据集上相较于基础CALF模型实现了性能提升。具体而言，在ETTm1数据集上*MSE*相对降低5.74%、*MAE*相对降低3.04%，在ETTm2数据集上*MSE*相对降低6.94%、*MAE*相对降低8.95%，在ETTh1数据集上*MSE*相对

**Table 2 Comparative Table of Long-Term Prediction Results of Various Models on Different Datasets**

**表2 各模型在不同数据集上的长期预测结果对比表**

| 数据集 | Enhanced CALF（本文） | | CALF | | TimeLLM | | GPT4TS | | PatchTST | | iTransformer | | Crossformer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* |
| ETTm1 | **0.378** | **0.383** | 0.401 | 0.395 | 0.412 | 0.409 | 0.389 | 0.397 | 0.384 | 0.398 | 0.416 | 0.420 | 0.505 | 0.504 |
| ETTm2 | **0.268** | **0.295** | 0.288 | 0.324 | 0.298 | 0.343 | 0.291 | 0.335 | 0.285 | 0.327 | 0.297 | 0.341 | 1.219 | 0.712 |
| ETTh1 | **0.415** | **0.411** | 0.440 | 0.435 | 0.460 | 0.450 | 0.454 | 0.441 | 0.455 | 0.446 | 0.455 | 0.448 | 0.620 | 0.572 |
| ETTh2 | **0.321** | **0.351** | 0.351 | 0.384 | 0.411 | 0.414 | 0.384 | 0.411 | 0.374 | 0.401 | 0.384 | 0.409 | 0.942 | 0.684 |
| Weather | 0.262 | 0.285 | **0.250** | **0.274** | 0.275 | 0.290 | 0.265 | 0.286 | 0.258 | 0.280 | 0.259 | 0.281 | 0.259 | 0.315 |
| Electricity | **0.161** | **0.243** | 0.176 | 0.266 | 0.229 | 0.316 | 0.213 | 0.297 | 0.220 | 0.315 | 0.179 | 0.275 | 0.246 | 0.337 |
| Traffic | **0.403** | 0.315 | 0.442 | 0.321 | 0.546 | 0.363 | 0.488 | 0.317 | 0.564 | 0.371 | 0.428 | **0.285** | 0.550 | 0.304 |

注：黑体数值表示最优值。

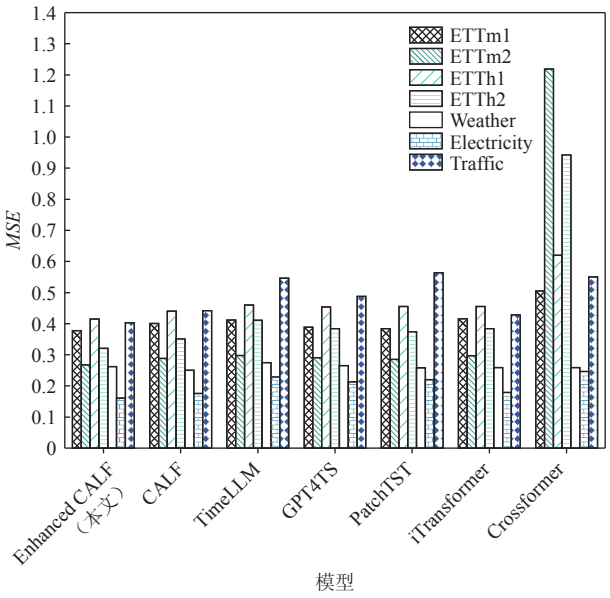



Fig. 2 *MSE* of different models in long-term prediction on multiple datasets

图 2 不同模型在长期预测多个数据集上的 *MSE*

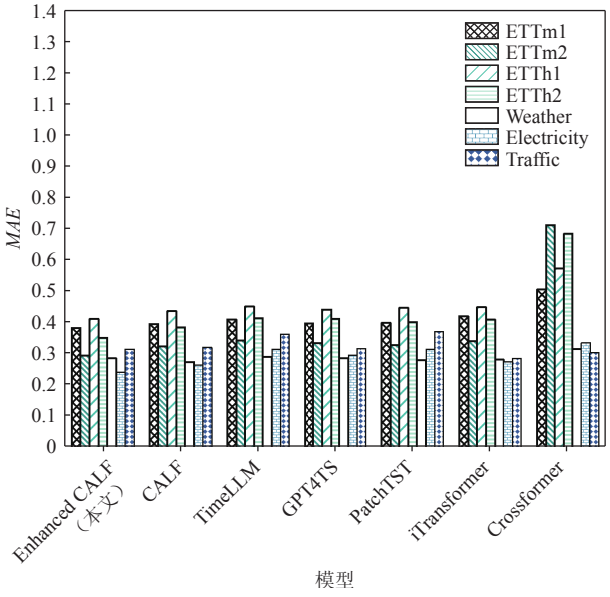



Fig. 3 *MAE* of different models in long-term prediction on multiple datasets

图 3 不同模型在长期预测多个数据集上的 *MAE*

降低 5.68%、*MAE* 相对降低 5.52%, 在 ETTh2 数据集上 *MSE* 相对降低 8.55%、*MAE* 相对降低 8.59%, 在 Electricity 数据集上 *MSE* 相对降低 8.52%、*MAE* 相对降低 8.65%, 在 Traffic 数据集上 *MSE* 相对降低 8.82%、*MAE* 相对降低 1.87%。上述 6 个数据集上均表现优越, 仅在 Weather 数据集上出现小幅上升, 其中 *MSE* 相对上升 4.80%、*MAE* 相对上升 4.01%。整体平均 *MSE* 相对降低达 7.38%, *MAE* 相对降低达 6.10%, 充分验证了 Enhanced CALF 在跨模态时间序列预测任务中的有效性和鲁棒性, 综合表现优于基线模型。基于以上观察, 进一步对实验结果进行定性分析。

1)跨模态对齐有效性分析

在 ETTh1, ETTh2 等电力变压器数据集上, Enhanced CALF 的性能优于基础 CALF, PatchTST, TimeLLM 等单模态时序基线模型。其根本原因在于, 电力负载数据往往受到运行日志、维护报告等文本信息的影响。基础 CALF 模型虽进行了跨模态融合, 但静态的注意力与对齐机制难以适应不同运行工况下模态关联性的动态变化。Enhanced CALF 引入的动态注意力机制与对齐度预测能够根据输入序列实时评估文本与时间序列的关联强度, 并自适应地调整融合策略, 从而在模态信息高度相关时强化互补, 在相关性弱时减少噪声干扰, 实现了更精准的模态对齐。相比之下, 单模态模型因完全缺失文本上下文信息而处于天然劣势。

2)复杂序列建模能力分析

在 Traffic 和 Electricity 这两类具有高噪声、复杂周期性与突发性模式的数据集上, Enhanced CALF 同样展现了最强的鲁棒性, Enhanced CALF 在 Traffic 数据集上的 *MSE* 为 0.403, 在 Electricity 数据集上的 *MSE* 为 0.161, 领先于 GPT4TS 和 TimeLLM 等同样利用大语言模型的基线。这主要归功于层次化知识蒸馏框架。传统模型通常在单一层级进行知识迁移, 无法捕捉时间序列中从短期局部波动到长期趋势的全尺度特征。Enhanced CALF 在 Transformer 的每一层都进行特征对齐与蒸馏, 使得模型能够同时学习到文本描述中的高级语义与时间序列中低级的局部模式, 从而对复杂动态序列的表征更为全面和深刻。

3)长期依赖捕捉能力分析

Crossformer 在 ETTh1 任务上的 *MSE* 为 0.620, 在 Traffic 任务上的 *MSE* 为 0.550, 这 2 项误差均为最大, 这凸显了 Crossformer 模型在捕捉长期依赖关系上的不足。对于长期预测任务, 模型必须有效地建模序列远端的历史信息与预测起点之间的依赖关系。Crossformer 虽然通过分段建模处理长序列, 但模块间的信息流动可能仍不够充分。相反, Enhanced CALF 以预训练的 GPT-2 为核心骨干, 继承了 Transformer 架构在长上下文建模上的强大能力, 并结合动态注意力机制进一步强化了对关键历史信息的筛选与记忆, 因此在预测范围较长的任务中展现出更强的性能。

综上所述, Enhanced CALF 的卓越性能源于架构

对 3 个关键挑战的系统性解决：动态注意力机制解决了模态对齐的适应性问题；层次化知识蒸馏解决了多尺度特征融合的问题；预训练大语言模型与对比学习则共同保障了长期依赖建模与表示的一致性。这使得 Enhanced CALF 模型无论是在相对稳定的电力数据，还是在高度复杂的交通流数据中，都能实现当前最优的预测性能。

3.2.2 少样本学习

为了全面验证所提出方法在数据稀缺环境下的泛化性能，本文设计了严格的少样本学习实验。本节实验评估模型在资源受限情况下的学习效率和预测性能。具体采用渐进式数据采样策略，分别使用 ETT 各数据集中前 5%, 10%, 15% 的训练样本进行模型训练。该实验设置通过极端数据约束，不仅能够有效检验模型从有限信息中提取关键特征的能力，更能充分验证在真实场景中应对数据不足挑战的实用性。实验结果见表 3~5。结果表明 Enhanced CALF 的方法优于大部分其他基线，充分证明了 Enhanced CALF 模型在少样本学习任务中具有鲁棒性。

在少样本学习场景下，Enhanced CALF 展现出优秀的数据效率与稳健的泛化能力。实验采用渐进式数据采样策略，分别使用 ETT 数据集前 5%, 10%, 15% 的训练样本进行模型训练。结果表明，Enhanced CALF 相比基础 CALF 在各实验数据比例上均取得性能提升。具体而言，在仅使用 5% 训练数据的条件下，模型已在多数数据集上表现优于基础 CALF 模型。例如，在 ETTm1 数据集上，其 *MSE* 相对降低 6.64%，*MAE* 相对降低 6.09%；在 ETTh2 数据集上，*MSE* 相对降低 8.74%，*MAE* 相对降低 5.43%。当训练数据比例提升至 10% 时，ETTh1 数据集上的 *MSE* 相对降低达到 9.09%，*MAE* 相对降低为 8.13%。在训练数据比例继续增加至 15% 的情况下，在 ETTm1 数据集上实现了 *MSE*

**Table 3 Results of Few Sample Learning (5% ETT Dataset)**

**表 3 少样本学习结果（5%ETT 数据集）**

| 数据集 | Enhanced CALF（本文） | | PatchTST | | TimeLLM | | GPT4TS | | CALF | | Crossformer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* |
| ETTm1 | **0.605** | **0.524** | 0.695 | 0.591 | 0.815 | 0.668 | 0.782 | 0.652 | 0.648 | 0.558 | 1.505 | 0.952 |
| ETTm2 | 0.415 | 0.395 | **0.391** | **0.372** | 0.465 | 0.421 | 0.418 | 0.388 | 0.398 | 0.382 | 2.201 | 1.148 |
| ETTh1 | **0.763** | **0.623** | 0.847 | 0.663 | 0.947 | 0.721 | 0.875 | 0.681 | 0.803 | 0.652 | 1.895 | 1.053 |
| ETTh2 | **0.564** | **0.522** | 0.675 | 0.571 | 0.725 | 0.598 | 0.705 | 0.588 | 0.618 | 0.552 | 3.396 | 1.442 |

注：黑体数值表示最优值。

**Table 4 Results of Few Sample Learning (10% ETT Dataset)**

**表 4 少样本学习结果（10%ETT 数据集）**

| 模型 | Enhanced CALF（本文） | | PatchTST | | TimeLLM | | GPT4TS | | CALF | | Crossformer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* |
| ETTm1 | **0.460** | **0.426** | 0.560 | 0.492 | 0.639 | 0.514 | 0.614 | 0.502 | 0.506 | 0.464 | 1.344 | 0.852 |
| ETTm2 | 0.325 | 0.345 | **0.295** | 0.334 | 0.348 | 0.343 | 0.306 | 0.338 | 0.302 | **0.330** | 1.988 | 1.050 |
| ETTh1 | **0.616** | **0.518** | 0.684 | 0.547 | 0.766 | 0.586 | 0.688 | 0.555 | 0.650 | 0.544 | 1.744 | 0.914 |
| ETTh2 | **0.439** | **0.443** | 0.550 | 0.487 | 0.589 | 0.498 | 0.580 | 0.498 | 0.491 | 0.472 | 3.149 | 1.385 |

注：黑体数值表示最优值。

**Table 5 Results of Few Sample Learning (15% ETT Dataset)**

**表 5 少样本学习结果（15%ETT 数据集）**

| 模型 | Enhanced CALF（本文） | | PatchTST | | TimeLLM | | GPT4TS | | CALF | | Crossformer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* | *MSE* | *MAE* |
| ETTm1 | **0.419** | **0.381** | 0.495 | 0.435 | 0.561 | 0.468 | 0.545 | 0.458 | 0.462 | 0.418 | 1.150 | 0.801 |
| ETTm2 | 0.278 | 0.315 | **0.265** | **0.302** | 0.285 | 0.318 | 0.272 | 0.310 | 0.275 | 0.305 | 1.005 | 0.685 |
| ETTh1 | **0.504** | **0.447** | 0.605 | 0.498 | 0.658 | 0.535 | 0.625 | 0.510 | 0.558 | 0.485 | 1.350 | 0.850 |
| ETTh2 | **0.384** | **0.375** | 0.485 | 0.440 | 0.515 | 0.455 | 0.505 | 0.450 | 0.425 | 0.392 | 2.800 | 1.250 |

注：黑体数值表示最优值。

相对降低 9.31% 和 *MAE* 相对降低 8.85%。Enhanced CALF 在各数据比例下优于大部分基线模型，体现了其在不同数据规模下的稳定性能与良好的可扩展性。得注意的是，随着训练数据从 5% 逐步增加至 15%，所有模型的预测性能均呈现提升趋势，但 Enhanced CALF 在 ETTm1, ETTh1, ETTh2 这 3 个关键数据集上始终展现出最稳定的泛化表现。这一现象揭示了模型内在机制具有应对数据稀缺场景的独特优势。

基于以上观察，进一步对实验结果进行定性分析：

1）层次化蒸馏机制的泛化能力分析

在 ETTh2 和 ETTm1 等数据集上，Enhanced CALF 相较于 PatchTST 和基础 CALF 模型在不同数据规模下均实现了提升，尤其在仅使用 5% 数据的极端场景下，性能优势最为突出。核心原因在于模型引入了更强大的归纳偏置，从而在不同数据规模下均能保持较高的学习效率。

层次化知识蒸馏作为有效的正则化器，在不同数据规模下均发挥了重要作用。当训练数据极其有限（如 5%）时，模型极易过拟合到训练集的噪声中；即使在数据量有所增加（如 15%）时，过拟合风险依然存在。Enhanced CALF 的层次化蒸馏机制通过在 Transformer 的每一个层级都施加跨模态对齐约束，相当于为模型训练注入了持续的正则化效果。这种层层监督迫使模型学习文本与时间序列之间本质的关联特征，而非表面的相关性，从而在不同数据规模下的测试集上均表现出优异的泛化能力。

2）动态机制的数据规模适应性分析

实验结果显示，基础 CALF 在 ETTm2 等数据集上的表现随着数据规模变化而波动较大，这凸显了 Enhanced CALF 中动态机制在不同数据规模下的关键作用。当训练数据从 5% 增加到 15% 时，数据分布的覆盖范围逐渐扩大，但不确定性依然存在。基础 CALF 采用的静态跨模态融合策略无法自适应地应对这种动态变化的环境。

相反，Enhanced CALF 的动态注意力与对齐度预测模块能够根据输入样本和当前数据规模的特点智能调整融合策略。在数据极度稀缺如 5% 的情况下，该机制能自动降低不可靠的跨模态信号权重，更多地依赖时间序列本身的信息；当数据量增加至 15% 时，又能适时提升有效模态交互的强度。这种自适应能力使模型在不同数据规模下都能作出稳健决策，减少了性能波动。

3）过拟合鲁棒性与数据利用效率分析

为定量评估模型的过拟合风险，本研究对比了模型在 5%, 10%, 15% 这 3 种数据规模下的性能变化。实验结果表明，Enhanced CALF 未出现此现象。如表 3~5 所示，随着训练数据从 5% 增至 15%，Enhanced CALF 在多数 ETT 数据集上的 *MSE* 和 *MAE* 均呈现持续下降。例如，在 ETTh2 数据集上，*MSE* 从 5% 时的 0.564，稳步下降至 10% 时的 0.439，并进一步降至 15% 时的 0.384。这一单调下降趋势证明，Enhanced CALF 始终能够从新增的训练样本中有效学习，其性能提升是泛化能力真实提高的体现。

相比之下，部分基线模型（如 Crossformer）在数据量增加时性能改善有限，甚至在个别数据集上出现波动，这表明可能已陷入过拟合或无法有效利用更多数据。Enhanced CALF 通过内置的正则化设计，有效避免了这一常见问题，提升了数据利用效率与模型的泛化性能。

综上所述，在少样本学习的多数据规模测试中，Enhanced CALF 表现出良好的鲁棒性。该模型通过层次化蒸馏机制持续增强抗过拟合能力，结合预训练骨干网络注入的稳定先验知识，并利用动态自适应策略来保障不同数据规模下的决策可靠性。实验结果表明，该框架在电网等数据获取成本高昂的实际场景中，能够适应不同程度的数据可获得条件，在少样本环境下均能保持预测精度稳定可靠。

## 4 结　　论

本文针对基于大语言模型的跨模态时间序列预测在少样本学习与长期依赖建模中面临的挑战，提出了增强式跨模态微调框架 Enhanced CALF。该框架通过集成动态注意力与层次蒸馏两大核心机制，提升了模型的性能。首先，通过动态注意力跨模态匹配模块，依据数据分布特性和模态关联强度自适应地调整注意力权重，实现更精准的模态交互；其次，通过层次化知识蒸馏与对比学习模块，在 Transformer 各层级进行特征投影与对齐，并结合层级蒸馏损失与自适应温度对比损失，实现从局部到全局的全层级知识迁移；最后，通过自适应对齐机制，基于对齐度分数动态优化损失函数，进一步提升训练效率与模型鲁棒性。实验结果表明，Enhanced CALF 模型在长期预测任务中表现优异，平均 *MSE* 相对降低了 7.38%，*MAE* 相对降低了 6.10%。在少样本学习场景下，该模型仅使用 5% 的训练数据时，在多数数据集上已超越基础 CALF 模型。例如，在 ETTm1 和 ETTh2 数据集上，*MSE* 分别相对降低 6.64% 和 8.74%，*MAE* 分别相

对降低 6.09% 和 5.43%。当数据比例升至 10% 时，在 ETTh1 数据集上 *MSE* 相对降低 9.09%，*MAE* 相对降低 8.13%；提升至 15% 时，在 ETTm1 数据集上 *MSE* 相对降低 9.31%，*MAE* 相对降低 8.85%。理论分析与实验验证均表明，该框架能有效协调局部与全局模式，缓解语义漂移，提升模型在数据稀缺与长期依赖场景下的预测性能与泛化能力。未来研究将进一步探索该框架在更复杂跨模态场景中的应用以及与其他先进方法的融合。

**作者贡献声明**：丁琳琳、卢晶负责方案整体设计并撰写论文；李佳林负责算法思路和实验方法；李默、王媛娜提出指导意见并修改论文。

## 参 考 文 献


[1] Liu Peiyuan, Guo Hang, Dai Tao, et al. CALF: Aligning LLM for time series forecasting via cross-modal fine-tuning[J]. 2024, 39(18): 18915−18923

[2] Zhou Tian, Niu Peisong, Wang Xue, et al. One fits all: Power general time series analysis by pretrained LLM[C/OL]//Proc of the Neural Information Processing Systems. 2023[2026-02-04]. http://papers.nips.cc/paper_files/paper/2023/hash/86c17de05579cde52025f9984e6e2ebb-Abstract-Conference.html

[3] Chang C, Peng W, Chen T. LLM4TS: Two-stage fine-tuning for time-series forecasting with pre-trained LLM[J]. ACM Transactions on Intelligent Systems and Technology, 2025, 16(3): 2157−6904

[4] Cao Defu, Jia Furong, Arik S, et al. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting[C/OL]//Proc of the Int Conf on Learning Representations. 2024: 18546−18578.[2026-02-05]. https://proceedings.iclr.cc/paper_files/paper/2024/file/5132940b1bced8a7b28e9695d49d435a-Paper-Conference. pdf

[5] Sun Chenxi, Li Hongyan, Li Yaliang, et al. TEST: Text prototype aligned embedding to activate LLM's ability for time series[C/OL]//Proc of the Int Conf on Learning Representations. 2024: 37854−37881. [2026-02-05]. https://proceedings.iclr.cc/paper_files/paper/2024/file/a4352e2c9d93582898a2a20e1f514e8f-Paper-Conference.pdf

[6] Jin Ming, Wang Shiyu, Ma Lintao, et al. Time-LLM: Time series forecasting by reprogramming large language models[C/OL]//Proc of the Int Conf on Learning Representations. 2024: 23857−23880.[2026-02-05]. https://proceedings.iclr.cc/paper_files/paper/2024/file/680b2a8135b9c71278a09cafb605869e-Paper-Conference.pdf

[7] Ding Linlin, Han Yiming, Li Mo, et al. Adaptive anchor-based attention networks for large-scale sparse bipartite graph[J]. Knowledge-Based Systems, 2025, 329: 114242

[8] Chen Honghui, Zheng Jianming, Cai Fei, et al. Adaptive self training modeling based on minority sample relationship extraction[J]. Journal of Computer Research and Development, 2023, 60(7): 1581−1591 (in Chinese)
(陈洪辉, 郑建明, 蔡飞, 等. 基于适应性自训练的少样本关系抽取建模[J]. 计算机研究与发展, 2023, 60(7): 1581−1591)

[9] Yang Jieyi, Dong Yihong, Qian Jiangbo. Research progress on small sample learning methods based on graph neural networks[J]. Journal of Computer Research and Development, 2024, 61(4): 856−876 (in Chinese)
(杨洁祎, 董一鸿, 钱江波. 基于图神经网络的小样本学习方法研究进展[J]. 计算机研究与发展, 2024, 61(4): 856−876)

[10] Liu Binghao, Ding Yao, Jiao Jianbin, et al. Anti-aliasing semantic reconstruction for few-shot semantic segmentation[C]//Proc of the 39th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 9747−9756

[11] Xu Jian, Liu Bo, Xiao Yanshan. A variational inference method forfew-shot learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 33(1): 269−282

[12] Shao Shuai, Xing Lei, Wang Yanjiang, et al. Attention-based multi-view feature collaboration for decoupled few-shot learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 33(5): 2357−2369

[13] Wang Zhiyuan, Shi Long, Mei Zhen, et al. Iterative transfer knowledge distillation and channel pruning for unsupervised cross-domain compression[C]//Proc of the Web Information Systems and Applications. Berlin: Springer, 2024: 3−15

[14] Zhang Jing, Ju Jialiang, Ren Yonggong. Data free knowledge distillation based on dual generator network[J]. Journal of Computer Research and Development, 2023, 60(7): 1615−1627 (in Chinese)
(张晶, 鞠佳良, 任永功. 基于双生成器网络的 Data-Free 知识蒸馏[J]. 计算机研究与发展, 2023, 60(7): 1615−1627)

[15] Pan Haiwei, Yu Fengming, Zhang Kejia, et al. A review of knowledge distillation research for visual algorithms[J]. Journal of Computer Research and Development, 2026, 63(1): 90−122 (in Chinese)
(潘海为, 于丰铭, 张可佳, 等. 面向视觉算法的知识蒸馏研究综述[J]. 计算机研究与发展, 2026, 63(1): 90−122)

[16] Li Zheng, Li Xiang, Yang Lingfeng, et al. Curriculum temperature for knowledge distillation[C]//Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2023: 1504−1512

[17] Li Mingchao, Kun Huang, Xiao Ma, et al. Mask distillation network for conjunctival hyperemia severity classification[J]. Machine Intelligence Research, 2023, 20(6): 909−922

[18] Yang Chuanguang, An Zhulin, Cai Linhang, et al. Mutual contrastive learning for visual learning[C]//Proc of the 36th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2022: 3045−3053

[19] Yang Chuanguang, An Zhulin, Huang Libo, et al. CLIP-KD: An empirical study of clip model distillation[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2024: 15952−15962

[20] Wu Kan, Peng Houwen, Zhou Zhenghong, et al. TinyCLIP: Clip distillation via affinity mimicking and weight inheritance[C]//Pro of

the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 21970−21980

[21] Wang Shuangcheng, Zhu Songhao. Three stage few sample object detection based on cross module knowledge distillation[J]. Journal of Computer Research and Development, 2025, 63(1): 199−214 (in Chinese)
(王双丞, 朱松豪. 基于跨模块知识蒸馏的 3 阶段少样本目标检测[J]. 计算机研究与发展, 2025, 63(1): 199−214)

[22] Yu Ying, Wei Wei, Tang Hong, et al. Multi-stage training with multi-level knowledge self-distillation for fine-grained imagerecognition[J]. Journal of Computer Research and Development, 2023, 60(8): 1834−1845 (in Chinese)
(余鹰, 危伟, 汤洪, 等. 多层次知识自蒸馏联合多步骤训练的细粒度图像识别[J]. 计算机研究与发展, 2023, 60(8): 1834−1845)

[23] Zheng Hao, Wang Runqi, Liu Jianzhuang, et al. Cross-level distillation and feature denoising for cross-domain few-shot classification[C/OL]//Proc of the 18th Int Conf on Learning Representations. Berlin: Springer, 2023[2025-11-15]. https://openreview.net/forum?id=Kn-HA8DFik

[24] Chen Ting, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]//Proc of Int Conf on Machine Learning. New York: PMLR, 2020: 1597−1607

[25] He Kaiming, Fan Haoqi, Wu Yuxin, et al. Momentum contrast forunsupervised visual representation learning[C]//Proc of IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 9726−9735

[26] Jia Chao, Yang Yinfei, Xia Ye, et al. Scaling up visual and vision-language representation learning with noisy text supervision[C/OL]//Proc of Int Conf on Machine Learning. New York: ACM, 2021[2026-02-05]. https://proceedings.mlr.press/v139/jia21b.html

[27] Wang Yuanzheng, Sun Wenxiang, Fan Yixing, et al. Cross modal entity linking model based on contrastive learning[J]. Journal of Computer Research and Development, 2025, 62(3): 662−671 (in Chinese)
(王苑铮, 孙文祥, 范意兴, 等. 基于对比学习的跨模态实体链接模型[J]. 计算机研究与发展, 2025, 62(3): 662−671)

[28] Ding Linlin, Gu Yinghao, Li Mo, et al. HyIE: An internal-external induced embedding for knowledge hypergraph link prediction[J]. Information Fusion, 2025, 127: 103744



**Ding Linlin**, born in 1983. Postdoc, professor, PhD supervisor. Member of CCF. Her main research interests include big data management, graph data management, knowledge graph, smart mining, and artificial intelligence.

**丁琳琳**, 1983 年生。博士后, 教授, 博士生导师。CCF 会员。主要研究方向为大数据管理、图数据管理、知识图谱、智慧矿山、人工智能。



**Li Jialin**, born in 2002. Master candidate. Student member of CCF. Her main research interests include large models and vector databases. (2459116136@qq.com)

**李佳林**, 2002 年生。硕士研究生。CCF 学生会员。主要研究方向为大模型、向量数据库。



**Lu Jing**, born in 1992. PhD, lecturer. Member of CCF. Her main research interests include data mining and distributed computing.

**卢　晶**, 1992 年生。博士, 讲师。CCF 会员。主要研究方向为数据挖掘、分布式计算。



**Li Mo**, born in 1992. PhD, lecturer. Member of CCF. Her main research interests include database, knowledge graph, and graph data query. (limo@lnu.edu.cn)

**李　默**, 1992 年生。博士, 讲师。CCF 会员。主要研究方向为数据库、知识图谱、图数据查询。



**Wang Aina**, born in 1993. PhD, lecturer. Member of CCF. Her main research interests include optimal theory and applications, machine learning, and artificial intelligence. (18842551695@163.com)

**王媛娜**, 1993 年生。博士, 讲师。CCF 会员。主要研究方向为最优化理论与应用、机器学习、人工智能。

## 附录 A.

**算法 A1.** 层次化知识蒸馏与对比学习增强的表示学习算法。

输入: 时间特征列表 $time_features_list = \{\boldsymbol{X}_{\text{time}}^{(1)}, \boldsymbol{X}_{\text{time}}^{(2)}, \cdots, \boldsymbol{X}_{\text{time}}^{(N)}\}$, $\boldsymbol{X}_{\text{time}}^{(i)} \in \mathbb{R}^{L\times B\times M}$, 文本特征列表 $text_features_list = \{\boldsymbol{X}_{\text{text}}^{(1)}, \boldsymbol{X}_{\text{text}}^{(2)}, \cdots, \boldsymbol{X}_{\text{text}}^{(N)}\}$, $\boldsymbol{X}_{\text{text}}^{(i)} \in \mathbb{R}^{L\times B\times M}$, 隐藏层维度 $M$, 序列长度 $L$, 层级自适应温度 $\tau_i$;

输出: 蒸馏损失 $L_{\text{distill}}$, 对比损失 $L_{\text{contrast}}$, 融合特征 $L_{\text{fused}} \in \mathbb{R}^{B\times M}$。

① 初始化 $\boldsymbol{W}_{\text{time}}^{(i)}(i=1\sim L)$, $\boldsymbol{W}_{\text{text}}^{(i)}(i=1\sim L), \tau = \tau_i$;

② for $i$=1 到 $L$ do

/*步骤 1: 特征投影*/

③ $\hat{\boldsymbol{T}}^{(i)} = \boldsymbol{W}_{\text{time}}^{(i)} \cdot \boldsymbol{X}_{\text{time}}^{(i)} + \boldsymbol{b}_{\text{time}}^{(i)}$;

④ $\hat{\boldsymbol{X}}^{(i)} = \boldsymbol{W}_{\text{text}}^{(i)} \cdot \boldsymbol{X}_{\text{text}}^{(i)} + \boldsymbol{b}_{\text{text}}^{(i)}$;

/*步骤 2: 平均池化*/

⑤ $\bar{\boldsymbol{T}}^{(i)} = \dfrac{1}{L_{\text{time}}} \sum_{j=1}^{L_{\text{time}}} \hat{\boldsymbol{T}}_j^{(i)}$;

⑥ $\bar{\boldsymbol{X}}^{(i)} = \dfrac{1}{L_{\text{text}}} \sum_{k=1}^{L_{\text{text}}} \hat{\boldsymbol{X}}_k^{(i)}$;

/*步骤 3: 计算蒸馏损失*/

⑦ $L_{\text{distill}}^{(i)} = MSE(\bar{\boldsymbol{T}}^{(i)}, \bar{\boldsymbol{X}}^{(i)})$;

/*步骤 4: 计算对比损失*/

⑧ $\bar{\boldsymbol{T}}_b^{(i)} = normalize(\bar{\boldsymbol{T}}^{(i)}, 1)$;

⑨ $\bar{\boldsymbol{X}}_b^{(i)} = normalize(\bar{\boldsymbol{X}}^{(i)}, 1)$;

⑩ $S = \bar{\boldsymbol{T}}_b^{(i)} \bar{\boldsymbol{X}}_b^{(i)\mathrm{T}} / \tau_i$;

⑪ $L_{\text{contrast}}^{(i)} = CrossEntropy(S, y),\ y = 0, 1, \cdots, B-1$;

/*步骤 5: 特征融合*/

⑫ $\boldsymbol{F}^{(i)} = f_{\text{fusion}}(concat(\bar{\boldsymbol{T}}^{(i)}, \bar{\boldsymbol{X}}^{(i)}))$;

⑬ 储存 $\boldsymbol{F}^{(i)}$ 到 $layer_outputs$;

⑭ end for

/*步骤 6: 加权损失*/

⑮ $L_{\text{distill}} = \sum_{i=1}^{L} \alpha_i L_{\text{distill}}^{i}$;

⑯ $L_{\text{contrast}} = \sum_{i=1}^{L} \alpha_i L_{\text{contrast}}^{i}$;

/*步骤 7: 最终融合*/

⑰ $\boldsymbol{F} = \sum_{i=1}^{N} \sigma(\alpha_i) \cdot \boldsymbol{F}^{(i)}$, $\sigma$指 softmax 函数;

⑱ return $L_{\text{distill}}$, $L_{\text{contrast}}$, $\boldsymbol{F}$。

**算法 A2.** 动态注意力机制增强的跨模态匹配算法。

输入: 时间目标分支的顶层特征 $\boldsymbol{T} \in \mathbb{R}^{L_{\text{time}}\times B\times M}$, 文本源分支的顶层特征 $\boldsymbol{X} \in \mathbb{R}^{L_{\text{text}}\times B\times M}$, 隐藏层维度 $M$, 头数 $N$, drpout 率 $p$;

输出: 融合特征 $\boldsymbol{V}_{\text{cross}} \in \mathbb{R}^{L_{\text{time}}\times B\times M}$, 注意力权重 $\boldsymbol{A}$, 对齐得分 $\alpha$。

① 初始化 $SelfAttn \leftarrow MultiHeadAttention(M, N, p)$;

② $CrossAttn \leftarrow MultiHeadAttention(H, N, p)$;

③ $WeightGen \leftarrow Sequential(Linear(2M, M), ReLU, Linear(M, 2)), Softmax$;

④ $AlignPred \leftarrow Sequential(Linear(M, M/2), ReLU, Linear(M/2, 1)), Sigmod$;

⑤ $\tau \leftarrow 1.0$, $Norm1 \leftarrow LayerNorm(M)$, $Norm2 \leftarrow LayerNorm(M)$;

/*步骤 1: 计算时间序列自注意力*/

⑥ $\boldsymbol{A}_{\text{self}} \leftarrow SelfAttn(\boldsymbol{T}, \boldsymbol{T}, \boldsymbol{T})$;

⑦ $\boldsymbol{T}' \leftarrow Norm1(\boldsymbol{T} + Dropout(\boldsymbol{A}_{\text{self}}))$;

/*步骤 2: 跨膜态交叉注意力*/

⑧ $\boldsymbol{A}_{\text{cross}}, \boldsymbol{W}_{\text{attn}} \leftarrow CrossAttn(\boldsymbol{T}', \boldsymbol{X}, \boldsymbol{X}, need_weights = \text{True})$;

/*步骤 3: 全局特征计算*/

⑨ $\boldsymbol{g}_{\text{time}} \leftarrow mean(\boldsymbol{T}', dim = 0)$;

⑩ $\boldsymbol{g}_{\text{text}} \leftarrow mean(\boldsymbol{X}, dim = 0)$;

/*步骤 4: 动态权重生成*/

⑪ $\boldsymbol{c} \leftarrow concat([\boldsymbol{g}_{\text{time}}, \boldsymbol{g}_{\text{text}}], dim = -1)$;

⑫ $\boldsymbol{w} \leftarrow WeightgGen(\boldsymbol{c})$;

/*步骤 5: 对齐程度预测*/

⑬ $\alpha \leftarrow AlignPred(\boldsymbol{A}_{\text{cross}})$;

/*步骤 6: 加权特征融合*/

⑭ $\boldsymbol{F}_{\text{weight}} \leftarrow \omega_0 \cdot \boldsymbol{T}' + \omega_1 \cdot Droupt(\boldsymbol{A}_{\text{cross}})$;

⑮ $\boldsymbol{V}_{\text{cross}} \leftarrow Norm2(\boldsymbol{F}_{\text{weighted}})$;

⑯ return $\boldsymbol{V}_{\text{cross}}$, $\boldsymbol{A}$, $\alpha$。

**算法 A3.** 增强的注意力计算算法(带门控机制)。

输入: $\boldsymbol{T}$作为 $\boldsymbol{Q}$, $\boldsymbol{X}$作为 $\boldsymbol{K}$, $\boldsymbol{V}$;

输出: 注意力输出 $\boldsymbol{V}_{\text{cross}}$, 注意力权重 $\boldsymbol{A}$。

/*步骤 1: 计算原始注意力分数*/

① $\boldsymbol{A}_{\text{raw}} = \dfrac{\boldsymbol{T}\boldsymbol{X}^{\mathrm{T}}}{\sqrt{d_{\boldsymbol{K}}}}$;

/*步骤 2: 自适应门控机制*/

② $\boldsymbol{G} = \sigma\left(\dfrac{1}{d_{\text{time}}} \sum \boldsymbol{T}\right)$, $\sigma$指 sigmoid 函数;

/*步骤 3: 计算注意力权重*/

③ $\boldsymbol{A} = \sigma(\boldsymbol{A}_{\text{raw}} \odot \boldsymbol{G})$;

/*步骤 4: 应用注意力*/

④ $\boldsymbol{V}_{\text{cross}} = \boldsymbol{A} \cdot \boldsymbol{X}$;

⑤ return $\boldsymbol{A}$, $\boldsymbol{V}_{\text{cross}}$。

**算法 A4.** 增强版 CALF 算法。

输入: $\boldsymbol{P}_{\text{text}}$, $\boldsymbol{P}_{\text{time}} \in \mathbb{R}^{L\times B\times M}$, $text_config$, $time_config$,

$H$, $L$, $P$;

输出：$\boldsymbol{P}_{\text{text}}$，$\boldsymbol{P}_{\text{time}}$，$L_{\text{total}}$，$\boldsymbol{T}_{\text{enhanced}}$，$\boldsymbol{F}$。

/*步骤 1: 初始化编码器和模块*/

① $TextEncoder, TimeEncoder \leftarrow Transformer\text{-}Layers(L)$;

② $CrossModalMatch \leftarrow DynamicAttention\text{-}Module()$;

③ $HierarchicalDistill \leftarrow DistillationModule()$;

/*步骤 2: 编码特征*/

④ $\boldsymbol{X} \leftarrow encode_text(\boldsymbol{X}_{\text{text}})$;

⑤ $\boldsymbol{T} \leftarrow encode_time(\boldsymbol{X}_{\text{time}})$;

/*步骤 3: 应用动态注意力*/

⑥ if *use_dynamic_attention* then

⑦ $\boldsymbol{T}_{\text{enhanced}}, \alpha \leftarrow CrossModalMatch(\boldsymbol{X}, \boldsymbol{T})$;

⑧ else

⑨ $\boldsymbol{T}_{\text{enhanced}} \leftarrow \boldsymbol{T}, \alpha \leftarrow 0.0$;

⑩ end if

/*步骤 4: 应用层次化蒸馏*/

⑪ if *use_hierarchical_distillation* then

⑫ $L_{\text{distill}}, L_{\text{contrast}}, \boldsymbol{F} \leftarrow HierarchicalDistill(\boldsymbol{T}, \boldsymbol{X})$;

⑬ else

⑭ $L_{\text{distill}}, L_{\text{contrast}} \leftarrow 0.0$, $\boldsymbol{F} \leftarrow mean([\boldsymbol{T}, \boldsymbol{X}])$;

⑮ end if

/*步骤 5: 计算对齐因子*/

⑯ $f_{\text{aligned}} = 1.0 + mean(\alpha)$;

/*步骤 6: 生成预测*/

⑰ $\boldsymbol{P}_{\text{time}} \leftarrow TimeHead(\boldsymbol{F})$;

⑱ $\boldsymbol{P}_{\text{text}} \leftarrow TextHead(\boldsymbol{T})$;

/*步骤 7: 计算损失*/

⑲ $L_{\text{sup}} \leftarrow MSE(\boldsymbol{P}_{\text{time}}, \boldsymbol{P}_{\text{text}})$;

⑳ $L_{\text{feature}} \leftarrow \frac{1}{L}\sum_{i=1}^{L} MSE(T_i, X_i)$;

㉑ $L_{\text{output}} \leftarrow MSE(\boldsymbol{P}_{\text{time}}, \boldsymbol{P}_{\text{text}})$;

/*步骤 8: 总损失计算*/

㉒ $L_{\text{total}} = L_{\text{sup}} + \lambda_{\text{f}} L_{\text{feature}} + \lambda_{\text{o}} L_{\text{output}}$;

㉓ if *use_hierarchical_distillation* then

㉔ $L_{\text{total}} = \lambda_{\text{d}} L_{\text{distill}} + \lambda_{\text{c}} L_{\text{contrast}}$;

㉕ end if

㉖ if *use_dynamic_attention* then

㉗ $L_{\text{total}} = L_{\text{total}} \cdot f_{\text{aligned}}$;

㉘ end if

㉙ return $\boldsymbol{P}_{\text{text}}$，$\boldsymbol{P}_{\text{time}}$，$L_{\text{total}}$，$\boldsymbol{T}_{\text{enhanced}}$，$\boldsymbol{F}$。