变革车辆诊断：错误模式预测的多模态方法

Hugo Math[1],Rainer Lienhart[2]

arXiv:2602.01109v1 [cs.AI] 1 Feb 2026

摘要——**准确诊断和预测车辆故障，对于汽车行业的维护与安全至关重要。现代诊断系统主要依赖车载诊断（OBD）系统中记录的车辆诊断故障码（DTC）序列，却往往忽略温度、湿度和压力等原始传感器数据所包含的宝贵上下文信息。这些上下文数据对领域专家分类车辆故障至关重要，但其复杂性以及真实世界数据中的噪声特性也带来了独特挑战。本文提出BiCarFormer：首个将DTC序列与环境条件相结合、用于将错误代码分类为错误模式的多标签序列分类多模态方法。BiCarFormer是一种针对车辆事件序列设计的双向Transformer模型，采用嵌入融合和协同注意力机制，以捕捉诊断码与环境数据之间的关系。在包含22,137个错误代码和360种错误模式的复杂真实世界汽车数据集上进行的实验结果表明，与仅依赖DTC序列和传统序列模型的方法相比，我们的方法显著提升了分类性能。这项工作凸显了纳入上下文环境信息对于实现更准确、更稳健的车辆诊断的重要性，从而降低维护成本并提升汽车行业的自动化流程。**

I. 引言

MODERN 智能车辆会产生海量数据。这些数据通常以事件的形式记录，且事件发生的时间并不规律。有些事件会同时发生，而另一些事件之间的间隔则不均匀。这些数据往往包含冗余且带有噪声的类别特征和数值特征，因此需要有效利用这些数据，以改进车辆诊断和预测性维护。在这项工作中，我们重点处理并分析现代车辆生成的多变量、不规则事件流——这是一个仍需深入研究的领域。这些事件序列称为诊断故障码（DTC），记录了车辆整个生命周期中在时间和空间上发生的离散事件。与原始传感器数据相比，诊断故障码更受青睐，因为它们提供了更具结构化的离散

这些信息使其更易于分析。然而，在对这些DTC序列进行分类以识别错误模式（EPs）等关键故障时，领域专家通常会依赖额外的上下文数据（图1），例如车辆的环境条件（如温度、压力和电压读数）。领域专家在观察了大量DTC序列后定义EPs。这使他们能够更准确地判断车辆所存在的关键故障。尽管某些DTC也可能存在噪声和冗余

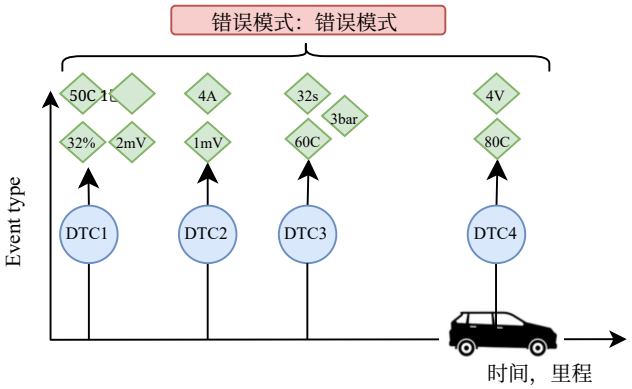


图 1. 基于过去的诊断故障码（DTC）和环境条件（例如温度、电压等）进行错误模式（EP）预测。

关于重复出现的错误（如软件更新、电气问题）的事件，EPs刻画了由精确车辆故障（如发动机或电池故障）组成的完整错误序列。当前研究仅依赖DTC，通过Transformer [1], [2] 和循环神经网络 [3] 预测后续DTC，或预测后续EPs [4]–[6]。因此，迫切需要利用这些额外的上下文数据。尽管这些辅助数据可以增强EPs的分类或聚类，但由于其数据量、可变性和高维度，整合这些数据也带来了挑战。我们的目标是揭示与特定EPs相关的诊断故障码（DTC）和环境条件的出现模式。我们证明，与仅依赖DTC [2], [3] 以及BERT [7]等经典“序列到序列模型”的传统模型相比，我们的方法显著提升了车辆错误模式（EPs）的多标签序列分类性能。通过提高车辆EPs的预测性能，我们的方法能够减少对昂贵领域专家的依赖，并推动传统车辆诊断转型。我们进一步通过解释交叉注意力分数来说明分类预测，并展示BiCarFormer能够检测连续值的波动以进行预测，尽管这些值涵盖了不同的单位以及数千种不同的环境条件。我们强调真实世界数据对所提方法的意义及其潜在的进一步应用。**贡献**。在本工作中，我们提出了BiCarFormer，这是首个通过整合诊断故障码（DTC）和环境条件来对车辆错误模式（EPs）进行分类的多模态双向Transformer。我们的主要贡献如下：

- 用于预测错误模式的多模态学习：不同于以往仅依赖DTC序列的方法，我们采用

[1]Hugo Math 任职于德国奥格斯堡 86159 的奥格斯堡大学。[2]Rainer Lienhart 教授、博士任职于德国奥格斯堡 86159 的奥格斯堡大学，担任机器学习与计算机视觉讲席主任。

多模态传感器融合，纳入环境条件（例如温度、电压、压力），以提高分类准确率。

- 协同注意力与事件嵌入融合：我们提出了一种新颖的融合机制，可高效整合多样的传感器输入，捕捉诊断故障码与环境数据之间有意义的交互。
- 增强可解释性：通过解释交叉注意力分数，我们深入了解 BiCarFormer 如何检测连续值中的波动，从而展示其在现实世界车辆诊断中的适用性。

II. 背景与相关工作

A. 车辆故障检测

历史上，人们一直使用机器学习模型来捕捉机器故障。传统的预测性维护（PdM）包括基于相关性 [8], 贝叶斯网络 [9], 决策树模型 [10],以及较新的神经状态空间模型 [11] 等技术，用于预测机器故障或剩余使用寿命（RUL）。这些方法依赖历史数据来识别模式并预测潜在故障。贝叶斯网络能够对车辆各个部件之间的概率依赖关系进行建模。一些论文探讨了将深度神经网络（DNN）用于预测性维护任务。例如， [12]在一个架构中结合卷积模块、全连接模块和 Transformer 模块，对机器事件日志进行分类。在车辆事件序列领域，一些论文建议使用 RNN、长短期记忆网络 [3], 和 Transformer [1] ，利用诊断故障码数据构建车辆诊断故障代码（DTC）序列，并预测下一个词元。 [2] 提出了 DTC-TranGRU，通过整合 Transformer 和循环单元，进一步提升了下一诊断故障代码预测的效果。然而，仅预测下一个诊断故障码在现实应用中存在局限；由于诊断故障码数量庞大，且与语言十分相似，这仍然是一项艰巨任务，正如 [4]所指出的。事实上，[4]建议使用车辆的保修数据构建一个监督学习问题，将错误模式（例如发动机故障、电池老化）与 DTC 序列关联起来。通过类比自然语言处理领域和机器生成数据，作者仅利用后验错误模式，就能推断车辆最有可能发生何种错误模式，进而利用 DTC 序列预测车辆的剩余使用寿命。本文将在 [2], [4] 的基础上，进一步对这些多变量车辆事件序列进行建模，并重点对错误模式进行分类，以评估所提出方法的性能。

B. 事件序列建模

基于事件数据构建事件序列模型，在多个领域都有大量实际应用。例如，在医疗保健领域，事件模型会读取电子健康记录（EHR），从而提供诊断和治疗计划 [13], [14]。在网络安全领域，事件模型可检测入侵并防范潜在的网络攻击 [15]。

这些数据通常由发生时间$t \in \mathbb{R}^+$ 和事件类型 $u \in U$ 组成，构成一个配对 $(t,\ u)$，其中 U 是由有限个离散事件类型组成的集合。随后，利用多个配对构建序列，例如$S = \{(t_1,\ u_1).....,(t_L,\ u_L)\}$，其中 $0 < t_1 < \ldots < t_n$。当特征不止事件类型及其发生时间时，我们通常称之为多变量事件序列。在本例中，每种事件类型还会对应环境条件 u。这类离散序列数据通常使用霍克斯过程 [16], 建模，我们希望根据事件历史 H_t 来估计下一个事件 $(u',\ t')$ 的概率：$= \{(t_i,\ u_i) \in \mathbb{R}^+ \times U \mid t_i < t\}$。随着自然语言处理的发展，基于神经网络的方法（RNN、LSTM、Transformer） [17], [18] 应运而生，并针对事件序列进行了相应调整，在各类预测基准 [19]上都展现出良好效果。

大多数采用 Transformer 模型处理序列数据的研究，都使用 [1] 提出的架构。我们将输入嵌入定义为 $\boldsymbol{U} = \boldsymbol{E} + \boldsymbol{PE} \in \mathbb{R}^{L \times d}$，其中 $\boldsymbol{E}$ 是事件类型嵌入， $\boldsymbol{PE}$ 是位置嵌入（通常是可学习的），用于指示每个词元的位置。我们定义三个线性投影矩阵，即查询 $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{W}_Q$、键 $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{W}_K$ 和值 $\boldsymbol{V} = \boldsymbol{U}\boldsymbol{W}_V$，其中 $\boldsymbol{W}_Q$、 $\boldsymbol{W}_K$ 和 $\boldsymbol{W}_V$ 是可训练权重。本质上， $\boldsymbol{Q}$ 表示模型根据输入 $\boldsymbol{U}$， $\boldsymbol{K}$ 寻找的内容； $\boldsymbol{K}$ 是输入信息的标签，而 $\boldsymbol{V}$ 是输入 $\boldsymbol{U}$ 的期望表示。注意力分数 $\boldsymbol{A}$ 的计算方式如下：

$$\boldsymbol{A} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right) \tag{1}$$

其中 d 表示隐藏维度， $\boldsymbol{A} \in \mathbb{R}^{L \times L}$ 表示每个事件对 i， j的注意力分数。新的表示通过以下方式计算

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{V} \tag{2}$$

最终的隐藏表示 $\boldsymbol{H}$ 通过层归一化（通常为LayerNorm）、逐点前馈神经网络（FFN）以及残差连接获得，具体如下：

$$\boldsymbol{C}' = \mathit{LayerNorm}(\boldsymbol{C} + \boldsymbol{U}) \tag{3}$$

$$\boldsymbol{H} = \mathit{LayerNorm}(\boldsymbol{C}' + \text{FFN}(\boldsymbol{C}')) \tag{4}$$

$\boldsymbol{H} \in \mathbb{R}^{L \times d}$ 随后由预测头用于掩码语言建模或下一词元预测。在下游任务中，我们通常取第一个logit $\boldsymbol{H}_0$ 用于序列分类（对应于BERT [7]中引入的 [CLS]词元），或对自回归 Transformer取 [EOS] （“序列结束”）。对于后一种情况，会对 $\boldsymbol{A}$施加因果掩码，使得词元只能关注先前的词元。

C. 多模态融合与学习

近年来，在 Transformer 模型中引入其他模态得到了大量研究 [20], [21]。在自然语言处理领域，以及更广泛的事件序列建模中，融合两个词元嵌入 $\boldsymbol{X}_A \in \mathbb{R}^{L_a \times d}$ 的一种常见且简单的方法是

$\boldsymbol{X}_B \in \mathbb{R}^{L_b \times d}$ 将模态 A 和 B 的嵌入进行早期求和。具体而言，在输入层对多个嵌入进行逐词元加权求和 的定义如下：

$$\boldsymbol{U} = \alpha \boldsymbol{X}_a \oplus \beta \boldsymbol{X}_b. \tag{5}$$

在 MedBERT [14], 中，作者将三种嵌入（诊断代码、顺序和每次就诊的位置）相加，形成输入 $\boldsymbol{U}$。这种方法有几个优点：它易于集成到 Transformer 中，且不会显著改变计算过程。然而，如果$L_a \neq L_b$需要将不同词元对齐或投影到潜在空间，那么使用 α，β求和时应如何为不同嵌入赋权并不明确。例如，BERT 会将词元嵌入 $\mathbf{E}$ 与其位置 $\mathbf{PE}$ 进行早期求和，使得 $\boldsymbol{U} = \boldsymbol{E} + \boldsymbol{PE}$。

在我们的方案中，DTC-TranGRU [2] 为三个 DTC 组件分别使用独立的嵌入，并沿 d拼接这些嵌入，以形成全局 DTC 嵌入 $\boldsymbol{D} \in \mathrm{R}^{L \times d}$ （我们将此方法称为早期拼接）。

$$\boldsymbol{U} = concat(\boldsymbol{X}_a, \boldsymbol{X}_b.) \in \mathbb{R}^{L \times (d_b + d_a)} \tag{6}$$

通过选择嵌入大小 d_a，d_b，这种方法能够更细粒度地整合附加特征。它还通过保持各自独立的维度，保留了每个特征的独特属性，因此有可能在更深层的注意力层中学习到更细致的表示和交互。不过，由于沿 d进行拼接，这种方法通常需要更多计算，并会为逐点前馈层和投影矩阵增加更多参数 [20] 。对于基数较低的特征（例如 BERT 中的词元类型ID [7] ），这种方法的效果也不佳，因为它们在隐藏维度中仅占据很小的范围 d，因此在这种情况下，早期求和更为可取。在早期阶段组合嵌入，模型便能学习到一种统一表示，以捕捉不同特征之间的交互。如果特征高度相关，或特征间的交互对准确预测至关重要，这种方法尤其有用。这些方法广泛应用于许多领域，尤其是医学 [14], [22] 中的电子健康记录（EHR）。处理音频或图像等不一定对齐的复杂模态时，通常会使用 Transformer 的注意力机制 [1]进行融合。Transformer 中的后期融合通常会通过独立的 Transformer编码器对不同模态进行编码，以提取高层表示 $\boldsymbol{Z}_a$，$\boldsymbol{Z}_b$:

$$\boldsymbol{Z_a} = Tf_a(X_a) \in \mathbb{R}^{L_a \times d}$$
$$\boldsymbol{Z}_b = Tf_b(X_b) \in \mathbb{R}^{L_b \times d}$$

随后，可以将这些表示拼接起来，并通过另一个第三个 Transformer（*HierarchicalAttention*）进行编码，利用多头注意力输出融合后的表示:

$$\boldsymbol{Z} = Tf(concat(\boldsymbol{Z}_a, \boldsymbol{Z}_b)) \in \mathbb{R}^{L_a \times (d_a + d_b)}$$

还可以像 CLIP[23] 中那样，将 $\boldsymbol{Z}_a, \boldsymbol{Z}_b$ 与相似度乘积进行融合。CLIP 模型引入了一项新的自监督多模态学习任务，使模型学习哪段图像说明文字与哪幅图像相匹配，并在 30 个不同的现有计算机视觉数据集上展现出 SOTA 性能

该方法的问题在于其跨模态交互有限：我们首先分别对各个模态进行编码，之后再将它们融合。或许，如果模型能在 Transformer 中更早地学习多模态依赖关系，就能从中受益。

另一方面，中间融合（或称中期融合）方法涉及交叉注意力，更广泛地说，还包括协同注意力 机制。在 ViLBERT [24] 和 LXMBERT [25], 中，注意力模型根据图像和文本输入计算注意力分数 $\boldsymbol{A}$ 。例如，以$\boldsymbol{X}_a$ 作为查询 $\boldsymbol{Q}_a$ ，以 $\boldsymbol{X}_b$ 作为 $\boldsymbol{K}_b$，$\boldsymbol{V}_b$，即可计算出相应的 $\boldsymbol{C}_{a \longrightarrow b} \in \mathbb{R}^{L_a \times d}$ ，反之亦然（一个输入关注另一个输入）。在协同注意力 中，两个输入通过同时计算两组注意力相互关注（$\boldsymbol{C}_{a \longrightarrow b}$，$\boldsymbol{C}_{b \longrightarrow a}$），从而实现双向的多模态学习。这些方法的一个缺点是：不将 $\boldsymbol{X}_a$，$\boldsymbol{X}_b$ 投影到固定的潜在空间 d ，而是计算新的 $\boldsymbol{A} \in \mathbb{R}^{L_a \times L_b}$，如果$L_b \gg L_a$，就会因原始注意力的二次时间复杂度 $O(n^2)$而大幅增加计算开销。每个模态还会得到两个隐藏表示：$\boldsymbol{H}_a$，$\boldsymbol{H}_b$。乘积 $\boldsymbol{Q}.\boldsymbol{K}^T$ 承载了大部分计算，因此可以考虑选择特定的查询和键乘法，从而构建稀疏注意力变体，例如 LongFormer [26], 或 BigBird [27]。这些模型结合使用滑动注意力、全局注意力和随机注意力，将 LongFormer 的$O(n^2)$降低到 $O(w \times n)$的线性复杂度。另一种方法是将键和值投影到较低维空间，例如 Linformer [28] ，或使用多模态瓶颈变换器 [21], 在受限的潜在空间中计算交叉注意力。

我们注意到，事件序列模型中的特征融合尚未得到充分研究。大多数论文旨在融合每种事件类型对应的时间 t_i 信息 u_i ，通过早期求和 [12], [18], [29] 来整合时间成分；有时会省略时间嵌入中的位置嵌入 [29],，有时会将两者相加 [19], [30],，也有时会忽略它 [2]。

III. 数据

诊断故障码：诊断故障码

车辆中的各个电子控制单元（电子控制单元）会以不规则的时间间隔或同时生成诊断数据。诊断数据与原始传感器数据有很大不同，因为诊断数据是类别型的，且涉及车辆中的各种问题和软件状态。诊断故障码（诊断故障码）由同一时间戳 t_s 和里程 m_a下出现的三项信息构成，即：(1) 电子控制单元的识别编号，(2) 错误代码（基础DTC），以及 (3) 故障字节。单个诊断故障码由这三个元素组成，具体如下:

$$\text{DTC} = \text{ECU} \mid \text{Base-DTC} \mid \text{Fault-Byte}.$$

本研究使用了一个匿名车辆诊断故障码序列数据集，其中包含 500 万条序列，每条序列平均有 $L \approx 150 \pm 90$个诊断故障码。每条序列 S_{raw} 都属于一辆独立的车辆。在 S_{raw}中，每个诊断故障码（通常称为事件类型 u_i）都附带

• 时空特征：时间 t_i 和里程 m_i。• 一个小型序列 $S_i^e = \{(d_j,\ v_j,\ c_j)\}_{j=0}^{L_i^e}$ 环境条件，分别包含描述、值和单位等元素。

这些元素构成一个单独的事件（$u_i,\ t_i,\ m_i,\ S_i^e$）。为获得完整序列 $S_{raw} = \{(u_i,\ t_i,\ m_i,\ S_i^e)\}_{i=0}^{L}$，我们获取最近已知的时间戳 t_{sL} 和里程 m_{aL}，并选取距今不超过以下范围的所有诊断故障码：（1）给定的过去时间段（$t_{sL} - t_{si} < 30$ 天）；（2）给定的过去距离（$m_{aL} - m_{ai} \leq 300$ 千米）。随后，我们将 S_{raw} 拆分为两个不同的序列：$S = \{(u_i,\ t_i,\ m_i)\}_{i=0}^{L}$ 表示诊断故障码，$S^e = \{(d_i,\ v_i,\ c_i)\}_{i=0}^{L^e}$ 仅表示长度为 $L^e \approx 2275 \pm 2310$ 的环境条件。表 I 概述了这些元素，表 II 则列出了每个特征的不同值数量。

表 I 符号及其相应描述列表。

符号	描述
u	离散事件类型
m_a	车辆的绝对里程（km）
t_s	附加到每个诊断故障码的 Unix 时间戳
S	由已定义的故障诊断代码组成的三元组序列 即 $S = \{(u_i,\ t_i,\ m_i)\}_{i=0}^{L}$，长度为 L
S^e	环境条件三元组序列定义为 $S^e = \{(d_i,\ v_i,\ c_i)\}_{i=0}^{L^e}$，长度为 $L^e \gg L$
d	环境条件描述，例如温度 增加、车速、压力增加
v	环境条件值可以是整数、浮点数、字符串或 null
c	环境条件单位，例如 A、V、巴、摄氏度、秒

A. 环境条件

如何高效整合传感信息（例如用于预测性维护（PdM）的温度或压力值）与离散代码，仍是一个有待解决的问题 [2], [12]。直观来看，利用温度诊断车辆发动机的特定缺陷，或利用电池中的电压测量识别电芯故障，似乎轻而易举。领域专家通常会分析这些数据，以对车辆中的错误模式作出判断。然而，整合环境条件面临三项重大挑战：

1) 维度：描述的基数很高（$>10^3$），且值可能有多种类型：字符串、布尔值、整数、浮点数或 NaN。2) 可变性：各诊断故障码对应的数据各不相同；例如，我们无法确定是否会在 DTC1 中观察到“温度”，而在 DTC2 中观察到“压力”。3) 数据量：每个诊断故障码对应多个 e.条件，这些条件可能冗余、含噪，且在整个序列中重复出现 S^e。

维度问题会增加模型复杂度，通常需要先进行大量数据工程技术处理——
手动。可变性 使其难以整合到机器学习模型中；最后，数据量需要大量数据基础设施来管理各种情况，因为其规模通常是诊断故障码数据的数倍，并且会显著增加序列长度，而这正是 Transformer 中已知的一项挑战 [30]。我们通过以下方式部分应对这些挑战：

表 II 各特征的不同值数量。

Data	值的数量	描述
DTC	22,137	诊断故障码
ECU	132	电子控制单元
基础DTC	17,044	错误代码
故障字节	2	二进制值
E. 条件描述	2,559	（参见表 1：d）
E. 条件值	3,288	（参见表 1：v）
E. 条件单位	18	最常用的单位：u)

• 移除空值 、单位或描述。
• 按诊断故障码去除重复项：在同一 S_i^e 内多次出现的相同（$d,\ v,\ c$）三元组将被删除，仅保留一个实例。这显著减少了数据中的冗余和噪声，但我们仍观察到每个 S^e有 $L^e \approx 2275 \pm 2310$ 个三元组。这种方法的一个问题是，当多个诊断故障码出现在同一 t_i时，e.条件与诊断故障码会<非对齐>。在这种情况下，我们会丢弃首次出现之后的后续 e.条件。
• 选择最常用的前 18 个单位。

为了将所得 S^e 输入 Transformer，我们将连续值 $v \in \mathbb{R}^+$ 映射为具有有限词汇表的离散词元 [20]。因此，我们为每个单位c搜索最优分箱。我们采用 Greenwald-Khanna算法 [31]，其最坏情况下的内存需求为 $O(\frac{1}{\epsilon}\log(\epsilon N))$，并结合精确率 $\epsilon = 0.0001$，每个单位最多使用 $\theta = 4000$个词元 u，以限制潜在的 $18 \times \theta = 72000$ 值。在实践中，它们会发生重叠，并未达到最大值（表 II）。

IV. BICARFORMER

A. DTC嵌入

我们使用双向Transformer模型 [7]，并通过掩码语言建模对其进行训练。每个诊断故障码元素都嵌入到特定的特征空间中，分别是：电子控制单元 ($\boldsymbol{D}_{ecu} \in \mathbb{R}^{L \times d_{ecu}}$)、基础*DTC* ($\boldsymbol{D}_{base} \in \mathbb{R}^{L \times d_{base}}$)，以及故障字节($\boldsymbol{D}_f \in \mathbb{R}^{L \times d}$)，并使用独立的查找表。前两个沿特征维度拼接，类似于 [2]，再将 $\boldsymbol{D}_f$ 添加到结果中，如同 BERT[7] 中的词元类型标识。这与 [4] 相比，可减少词汇量和可训练参数，并确保高

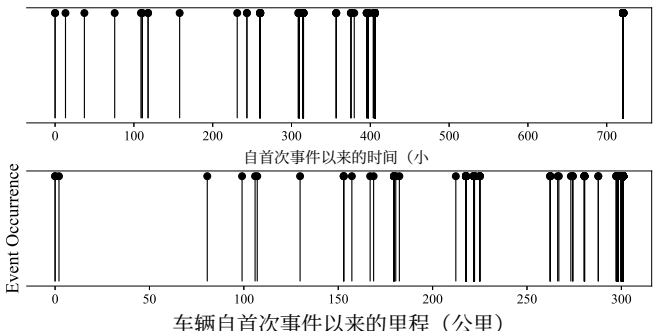


图2. 车辆事件的时空点过程表示。粗竖线表示在相同 t_i 时间 m_i或里程下发生的多个事件。

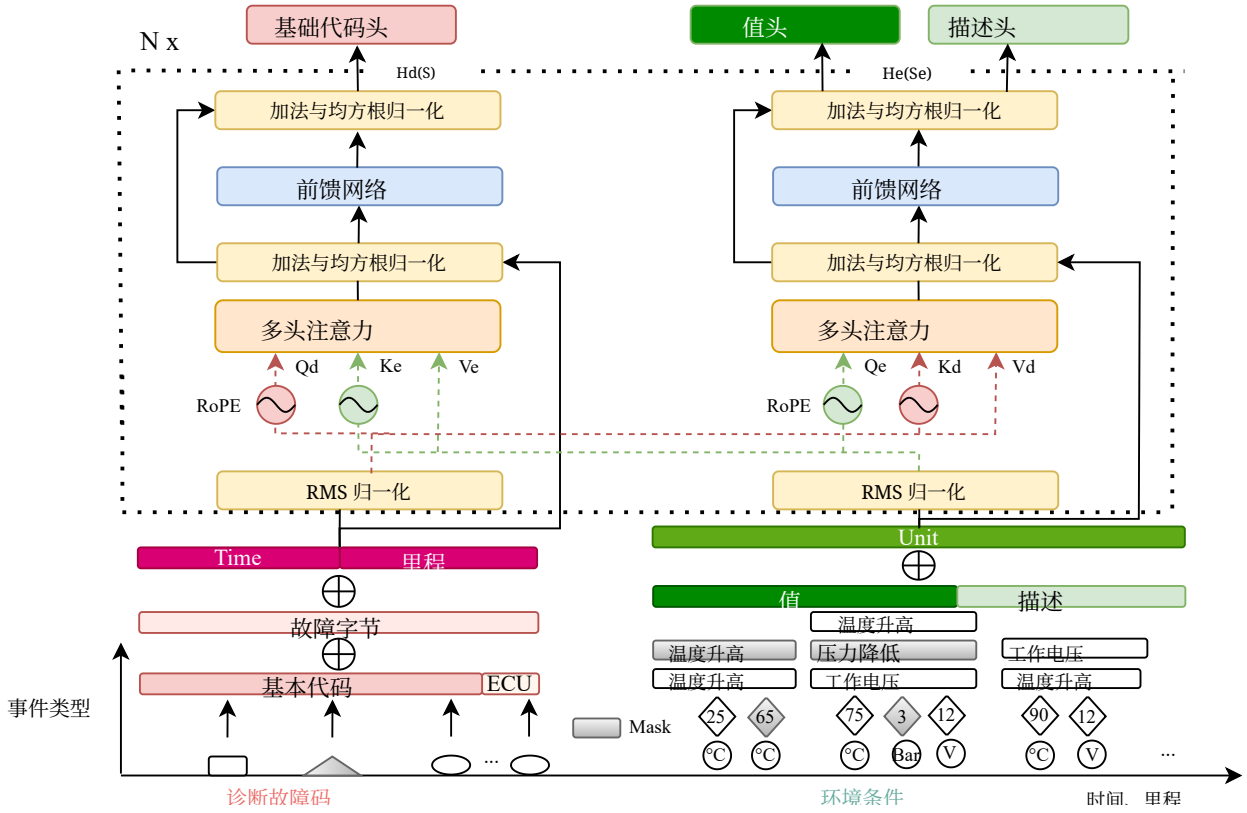


图 3. 采用多模态掩码的 BiCarFormer 整体架构。两个并行Transformer根据各自模态 $\boldsymbol{Q}$， $\boldsymbol{K}$， $\boldsymbol{V}$计算交叉注意力分数。每种模态都会生成两种最终表示：诊断故障码（$\boldsymbol{H}_d$）和 e.条件（$\boldsymbol{H}_e$）。输入层定义了多个嵌入，以考虑词元特定的特征。

基数的嵌入彼此独立地保留，从而捕获电子控制单元、基础 *DTC*和故障字节之间更多的关系。我们得到输入诊断故障码嵌入 $\boldsymbol{D} \in \mathbb{R}^{L\times d}$:

$$\boldsymbol{D} = \text{concat}(\boldsymbol{D}_{ecu}, \boldsymbol{D}_{base}) + \boldsymbol{D}_f$$

在预训练阶段，我们仅掩码基础DTC词元，因为它是主要信息。

V. 位置嵌入

了解故障事件在时间和里程维度上的位置，有助于预测性维护。我们的数据集中，故障事件呈现出空间和时间模式（图 2），这可能表明车辆处于平稳状态，或存在重复性时间故障 [4]。以往的工作（例如 [12],）采用自适应分箱，对长事件序列中的时间进行离散化。然而，鉴于我们的序列长度适中（258 个诊断故障码，间隔为 30 天和 300 千米），分箱并无必要。此外，离散事件建模中常用的查找表嵌入会带来较高的计算开销、引入不必要的参数，而且无法保留 $\mathbb{R}^+$ 中有意义的距离关系。

我们采用连续时间嵌入 [18]。具体而言，给定事件 u_i的绝对时间 t_i 和里程 m_i ，我们将时间嵌入定义为:

$$\mathbf{T}_{i,j} := \begin{cases} \sin(t_i \times \theta_{0,u}^{j/d}) & \text{if } j \mod 2 = 0 \\ \cos(t_i \times \theta_{0,u}^{(j-1)/d}) & \text{if } j \mod 2 = 1 \end{cases} \quad (7)$$

我们沿用公式（7）来创建里程嵌入 $\boldsymbol{M} \in \mathbb{R}^{L\times d/2}$。两个嵌入 $\boldsymbol{T}$， $\boldsymbol{M}$沿特征维度拼接，形成统一位置表示。这样既保留了它们各自的贡献，也能避免与 $\boldsymbol{D}$融合时发生干扰。最终的融合输入 $\boldsymbol{U} \in \mathbb{R}^{L\times d}$定义为:

$$\boldsymbol{U} = \text{concat}(\boldsymbol{D}_{base}, \boldsymbol{D}_{ecu}) + \boldsymbol{D}_f + \text{concat}(\boldsymbol{T}, \boldsymbol{M}) \quad (8)$$

A. 环境嵌入

为了更好地捕捉不同 e.条件 元素之间的关系（d、 v、 u），我们创建了三个不同的可学习嵌入：（1）e.条件的描述 $\boldsymbol{D}_e \in \mathbb{R}^{L^e\times d_d}$ （2）离散化数值 $\boldsymbol{V}_e \in \mathbb{R}^{L^e\times d_v}$ （3）其单位 $\boldsymbol{U}_e \in \mathbb{R}^{L^e\times d}$。我们在输入层面结合使用早期求和与拼接来融合这些嵌入，从而获得整体 e.条件嵌入 $\boldsymbol{E} \in \mathbb{R}^{L^e\times d}$:

$$\boldsymbol{E} = \text{concat}(\boldsymbol{V}_e + \boldsymbol{D}_e) + \boldsymbol{U}_e$$

为了融合 e.条件，我们无法简单地将其与各自的诊断故障码对齐，然后与 $\boldsymbol{D}$进行早期逐标记求和或拼接。由于我们面对的是真实世界的数据（存在噪声和冗余），并且进行了大量的数据过滤，因此每个诊断故障码都缺少过多的 e.条件。因此，我们选择创建一个独立的序列 S_e ，其长度远大于 S，并将所有 e.条件拼接起来，将其融合为一个嵌入 $\boldsymbol{E}$。这种直接的方法

具有灵活性，并且 $\boldsymbol{E}$ 可以在 [20]内部以中间融合的方式应用于注意力机制。

B. 车辆事件序列的协同注意力

BiCarFormer 的整体架构如图3所示。我们的架构直接受到 ViLBERT 的协同注意力机制启发， [32]，该机制通过计算以各模态为条件的注意力分数来实现多模态学习。两个多头注意力层对 S 和S^e进行处理， $\boldsymbol{D}$ （诊断故障码）与 $\boldsymbol{E}$ （e.条件）的融合嵌入分别作为条件，计算出两组注意力分数。我们在两组查询和键（分别用于交叉注意力）上应用 RoPE（旋转位置编码） [33]，以引入词元的绝对位置和相对位置（图3）。

具体而言，来自词元位置 $m \in [1, L], n \in [1, L^e]$ 的融合输入嵌入向量 $\boldsymbol{u}_m \in \mathbb{R}^d$ 和 $\boldsymbol{e}_n \in \mathbb{R}^d$ 通过权重 $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v$ 进行投影。随后，在复数坐标中，查询和键表示为：

$$\boldsymbol{q}_m = e^{im\theta_u} \boldsymbol{W}_q \boldsymbol{u}_m \tag{9}$$

$$\boldsymbol{k}_n = e^{in\theta_e} \boldsymbol{W}_k \boldsymbol{e}_n \tag{10}$$

其中， $\theta_u = \text{diag}(\theta_1, \ldots, \theta_{d/2})$，且 $\theta_i = \theta_{0\,u}^{-2i/d}$，$\theta_{0,u} = 5000$。 θ_e 的处理方式相同，但由于 $L^e \gg L$，其使用 $\theta_{0,e} = 80000$ 。我们使用对齐函数 [34] $f: \mathbb{R} \to [0, 1]$ 来计算两个词元之间的注意力权重：

$$a(\boldsymbol{u}_m, \boldsymbol{e}_n, m, n) = f(\boldsymbol{q}_m^T \boldsymbol{k}_n) = a_{dtc \to env} \tag{11}$$

$$a(\boldsymbol{u}_m, \boldsymbol{e}_n, m, n) = f\left(\boldsymbol{u}_m^T \boldsymbol{W}_q^T e^{i(n\theta_e - m\theta_u)} \boldsymbol{W}_k \boldsymbol{e}_n\right) \tag{12}$$

我们还针对 $a_{env \to dtc} = a(\boldsymbol{e}_m, \boldsymbol{u}_n, n, m)$ 计算公式（12），并为 $\boldsymbol{q}_m$，$\boldsymbol{k}_n$ 使用不同的权重。最后，生成两个交叉注意力上下文向量：

$$\boldsymbol{c}_{dtc \to env}(m) = \sum_{n=1}^{L^e} a_{dtc \to env}(m, n) \boldsymbol{v}_{e,n} \in \mathbb{R}^d \tag{13}$$

$$\boldsymbol{c}_{env \to dtc}(n) = \sum_{m=1}^{L} a_{env \to dtc}(n, m) \boldsymbol{v}_{u,m} \in \mathbb{R}^d \tag{14}$$

其中，值 $\boldsymbol{v}_u$，$\boldsymbol{v}_e$ 取自 $\boldsymbol{V}_e = \boldsymbol{W}_v^e \boldsymbol{E}$，$\boldsymbol{V}_u = \boldsymbol{W}_v^u \boldsymbol{D}$。我们尝试了两种不同的对齐函数 f [34] 来计算公式（12）：广泛使用的 *softmax* 和 *1.5-entmax* [35]。标准的*softmax* 会输出稠密注意力分数，因此会考虑过多的 e.条件。由于输入数据存在噪声且冗余，我们希望提取显著信息，因此稀疏对齐函数似乎是直观的选择。然而，实践中我们发现，在预训练和分类中使用 1.5-entmax 并无益处，且每秒迭代次数降低了 30%。因此，我们仍采用*softmax*。因此，每种模态在经过全连接层、残差连接和均方根归一化 [36], [37]后，会得到两个隐藏状态 $\boldsymbol{H}_d \in \mathbb{R}^{L \times d}$和 $\boldsymbol{H}_e \in \mathbb{R}^{L^e \times d}$ （图3）。

C. 多模态学习

为强化诊断故障码事件与 e.条件之间的关系，我们不仅对诊断故障码进行掩码，也对 e.条件进行掩码，以便

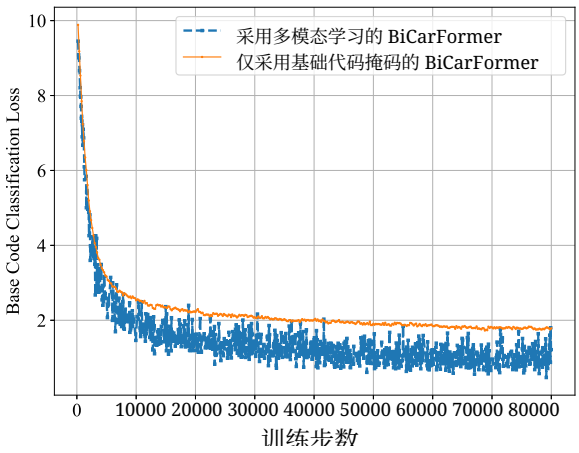


图 4. 采用和未采用多模态学习时，预训练基础DTC分类损失的比较。

(d, v)，并让单位解除掩码，以降低训练复杂度。这通过使用 S 重建 S^e （反之亦然）实现了多模态学习。因此，BiCarFormer 学会利用这一额外模态，这一点可以通过查看基础*DTC* 分类的交叉熵损失来确认： $\mathcal{L}_{\text{dtc}}$使用两种模态有助于降低预训练损失（图 4）。然而，由于多任务学习，损失不太稳定，可能需要进行梯度裁剪并采用较小的学习率来稳定训练。BiCarFormer 使用由静态系数平衡的三种交叉熵损失进行训练： $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{dtc}} + \beta \mathcal{L}_{\text{value}} + \gamma \mathcal{L}_{\text{description}} \tag{15}$$

VI. 实验

设置。我们将模型与成熟的“序列到序列”Transformer 架构进行了比较，包括 BERT [7] （不包含下一句预测任务）和 DTC-TranGRU [2]。由于 DTC-TranGRU 是自回归模型[2]，，我们采用下一词元预测对其进行训练，并使用三个独立的注意力头对三个 DTC 组件进行分类。为了评估各模型的性能，我们重点考察了下游任务中的错误模式多标签分类。对于 BERT 和 BiCarFormer，我们在预训练期间对 15% 的词元进行掩码。不同模型均包含约 2500 万个参数，且隐藏维度相同 $d = 600$，从而确保比较公平。对于下游的多标签序列分类任务，我们使用 [CLS] 词元进行分类；对于 BiCarFormer，则使用 $[CLS]_{dtc}$ 和 $[CLS]_{env}$ ，先沿特征维度 d拼接，再通过一个小型 MLP进行融合。对于 DTC-TranGRU，我们使用 [EOS] 词元，而不是[CLS]。每个分类器都有 3,461,160 个可训练参数，由一个小型 MLP 构成，并配有层归一化、残差连接和 Sigmoid 激活函数。在分类过程中，我们冻结了所有骨干网络。数据集按 75%、15% 和 15% 的比例划分为训练集、验证集和测试集，其中约有 360 个带标签的错误模式。预训练时，我们使用的学习率为

表 III 多标签错误模式分类的下游评估。每个序列到序列模型的参数数量相同（25M）。我们采用不同的平均方式进行更全面的评估。

模型	受试者工作特征曲线下面积（微平均）	F1 分数（微平均）	F1 分数（宏平均）	精确率（样本平均）	召回率（样本平均）	F1 分数（样本平均）
BiCarFormer	**0.809**	**0.77**	**0.71**	**0.68**	**0.62**	**0.64**
DTC-TranGRU [2]	0.602	0.36	0.28	0.23	0.19	0.2
BERT [7]	0.768	0.71	0.63	0.59	0.53	0.55

10^{-4}，所有模型均采用余弦调度器，并预热 2000 步。我们使用 AdamW 优化器 [38]，其中 β_1 设为 0.9，β_2 设为 0.99。我们采用 0.1 的权重衰减，并将梯度裁剪百分位数设为 5。最后，我们在 NVIDIA A10G GPU 上使用半精度浮点数进行训练，预训练的批次大小为 32，分类器的批次大小为 192。

指标。评估针对 360 种不同的错误模式进行。为衡量分类性能，我们采用了多种分类指标，即受试者工作特征曲线下面积（AUROC）、精确率、召回率和 F1 分数指标 [39]。后三项指标使用 0.8 的置信度阈值计算。指标采用不同的平均方式进行汇总：微平均、宏平均和样本平均。微平均在全局范围内计算指标，因此更侧重于频繁出现的类别。它能很好地反映模型在整体数据分布上的表现，但无法反映每个类别或每个实例的表现。相比之下，宏平均对每个类别一视同仁，分别计算各类别的指标，再对结果取平均。然而，在多标签分类中，我们通常更关注每个实例的表现，而非每个类别的表现。这意味着我们优先考虑每个实例的多标签预测整体质量，而不是单独优化各个标签的准确率。为此，样本平均会逐个实例计算指标，然后对所有样本取平均。

A. 多标签分类性能分析

表 III 对 BiCarFormer 与既有序列到序列模型在多标签错误模式分类上的表现进行了对比分析。由于错误模式的复杂程度各不相同，单个实例可能同时关联多个诊断故障码（诊断故障码）及环境条件，因此多标签分类尤其具有挑战性。DTC-TranGRU [2] 表现欠佳，尤其是

在按类别和按实例评估中，表现不佳可能是由于掩码注意力机制和模型架构的局限性。这些限制使模型难以捕获输入序列中的长程依赖，导致受试者工作特征曲线下面积较低（0.602），且 F1 分数表现不佳（微平均：0.36，宏平均：0.28，样本平均：0.2）。尽管 BERT [7]已经取得了相对较高的受试者工作特征曲线下面积和微平均 F1 分数，BiCarFormer 仍显著提升了受试者工作特征曲线下面积（微平均） +4%，并将 F1 分数（微平均）提升了 +6%。

BiCarFormer 的提升在按类别平均时尤为明显。面对许多错误模式时，有些模式由于其固有的复杂性和重叠性，更难以区分，因此可能会令分类器混淆。然而，BiCarFormer 能更好地区分稀有类别，这表明它在多种错误模式上的泛化能力有所提升。这一点在本实验中得到了证实：按类别平均时，我们观察到了分类性能上的差距，F1 分数比 BERT 高出 +8%。这意味着，样本量较少的类别通常仅凭诊断故障码难以分类，但利用 e.条件可以更好地区分。最后，按实例计算的精确率和召回率提升最大：F1 分数、精确率和召回率均比 BERT 高出 +9%。这些结果证实，BiCarFormer 能有效利用这些额外信息，并凸显了与标准“序列到序列”模型相比，其预测能力有所提升。

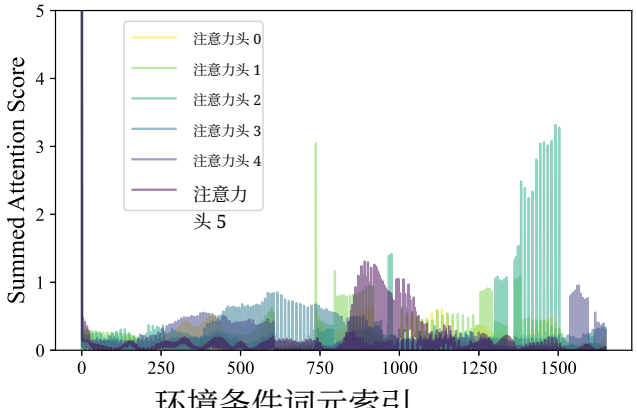


图 5. 各电子控制单元状态从诊断故障码中获得的注意力量。为提高清晰度，纵轴已截断，并打印了注意力头的数量。我们取自最后一层的 $\boldsymbol{A}_{dtc\to env}$。

B. 交叉注意力分数解读

我们希望了解协同注意力机制如何实现多模态学习并提升下游任务的表现。我们随机选取了一个具有特定电池问题的测试样本，并将其输入 BiCarFormer，以分析不同的交叉注意力分数模式。图6展示了 $\boldsymbol{A}_{dtc\to env}$ 的热力图可视化。这有助于我们了解从诊断故障码到 e.条件的一般注意力模式。交叉注意力揭示了某些诊断故障码与特定 e.条件之间的自然关联。以 DTC 190故障码为例，我们可以看到，该词元更多关注一系列局部的 e.条件，其单位为千米/小时，而非其他单位。相比之下，DTC 33故障码和 485 更关注摄氏度。这很合理，因为特定的 e.条件能够体现某些诊断故障码的特征；例如，温度可能对应 DTC1，而 DTC2 可能表示电压升高。由于 S^e 中存在重复的 e. 条件单位，某个诊断故障码可能也会在序列稍后关注下一个 e.条件“电压”，从而在图6中形成这些线状模式，涉及 DTC 190、33 和 485。因此，我们注意到

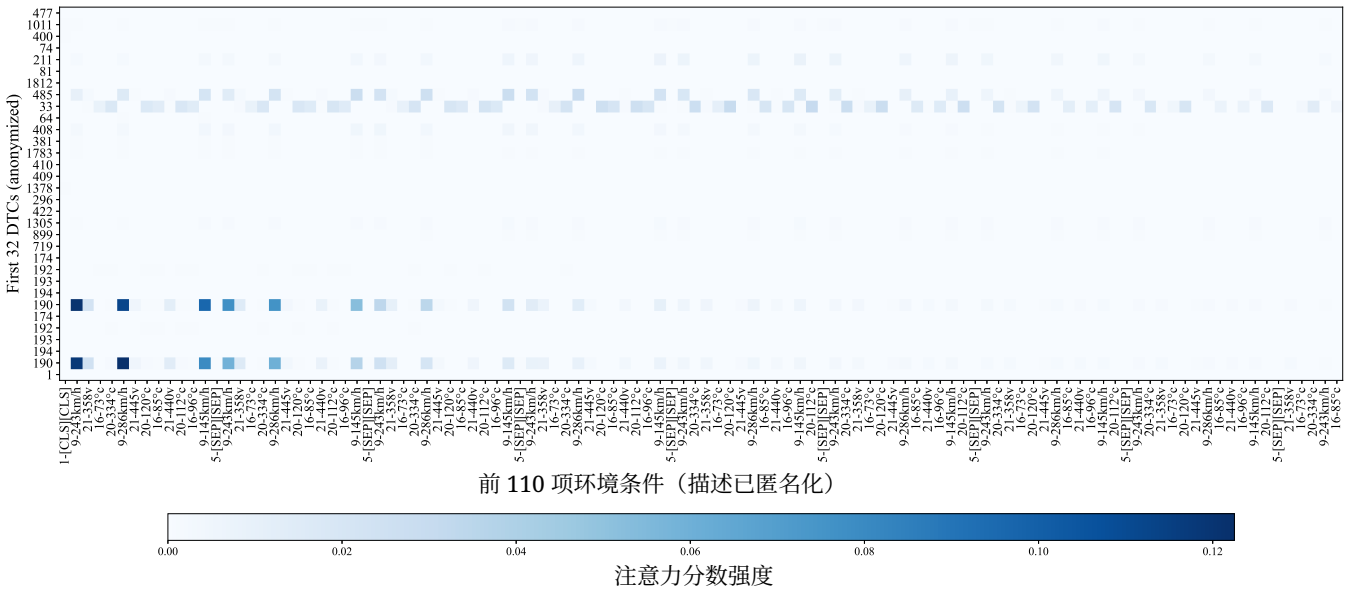


图 6. $\boldsymbol{A}_{dtc\to env}$ 的交叉注意力分数。纵轴显示诊断故障码（已匿名化），横轴显示拼接后的环境条件及其 3 个元素（d，v，u）（描述 d 已匿名化）。每个单元格的深浅反映注意力权重，颜色越深表示注意力值越高。

诊断故障码与其周围 e.条件之间存在局部时间模式（图6）。因此，初始诊断故障码应优先关注 Se 中最初的 e.条件，但由于 L 和 Le 的序列长度不同，会产生偏移。例如，位置 1 和 6 处的 DTC 190故障码应优先关注最初的 e.条件。

我们希望评估诊断故障码是否会关注少数几个关键的 e.条件。图 5 展示了来自同一测试样本的诊断故障码对各个 e.条件分配的注意力大小。值得注意的是，我们观察到每个注意力头对应不同的簇。每个注意力头都会学习关注特定的 e.条件，有时还会关注特定索引，例如注意力头 *1* ($i \approx 749$ 和 800)，以及注意力头 *2*($i \approx 960$)。传统上， [CLS] 词元会受到大量关注。有趣的是，序列末尾部分似乎更受特定注意力头关注： $i \in [1300, 1500]$，这表明即使序列长度相差很大（即 $L^e \gg L$），我们也能捕捉诊断故障码与 e.条件词元之间的长程依赖。此外，靠后的 e.条件可能包含大量缺陷信号，这也合情合理，因为我们正逐渐接近时间 $t_i = t_L$发生的关键故障。因此，可以观察到注意力分数有所上升。

C. 电池电动汽车中的错误模式

随着电池电动汽车 （电池电动汽车）日益普及，对这类车辆进行故障诊断的需求也日益迫切。我们以一种具体的错误模式为例，例如电池老化，这是电动汽车中常见的现象。并绘制与之前相同的测试样本中 e. 条件单元的累积注意力。由于注意力头会关注 S^e 的某个部分（图 5），有些注意力头应会关注特定单位（图 8），例如千米/小时（注意力头 *0*、*4*、*9*）；而其他注意力头则关注摄氏度（注意力头 *6*、*7*）和电压（注意力头 *1*、*2*）。更有意思的是，我们选取一个倾向于专门关注某个单位的注意力头，例如注意力头 *2* ，用于电压，并绘制 S^e 中离散化电压值随其索引变化的情况，如图 7 所示。因此，我们主要试图回答：电压值是否存在波动，以及注意力模式是否随之发生相关变化。我们还绘制由诊断故障码产生的相应注意力分数，并找出多个触发点，在这些触发点上，电压值与注意力分数呈强相关。我们注意到，这种情况因注意力头而异：有些注意力头呈现注意力分数指数递减的形态，而另一些则呈现噪声信号。因此，BiCarFormer 能够从词元化输入中提取连续电压值的特定波动。由此，可以提取触发点，并分析这些特定变化如何影响车辆，从而为领域专家提供更具可解释性的结果。

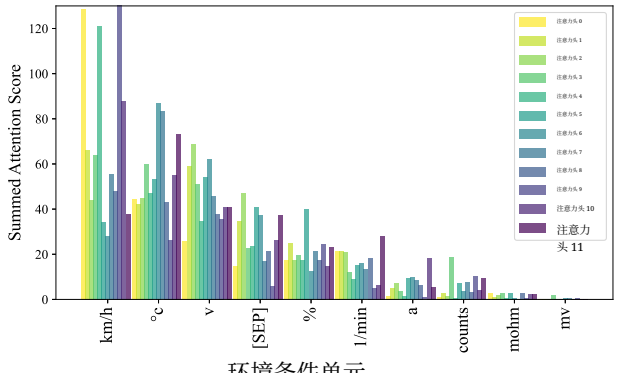


图 7. 在电池电动汽车的电池老化 错误模式中，各环境单元从诊断故障码接收到的注意力量。我们取最后一层的 $\boldsymbol{A}_{dtc\to env}$ 并打印所有注意力头。

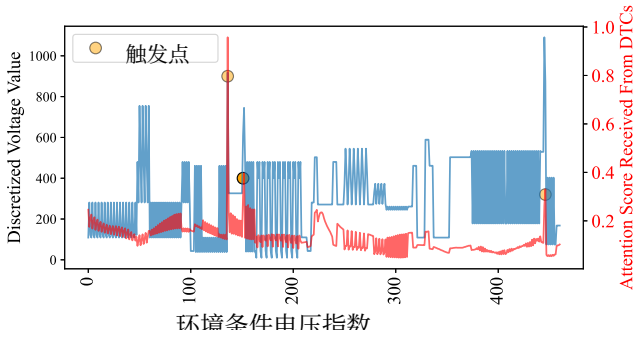


图8. 作为环境条件词元函数的离散化电压变化，以及从诊断故障码接收到的相关注意力大小。我们仅从 S^e 中选取 $u =$ 为"v"的三元组 (d, v, u)，并从最后一层和注意力头2中提取 $\boldsymbol{A}_{dtc \to env}$。该样本取自电池老化错误模式。

VII. 应用型下游任务

通过为 e.条件设置单独的隐藏状态，也可以将其用于无监督学习，因为我们可能不知道某些错误模式，这是一种常见现象。因此，不必只依赖 H_d，还可以利用 H_e，实现更准确的降维，并通过聚类技术观察未标记错误模式之间更清晰的决策边界。此外，如果提供领域知识，我们可以预先直接筛选 S^e，以选择特定的 e.条件，而不是注入大量信息。我们的模型和整体范式使保留这一架构成为可能，并可通过将领域知识直接注入预处理步骤来降低计算量，从而便于比较结果。另一个可能有用的下游应用是可解释性，即了解 e.条件和诊断故障码对错误模式分类的贡献。由于错误模式是由领域专家硬编码定义的规则，我们可以尝试基于特征归因方法[40],交叉注意力分数、基于扰动的方法，或使用 Transformer 进行多标签因果发现，为未知错误模式推导新规则 [41]。与 [4]，相反，BiCarFormer 模型会在事后对错误模式进行分类，因此可能无需优化推理，因为我们不依赖车辆内的边缘计算能力。不过，由于交叉注意力分数具有二次时间复杂度，因此亟需优化公式 (13) 和 (14)。对于事件序列中的协同注意力，应采用稀疏注意力 [26], [27],[30], 、令牌合并 [42] 或池化等替代方法。

VIII. 结论

在本研究中，我们引入了 BiCarFormer，这是一种新型多模态 Transformer 模型，旨在捕捉现代车辆中诊断故障码（诊断故障码）与环境条件之间的复杂关系。通过协同注意力机制，我们展示了 BiCarFormer 如何有效学习错误代码与其环境条件之间的跨模态交互，从而提升分类性能，并生成更易于解释的注意力模式。通过解读注意力分数，领域专家可以理解 BiCarFormer 的预测结果，并验证或推导出新的错误模式规则。该模型在每类和每实例 评估中的优越性能，凸显了其在诊断应用中实际部署的潜力，因为这些应用对准确的多标签预测至关重要。除分类之外，BiCarFormer 还为错误模式的无监督异常检测、可解释性以及领域知识导向模型提供了有前景的研究方向。未来的工作可以探索更高效的协同注意力机制，以处理较长的环境条件序列。

参考文献

[1] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、L. u. Kaiser 和 I. Polosukhin，"Attention is all you need"，载于《神经信息处理系统进展》，I. Guyon、U. V. Luxburg、S. Bengio、H. Wallach、R. Fergus、S. Vishwanathan 和 R. Garnett 编，第30卷。Curran Associates, Inc.，2017年。[2] A. B. Hafeez、E. Alonso 和 A. Riaz，"DTC-TranGRU：利用Transformer和GRU提升下一诊断故障代码预测模型的性能"，《第39届ACM/SIGAPP应用计算研讨会论文集》，2024年。[在线]。获取地址：https://api.semanticscholar.org/ CorpusID:269951398[3] A. B. Hafeez、E. Alonso 和 A. Ter-Sarkisov，"面向车辆预测性维护的序列多变量故障预测"，载于《2021年第20届IEEE机器学习与应用国际会议（ICMLA）》，2021年，第1016–1021页。[4] H. Math、R. Lienhart 和 R. Sch¨on，"利用车辆事件传感数据预测错误模式：一种语言模型方法"，2024年，即将发表于AAAI2025论文集。 [在线]。获取地址：https://arxiv.org/abs/2412 13041 。H. Math、R. Schon 和 R. Lienhart，"高维[5] ¨事件序列中的单次多标签因果发现"，载于《NeurIPS 2025 CauScien研讨会：揭示科学中的因果关系》，2025年。[6] [ ] = 7 8 2 在线。获取地址：https://openreview.net/forum?id z NT vGWC H. Math 和 R. Lienhart，"通过单次图聚合实现高维事件序列中实用的多标签因果发现"，载于《NeurIPS 2025结构化概率推断与生成建模研讨会》，2025年。 [在线]。获取地址：https: //openreview.net/forum?id=lHZfpuDVeW [7] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova，"BERT：用于语言理解的深度双向Transformer预训练"，载于《计算语言学协会北美分会》，2019年。 [在线]。获取地址：https://api.semanticscholar.org/CorpusID: 52967399[8] P. Zhao、M. Kurihara、J. Tanaka、T. Noda、S. Chikuma 和 T. Suzuki，"用于预测性维护的先进基于相关性的异常检测方法"，载于《2017年IEEE预测与健康管理国际会议（ICPHM）》，2017年，第78–83页。[9] H. Langseth 和 L. Portinale，"可靠性领域中的贝叶斯网络"，《可靠性工程与系统安全》，第92卷，第1期，第92–108页，2007年。 [在线]。获取地址：https://www.sciencedirect.com/science/ article/pii/S0951832005001985[10] R. Sipos、D. Fradkin、F. Moerchen 和 Z. Wang，"基于日志的预测性维护"，载于《第20届ACM SIGKDD知识发现与数据挖掘国际会议论文集》，丛书名KDD '14。美国纽约州纽约市：计算机协会，2014年，第1867–1876页。 [在线]。获取地址：https://doi.org/10.1145/2623330. 2623340[11] R. He、Z. Tian 和 M. Zuo，"基于状态空间和神经网络建模的变化运行条件下机器预后"，《机械系统与信号处理》，第182卷，文章编号109598，2023年。[在线]。获取地址：https://www.sciencedirect.com/science/article/pii/ S0888327022006896[12] Y. Zhou、Y. Wang、H. Song、T. Meharizghi、M. A. Jazaery、P. Xu、D. Colin-Magana 和 A. Abeyakoon，"面向基于事件日志的预测性维护的深度序列建模"，载于《KDD 2023时间序列挖掘与学习国际研讨会（MileTS）》，2023年。 [在线]。获取地址：https://www.amazon.science/publications/deep-sequence-modeling-for-event-log-based-predictive-maintenance [13] A. Labach、A. Pokhrel、X. S. Huang、S. Zuberi、S. E. Yi、M. Volkovs、T. Poutanen 和 R. G. Krishnan，"Duett：面向电子健康记录的双事件时间Transformer"，载于《第8届医疗保健机器学习会议论文集》，丛书名《机器学习研究论文集》，K. Deshpande、M. Fiterau、S. Joshi,

Z. Lipton、R. Ranganath、I. Urteaga 和 S. Yeung 编，第219卷。PMLR，2023年8月11日至12日，第403–422页。 [在线]。获取地址：https://proceedings.mlr.press/v219/labach23a.html [14] L. Rasmy、Y. Xiang、Z. Xie、C. Tao 和 D. Zhi，“Med-BERT：基于大规模结构化电子健康记录的预训练上下文嵌入，用于疾病预测”，NPJ Digit Med.，2021年5月20日；4(1):86，第abs/2005.12833卷，2020年。[15] L. D. Manocchio、S. Layeghy、W. W. Lo、G. K. Kulatilleke、M. Sarhan 和 M. Portmann，“FlowTransformer：面向基于流的网络入侵检测系统的Transformer框架”，《Expert Systems with Applications》，第241卷，文章编号122564，2024年。 [在线]。获取地址：https://www.sciencedirect.com/science/article/pii/S095741742303066X [16] A. G. Hawkes，“若干自激和互激点过程的谱”，《Biometrika》，第58卷，第1期，第83–90页，1971年4月。 [ ] 10 1093 58 1 83 [17] 在线。获取地址：https://doi.org/ . /biomet/ . . N. Du、H. Dai、R. Trivedi、U. Upadhyay、M. Gomez-Rodriguez 和 L. Song，“循环标记时间点过程：将事件历史嵌入为向量”，载于《第22届ACM SIGKDD知识发现与数据挖掘国际会议论文集》，丛书名KDD ’16。美国纽约州纽约市：计算机协会，2016年，第1555–1564页。 [在线]。获取地址：https://doi.org/10. 1145/2939672.2939875 [18] Q. Zhang、A. Lipani、O. Kirnap 和 E. Yilmaz，“自注意力Hawkes过程”，载于《第37届国际机器学习会议论文集》，丛书名《机器学习研究论文集》，H. D. III 和 A. Singh 编，第119卷。PMLR，2020年7月13日至18日，第11 183–11 193页。 [在线]。获取地址：https: //proceedings.mlr.press/v119/zhang20q.html [19] X. Shou、D. Subramanian、D. Bhattacharjya、T. Gao 和 K. P. Bennet，“面向多变量点过程的自监督对比预训练”，arXiv，第abs/2402.00987卷，2024年。 [在线]。获取地址：https://api.semanticscholar.org/CorpusID:267406624 [20] P. Xu、X. Zhu 和 D. A. Clifton，“基于Transformer的多模态学习：综述”，《IEEE模式分析与机器智能汇刊》，第45卷，第10期，第12113–12132页，2023年10月。 [在线]。获取地址：https://doi.org/ 10 1109 2023 3275156 . /TPAMI. . A. Nagrani、S. Yang、A. Arnab、A. Jansen、C. Schmid 和 C. Sun，“用于多模态融合的注意力瓶颈”，载于《第35届神经信息处理系统会议论文集》，丛书名NIPS ’21。美国纽约州雷德胡克：Curran Associates Inc.，2024年。[22] B. Shickel、B. Silva、T. Ozrazgat-Baslanti、Y. Ren、K. Khezeli、Z. Guan、P. J. Tighe、A. Bihorac 和 P. Rashidi，“通过纵向电子健康记录标记化和灵活的Transformer网络进行多维患者病情严重程度估计”，《Frontiers in Digital Health》，第4卷，2022年。 [ ] 10401752 在线。获取地址：https://par.nsf.gov/biblio/ A. Radford、J. W. Kim、C. Hallacy、A. Ramesh、G. Goh、S. Agarwal、G. Sastry、A. Askell、P. Mishkin、J. Clark、G. Krueger 和 I. Sutskever，“通过自然语言监督学习可迁移视觉模型”，载于《第38届国际机器学习会议论文集》，丛书名《机器学习研究论文集》，M. Meila 和 T. Zhang 编，第139卷。PMLR，2021年7月18日至24日，第8748–8763页。 [在线]。获取地址：https://proceedings.mlr.press/v139/radford21a.html [24] J. Lu、D. Batra、D. Parikh 和 S. Lee，《ViLBERT：为视觉与语言任务预训练任务无关的视觉语言表示》。美国纽约州雷德胡克：Curran Associates Inc.，2019年。[25] H. H. Tan 和 M. Bansal，“LXMERT：从Transformer中学习跨模态编码器表示”，载于《自然语言处理实证方法会议》，2019年。 [在线]。获取地址：https://api.semanticscholar.org/CorpusID:201103729 [26] I. Beltagy、M. E. Peters 和 A. Cohan，“LongFormer：长文档Transformer”，2020年。 [在线]。获取地址：https://arxiv.org/abs/ 2004.05150 [27] M. Zaheer、G. Guruganesh、K. A. Dubey、J. Ainslie、C. Alberti、S. On-tanon、P. Pham、A. Ravula、Q. Wang、L. Yang 和 A. Ahmed，“BigBird：面向更长序列的Transformer”，载于《神经信息处理系统进展》，H. Larochelle、M. Ranzato、R. Hadsell、M. Balcan 和 H. Lin 编，第33卷。Curran Associates, Inc.，2020年，第17 283–17 297页。[28] S. Wang、B. Li、M. Khabsa、H. Fang 和 H. Ma，“Linformer：具有线性复杂度的自注意力”，2020年，引用arXiv:2006.04768。 [ ] 2006 04768 [29] 在线。获取地址：http://arxiv.org/abs/ . S. Zuo、H. Jiang、Z. Li、T. Zhao 和 H. Zha，“Transformer Hawkes过程”，载于《第37届国际机器学习会议论文集》，丛书名《机器学习研究论文集》，H. D. III 和 A. Singh 编，第119卷。PMLR，2020年7月13日至18日，第11 692–11 702页。 [在线]。可访问：https: //proceedings.mlr.press/v119/zuo20a.html [30] Z. Li 和 M. Sun，“面向长事件序列的稀疏 Transformer 霍克斯过程”，载于《数据库中的机器学习与知识发现：研究赛道》，D. Koutra、C. Plant、M. Gomez Rodriguez、E. Baralis 和 F. Bonchi 编。Cham：Springer Nature Switzerland，2023年，第172–188页。[31] M. Greenwald 和 S. Khanna，“分位数摘要的节省空间的在线计算”，SIGMOD Rec.，第30卷，第2期，第58–66页，2009年5月。 2001 [ ] 10 1145 376284 375670 在线。可访问：https://doi.org/ 。/ 。J. Lu、D. Batra、D. Parikh 和 S. Lee，《ViLBERT：面向视觉与语言任务的任务无关视觉语言表示预训练》。美国纽约州雷德胡克：Curran Associates Inc.，2019年。[33] J. Su、M. Ahmed、Y. Lu、S. Pan、W. Bo 和 Y. Liu，“Roformer：采用旋转位置嵌入的增强型 Transformer”，Neurocomput.，第568卷，C期，2024年3月。 [在线]。可访问：https://doi.org/10.1016/ j.neucom. 2023 127063 。G. Brauwers 和 F. Frasincar，“深度学习中注意力机制的综合综述”，IEEE Trans. on Knowl. and Data Eng.，第35卷，第4期，第3279–3298页，2023年4月。 [在线]。可访问：https://doi.org/ 10 1109 2021 3126456 。/TKDE. 。B. Peters、V. Niculae 和 A. F. T. Martins，“稀疏序列到序列模型”，载于《第57届计算语言学协会年会论文集》，A. Korhonen、D. Traum 和 L. M`arquez 编。意大利佛罗伦萨：计算语言学协会，2019年7月，第1504–1519页。 [在线]。可访问：https://aclanthology.org/P19-1146/ [36] B. Zhang 和 R. Sennrich，《均方根层归一化》。美国纽约州雷德胡克：Curran Associates Inc.，2019年。[37] H. Touvron、T. Lavril、G. Izacard、X. Martinet、M.-A. Lachaux、T. Lacroix、B. Rozi`ere、N. Goyal、E. Hambro、F. Azhar、A. Rodriguez、A. Joulin、E. Grave 和 G. Lample，“Llama：开放且高效的基础语言模型”，2023年。 [在线]。可访问：https: //arxiv.org/abs/ 2302 13971 。I. Loshchilov 和 F. Hutter，“解耦权重衰减正则化”，载于《国际学习表征会议》，2019年。 [在线]。可访问：https://openreview.net/forum?id = 6 7 Bkg RiCqY [38] J. Opitz，“深入探讨分类评估指标并批判性反思常见评估实践”，《计算语言学协会汇刊》，第 12, 820 836, 06 卷，第–页。在线。2024 [ ] 10 1162 \ 可访问：https://doi.org/ 。/tacl a \ 00675 [39] S. M. Lundberg 和 S.-I. Lee，“一种解释模型预测的统一方法”，载于《第31届神经信息处理系统国际会议论文集》，NIPS’17系列。美国纽约州雷德胡克：Curran Associates Inc.，2017年，第4768–4777页。[41] N. I. Alonso 等，“用于因果推断的 Transformer”，《用于因果推断的 Transformer》（2024年12月5日），2024年。[42] D. Bolya、C.-Y. Fu、X. Dai、P. Zhang、C. Feichtenhofer 和 J. Hoffman，“令牌合并：让你的视觉 Transformer 更快”，载于《第十一届国际学习表征会议》，2023年。 [在线]。可访问：https://openreview.net/forum?id=JroZRaRw7Eu