# 基于规则的错误检测与校正以实现运动轨迹分类

Bowen Xi[1], Kevin Scaria[1], Divyagna Bavikadi[2] and Paulo Shakarian[2]

[1]Arizona State University

[2]Syracuse University

{bowenxi, kscaria}@asu.edu, {dbavikad, pshakari}@syr.edu



Abstract

运动轨迹分类在交通运输中有许多应用，是大规模运动轨迹生成和异常检测的关键组成部分，在具有未知运动类型的环境中具有关键的安全应用。然而，目前最先进的技术基于监督深度学习——这使得它们在遇到新型未见过的类别时面临挑战。我们提供了一个神经符号规则框架来进行这些模型的错误修正和检测，并将其整合到我们的运动轨迹平台中。我们在几个最近的最先进模型上进行了一系列实验，展示了高度准确的错误检测能力、提高包含训练集中未见的新类型运动的测试数据的准确性以及基础用例的准确性提升，此外还有一系列指导算法开发的理论属性。具体来说，我们展示了预测误差的 F1 分数高达 0.984，对于未见过的运动类型的准确性有显著性能提升（比最先进模型在零样本准确性上提高了 8.51%），并且比最先进的模型有更好的准确性。

## 1 介绍

对带有时间戳的全球定位系统（GPS）序列，即所谓的“移动轨迹”的旅行模式识别，在出行需求分析[Huang et al., 2019]、交通规划[Lin and Hsu, 2014]和海运船只运动分析[Fikioris et al., 2023]等领域有重要应用。最近，这个问题在安全应用方面引起了关注，例如导致了如 IARPA HAYSTAC 项目 [1] 的努力，我们为此创建并部署了一个轨迹分析平台。该问题的一个关键方面是通过运动类型对轨迹进行正确分类——特别是在外部冲击（如自然灾害）发生后尤为如此。然而，目前的最新技术依赖于监督神经模型[Kim et al., 2022; Dabiri and Heaslip, 2018]，这些模型已被证明表现良好，但在暴露于先前未见过的数据时可能会失败，特别是对于之前未识别出的运动类型。本文扩展了当前的监督神经方法，并引入了一个轻量级错误检测与纠正规则（EDCR）框架，提供了一种整体的神经符号系统。该框架进一步支持关键技术，尤其是运输领域的人工智能，在这个领域通常会遇到未曾见过的数据并要求模型不将其误分类。核心直觉是使用训练和操作数据来学习可以预测并修正监督模型中错误的规则。一旦训练完成，这些规则在操作中分两个阶段运用：首先检测规则识别出潜在被误分类的移动轨迹。然后使用另一种类型的规则（“纠正规则”）重新归类轨迹，将样本重新分配到新的类别中。我们提出了一种基于逻辑和规则挖掘的强大的理论框架，并正式证明了与学习规则相关的量（如置信度和支持度）如何与诸如精确率和召回率等分类级别的机器学习指标的变化相关。为了实证地展示有效性，我们提供了一系列实验，表明该框架在检测错误方面非常有效（SOTA 模型的错误检测 F1 值为 0.875，基于检查的模型可达 0.984），零样本调整下的未见过的动作准确性比 SOTA 高出 8.51%，以及标准分类准确性的提高超过了 SOTA 模型。接下来，我们将进一步介绍我们领域问题的背景和当前轨迹分析平台（其中部分内容是对[Bavikadi et al., 2024]的回顾），引入 EDCR 的算法框架及其理论性质，并提供我们的实验结果套件，在此之后总结我们的发现及未来工作。

## 2 背景

**总体概念和部署系统**。在地面真实数据中通常不包含的运动类型会随着某些目标环境的出现而变得显著（例如，在某些城市区域中的付费滑板车、南亚地区的自动三轮车或威尼斯的船只）。因此，IARPA（情报高级研究项目活动）已将表征和生成正常运动的问题识别为 HAYSTAC 计划中的关键问题。在这里，目标是在细粒度层面上建立人类正常运动模型，并在政府环境中部署系统进行评估时操作化这些模型和技术。作为该计划的执行者，[Bavikadi et al., 2024] 考察生成现实运动轨迹

[1]https://www.iarpa.gov/研究计划/haystac

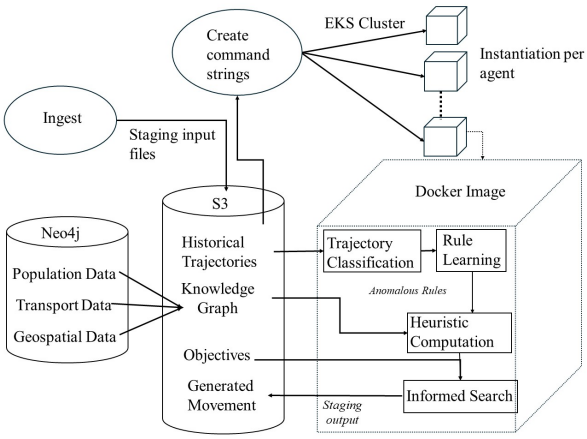


图 1: 整体系统部署用于政府测试

的问题。最初针对轨迹生成的政府测试仅涉及单一交通方式的运动轨迹。然而，在为过渡到实际应用做准备的过程中，政府设置了分析来自各种运动类型的轨迹的要求——其中交通方式未知。因此，我们计划操作化一个运动轨迹分类模块，该模块我们在部署的基于云的架构上下文中进行了描述，如图 1 所示。此管道与政府系统接口以访问各个地理位置相关的原始地理空间数据、历史代理轨迹及其相应的目标文件。我们的初始摄取和容器化过程在一个有向无环图（DAG）中作为节点保持。我们的摄取机制首先解析给定代理的历史轨迹，然后将它们放置在 S3 存储桶中进行暂存。接着，在 Neo4j 中存储的地理空间数据被整合为知识图谱，并暂存在 S3 存储桶中。我们在 Amazon Elastic Kubernetes 服务（EKS）集群上通过 Docker 镜像实例化所有代理的 pod，以分析暂存文件夹并创建特定于每个代理的具体字符串命令。轨迹分类模块识别并在相应的轨迹中标记交通方式，这进一步用于在考虑不同类型的运动时学习规则。这些规则连同知识图谱一起被用来计算启发式值，以便使用有信息搜索方法（A*搜索）生成运动轨迹 [Bavikadi et al., 2024]。随着容器运行，生成的运动指令文件会被推送到适当的输出目录中，如图 1 所示。此外，生成的运动遵循预定义的空间时间约束（目标）。

**运动轨迹分类问题**。运动轨迹分类问题已在文献中被研究过 [Zheng et al., 2008; Wang et al., 2017; Simoncini et al., 2018; Dabiri and Heaslip, 2018; Kim et al., 2022]，我们将它称为运动轨迹分类问题（MTCP）。我们还注意到，这项工作与轨迹生成 [Bavikadi et al., 2024; Janner et al., 2021; Chen et al., 2021; Itkina and Kochenderfer, 2022]不同且互补，后者并不寻求识别交通工具类型。一个 MTCP 实例被定义为给定一系列 GPS 点，$\omega$，分配一个来自 $\mathcal{C}$ 的 $n$ 运动类别，这通常被定义为[Kim et al., 2022; Dabiri and Heaslip, 2018]$\mathcal{C} = \{\text{walk}, \text{bike}, \text{bus}, \text{drive}, \text{train}\}$。目前解决 MTCP 问题的方法是创建一个神经模型 $f_\theta$，该模型使用一组权重 $\theta$ 将序列映射到运动类别。在此方法中，使用传统的方法（即梯度下降）来寻找一组参数，使得基于某个训练集 $\mathcal{T}$ 的损失函数最小化（其中每个样本 $\omega \in \mathcal{T}$ 与一个真实类别 $gt(\omega)$ 相关联）。形式上：$\arg\min_\theta \mathbb{E}_{\omega\in\mathcal{T}}\text{Loss}(f_\theta(\omega), gt(\omega))$。在此范式下，提出了几种方法。最值得注意的是基于 CNN 的架构[Dabiri and Heaslip, 2018]和当前的最新方法称为长时递归卷积网络（LRCN）[Kim et al., 2022]，该方法结合了较低的 CNN 层和较高的 LSTM 层——我们将其用作基线，同时还扩展了 LRCN 并使用额外的注意力头（LRCNa）。

**当前 SOTA 的局限性**。然而，这些方法在 IARPA HAYSTAC 用例的背景下存在一些问题。

- *不适用于未见过的动作*。任何监督下的 MTCP 模型都需要一个其运动类别与目标环境相匹配的数据集。为了应对政府客户更动态的需求，我们需要能够识别出何时可能给出错误结果的方法，以适应新环境。
- *所有类先验已知*。在先前的工作中，集合 $\mathcal{C}$ 被视为静态和完整的，这意味着训练集中未出现的新动作类型将无法被正确分类，并且不会被视为与类 $\mathcal{C}$ 中的动作类型不同。
- *先前结果在重叠的训练和测试集上进行评估*。正如在[Zeng et al., 2023]中所指出的，MTCP 方法的标准评估是在存在训练和测试之间泄漏的数据集上进行的。由于这项工作的操作性质，我们必须检查其他划分。

将运动轨迹分类模型部署到特定环境中可能导致未在训练中见到的运动（例如，在训练中没有出现电动滑板车，但在某些城市区域非常普遍）。因此，这些“新颖运动”本质上会被错误分类。所有这些限制的共同点是对何时这类分类器可能出错的理解。然而，这超出了重新训练或从不同训练数据中选择的范围，因为政府客户设想的用例包括未见过的运动类型——因此训练数据会受到限制。这通常排除了元学习和领域泛化 [Hospedales et al., 2021; Zhou et al., 2022; Vanschoren, 2018; Maes and Nardi, 1988]，这些方法试图考虑到数据分布的变化以及/或者选择一个在与当前问题类似的数据上进行过训练的模型。此外，这些问题还必须在我们现有的系统（图 1）背景下解决，该系统采用符号推理生成运动轨迹——确保它们达到一定程度的正常性 [Bavikadi et al., 2024]。因此，我们考察了表征机器学习模型失败的方法，例如内省 [Daftry et al., 2016; Ramanagopal et al., 2018]，然而这些方法仅预测模型失效，并不试图解释或纠正它。另

一个相关领域是机器学习验证，它 [Ivanov et al., 2021; Jothimurugan et al., 2021; Ma et al., 2020]旨在确保机器学习模型的输出满足逻辑规范——然而，到目前为止，这项工作尚未应用于纠正机器学习模型的输出，并且通常依赖于预先知道逻辑规范（这与我们的使用场景不符）。

最近关于溯因学习[Huang et al., 2023; Dai et al., 2019]和神经符号推理[Cornelio et al., 2022]的研究中，整合了基于与领域知识不一致性的错误纠正机制作为逻辑规则——但与验证一样，我们没有预先知道这些符号知识。

## 3 错误检测和纠正规则

为了解决上一节中的问题，我们正在采用一种基于规则的方法来纠正 $f_\theta$ 模型。直觉是通过有限的数据，我们将学习一组规则（表示为 $\Pi$），这组规则将能够通过逻辑推理 [Aditya et al., 2023]检测并修正 $f_\theta$ 的错误。然后，在部署处理某个新的序列 $\omega$ 时，我们会首先计算类别 $f_\theta(\omega)$，然后使用集合 $\Pi$ 中的规则来判断是否接受 $f_\theta$ 的结果，并且如果不接受，则提供一个替代类别以尝试纠正错误。在本节中，我们用简单的谓词逻辑（FOL）形式化了纠错框架，并提供了分析结果，这些结果涉及所学规则的不同方面，指导我们采用分析方法学习这样的错误检测和纠正规则。我们在本节的最后讨论如何提取各种潜在的“故障条件”来创建用于纠正错误的规则。

在这篇论文中，我们将假设一个操作序列集 $\mathcal{O}$，在模型训练后可以获取其真实值。这个集合可以是训练数据集、子集或超集。后来，在我们的实验中，我们研究了 $\mathcal{O} = \mathcal{T}$ 和 $\mathcal{T} \subseteq \mathcal{O}$ 的情况——然而这些并不是必需的，因为我们的结果基于模型在 $\mathcal{O}$ 上的表现——并且我们设想使用场景中 $\mathcal{O}$ 与 $\mathcal{T}$ 有显著差异。在这些样本中，对于每个类别 $i$，模型 ($f_\theta$) 返回类别 $i$ 给 $N_i$ 的样本，并且对于每个类别 $i$，我们有真正例的数量 $TP_i$，假正例的数量 $FP_i$，真负例的数量 $TN_i$，和假负例的数量 $FN_i$。我们有精度 $P_i = TP_i/N_i = TP_i/(TP_i + FP_i)$，召回率 $R_i = TP_i/(TP_i + FN_i)$，以及预测类别 $i$ 的先验概率：$\boldsymbol{\mathcal{P}}_i = N_i/N$。

**语言**。我们使用一种简单的谓词逻辑语言，其中样本由常量符号（$\omega$）表示。我们定义了与每个样本相关的“条件”一元谓词集合 $Cmcond_1, \ldots, cond_m$，这些谓词可以为真或假——这些条件可以被认为是可能导致失败的条件（但我们的学习算法将确定哪些条件会导致给定预测的失败）。我们还为每个类 $i$ 定义了一元谓词：下面定义的 $pred_i$、$corr_i$ 和 $error$。

- $pred_i$：模型预测类别为 $i$ 的充要条件，即 $pred_i(\omega)$ 为真的充要条件是 $f_\theta(\omega) = i$。
- $corr_i$：当且仅当 $\omega$ 的正确运动类别是 $i$ 时，即 $corr_i(\omega)$ 为真当且仅当 $gt(\omega) = i$。
- $error$：当且仅当一个 EDCR 规则得出模型的预测存在错误时，这个谓词为真。

**规则**。规则集 $\Pi$ 将包括每类的两条规则：一条“错误检测”和一条“错误纠正”。错误检测规则将确定由 $f_\theta$ 所做的预测是否无效。本质上，我们可以将这样的规则视为更改 $f_\theta$ 为样本 $\omega$ 分配的动作类别从 $i$ 到“未知”。对于给定的类 $i$，我们将有一个相关的检测条件集 $DC_i$，它是条件的一个子集，这些条件的析取用于确定 $f_\theta$ 是否给出了错误分类。

$$error(\omega) \leftarrow pred_i(\omega) \wedge \bigvee_{j \in DC_i} cond_j(\omega) \tag{1}$$

在应用每一类的错误检测规则后，我们可以考虑使用一种称为“修正规则”的第二种类型的规则将样本重新分配给另一类。此类规则是基于条件-类别对的一个子集 $CC_i \subseteq C \times \mathcal{C}$ 形成的。

$$corr_i(\omega) \leftarrow \bigvee_{q,r \in CC_i} (cond_q(\omega) \wedge pred_r(\omega)) \tag{2}$$

与两种类型规则相关联的是以下值——如果没有条件，则两者均定义为零。

支持 ($s$): 在 $\mathcal{O}$ 中身体为真的样本比例。

支持关于类 $i$($s_i$): 在模型预测类别为 $i$ 的样本子集中，其中主体为真的样本比例（注意分母是 $N_i$）。

置信度 ($c$): 身体和头部同时为真的次数除以身体为真的次数。

现在我们展示一些分析结果，这些结果为我们的学习算法提供了指导。我们学习策略的第一步是学习检测规则（确定由 $f_\theta$ 做出的分类决策被认为错误的条件），然后学习校正规则（通过将样本分配到新的运动类别来纠正已检测到的错误）。我们将这两项任务形式化如下。

通过错误检测规则改进。对于给定的类别 $i$，找到一组条件 $DC_i$，使得精度最大化且召回率最多减少 $\epsilon$。

通过纠错规则的改进。对于给定的类别 $i$，找到 $C \times \mathcal{C}$ 的一个子集 $CC_i$，使得精确率和召回率最大化。

**检测规则的性质**。首先，我们考察使用一个错误检测规则对精确率和召回率的影响。我们的第一个结果表明了精确率提升的一个界限。如果类别支持（$s_i$）小于 $1 - P_i$（这是我们预期的，因为该规则设计用于检测失败结果中的 $1 - P_i$ 部分），那么我们也可以证明数量 $c \cdot s_i$ 给出了精确率提升的一个上限。[2]

[2]所有形式结果的完整证明可以在 https://arxiv.org/abs/2308.14250 找到。

**定理** 1. 在条件 $s_i \leq 1 - P_i$ 下，模型 $f_\theta$ 对类别 $i$ 的精度，初始精度为 $P_i$，在应用具有支持 $s_i$ 和置信度 $c$ 的错误检测规则后，会增加一个关于 $s_i$ 和 $c$ 的函数，且不超过 $c \cdot s_i$，并且该数量是关于规则中条件集合的归一化多 matroid 次模函数 $DC_i$。

错误检测规则可能导致召回率保持不变或降低。我们的下一个结果精确地告诉我们召回率将减少多少。

**定理** 2. 应用该规则检测错误后，召回率将下降 $(1 - c)s_i \frac{R_i}{P_i}$，并且该数量是关于规则 $DC_i$ 中条件集合的归一化多 matroid 次模函数。

Algorithm 1 **规则学习检测**

Require: 类别 $i$，召回减少阈值 $\epsilon$，条件集 $C$
Ensure: 条件子集 $DC_i$
$DC_i := \emptyset$
$DC^* := \{c \in C \text{ s.t. } NEG_{\{c\}} \leq \epsilon \cdot \frac{N_i P_i}{R_i}\}$
while $DC^* \neq \emptyset$ do
  $c_{best} = \arg\max_{c \in DC^*} POS_{DC_i \cup \{c\}}$
  Add $c_{best}$ to $DC_i$
  $DC^* := \{c \in C \setminus DC_i \text{ s.t. } NEG_{DC_i \cup \{c\}} \leq \epsilon \cdot \frac{N_i P_i}{R_i}\}$
end while
return $DC_i$

由于定理 1 和 2 中识别的数量是次模且单调的，我们可以看到选择一组规则以最大化 $c \cdot s_i$ 并受约束于 $(1 - c)s_i \frac{R_i}{P_i} \leq \epsilon$ 是一个特殊情况的“次模成本次模背包”(SCSK) 问题，并可以用一个简单的贪婪算法[Iyer and Bilmes, 2013]近似求解，该算法具有多项式运行时间的近似保证（定理 4.7 来自[Iyer and Bilmes, 2013]）。我们的算法**判规则学习**是这样一种方法的具体实现，即为给定的类创建一个错误检测规则，以最大化精度而不降低召回率超过 $\epsilon$。这里，$\epsilon$ 被视为超参数。此外，$POS_{DC}$ 和 $NEG_{DC}$ 简单来说是满足某个集合 $DC$ 条件的样本数量，并且是错误（对于 $POS_{DC}$）和非错误（对于 $NEG_{DC}$）的数量。换句话说，给定一组条件类别对以及感兴趣的规则，这里 $BOD$ 是满足错误检测规则体（类别-条件对）的例子数量，而这里 $POS$ 是满足错误检测规则体（类别-条件对）和头部的例子数量。$P_i, R_i$ 是类 $i$ 的精度和召回率，而 $N_i$ 是模型分类为类 $i$ 的样本数量。

**校正规则的性质**。在接下来的内容中，我们将检查修正规则的结果。这里，头部带有谓词 $corr_j$ 的纠错规则将具有集合 $CC_i \subseteq C \times \mathcal{C}$ 的元素的析取。此外，请注意，这里使用的是支持 $s$ 而不是类支持（$s_i$）。在这里我们发现精度和召回率随着规则置信度的增加而增加（定理 3）。

**定理** 3. 对于纠错规则的应用，精度和召回率仅在规则置信度（$c$）增加时才会增加。

此结果表明，优化置信度将同时优化精确率和召回率。然而，这不是一个在 $CC_i$ 上的单调函数，因此我们采用了一种基于[Buchbinder et al., 2012]的快速启发式方法进行非单调优化，该方法由本文中的**相关规则学习**提出。在这里，我们将考虑一组初始条件-类别对 $CC_{all}$，它是 $C \times \mathcal{C}$ 的一个子集。对于我们要为其创建纠错规则的给定类别，我们使用我们的方法从这个较大的集合中选择 $CC_i$。注意这里，$POS_{CC}$ 是满足规则体和头（在这种情况下为 $corr_i(\omega)$）的样本数量，给定一组条件-类对 $CC$，而 $BOD_{CC}$ 是满足由集合 $CC$ 形成的体的样本数量。

Algorithm 2 **关联规则学习**

Require: 类 $i$，条件-类对集合 $CC_{all}$
Ensure: 条件-类对的子集 $CC_i$
$CC_i := \emptyset$
$CC'_i := CC_{all}$
Sort each $(c, j) \in CC_{all}$ from greatest to least by $\frac{POS_{\{(c,j)\}}}{BOD_{\{(c,j)\}}}$
and remove $\frac{POS_{\{(c,j)\}}}{BOD_{\{(c,j)\}}} \leq P_i$
for $(c, j) \in CC_{all}$ selected in order of the sorted list do
  $a := \frac{POS_{CC_i \cup \{(c,j)\}}}{BOD_{CC_i \cup \{(c,j)\}}} - \frac{POS_{CC_i}}{BOD_{CC_i}}$
  $b := \frac{POS_{CC'_i \setminus \{(c,j)\}}}{BOD_{CC'_i \setminus \{(c,j)\}}} - \frac{POS_{CC'_i}}{BOD_{CC'_i}}$
  if $a \geq b$ then
    $CC_i := CC_i \cup \{(c, j)\}$
  else
    $CC'_i := CC'_i \setminus \{(c, j)\}$
  end if
end for
if $\frac{POS_{CC_i}}{BOD_{CC_i}} \leq P_i$ then
  $CC_i := \emptyset$
end if
**返回** $CC_i$

**学习检测和纠正规则**。使用**相关规则学习**创建的纠错规则将为目标类别的规则提供最优的精确度和召回率改进，但在多类别问题中，它会导致其他某些类别的召回率下降。然而，我们可以结合错误检测和纠正规则来克服这个困难。直观上，首先为每个类别创建错误检测规则，这实际上会重新将任何样本分配到一个“未知”类别。然后，我们根据错误检测规则选择的条件创建一组 $CC_{all}$（用作**关联规则学习**的输入）。这样，我们就不会降低超出应用错误检测规则时发生的召回率。

**算法效率**。我们注意到这些算法非常高效。例如，**决定规则学习**在条件数量上是二次的，在样本数量上是一次的。然而，在实践中它的表现更好，因为外部循环迭代次数远少于总条件数，并且每次迭代中选择的条件数量

Algorithm 3 **细节相关规则学习**

Require: 记忆减少阈值 $\epsilon$，条件集 $C$
Ensure: 规则集 $\Pi$
$\Pi := \emptyset$
$CC_{all} := \emptyset$
for Each class $i$ do
  $DC_i := \text{DetRuleLearn}(i, \epsilon, C)$
  if $DC_i \neq \emptyset$ then
    $\Pi := \Pi \cup$
    $\{error(\omega) \leftarrow pred_i(\omega) \wedge \bigvee_{j \in DC_i} cond_j(\omega)\}$
  end if
  for $cond \in DC_i$ do
    $CC_{all} := CC_{all} \cup \{(cond, i)\}$
  end for
end for
for Each class $i$ do
  $CC_i := \text{CorrRuleLearn}(i, CC_{all})$
  if $CC_i \neq \emptyset$ then
    $\Pi := \Pi \cup$
    $\{corr_i(\omega) \leftarrow \bigvee_{q,r \in CC_i} (cond_q(\omega) \wedge pred_r(\omega))\}$
  end if
end for
**返回** $\Pi$

都会减少。同样地，算法**规则学习**在样本数量上是一次的，在条件-类对的数量上也是一次的。

**错误检测和纠正的条件**。实际上，我们的算法创建EDCR规则（集合 $C$）的条件来源需要被具体化。我们采用了两种直接的方法来实现这一点。首先，我们使用分类器的二进制版本——对于给定类别 $i$，我们有一个二元分类器 $g_i$，如果 $g_i$ 将样本 $\omega$ 分配为 $i$，则返回“true”，否则返回“false”。通过这种方式，对于每个样本 $\omega$，我们为每个类别有一个 $g_i(\omega)$ 条件。创建条件的第二种方法是基于样本中车辆速度的异常行为。这里，如果给定样本的速度超过一个阈值（根据训练数据的真实最大值），则该速度条件为“true”，否则为“false”。

## 4 实验评估

**实验设置**。像[Kim et al., 2022]这样的先前工作由于训练集和测试集之间的分割，已知存在数据泄露问题，主要是因为运动序列的某些部分同时存在于训练集和测试集中 [Zeng et al., 2023]。本文中我们检查了一个没有重叠的训练-测试拆分方法，避免了这个错误，并且更接近我们的目标应用场景。本论文中的评估使用的是从 GeoLife 项目获得的 GPS 轨迹 [Zheng et al., 2008]，其中包括真实标签（请注意，在撰写本文时，针对我们的目标应用的真实标签数据不可用）。所有实验均在配备有 2000 MHz AMD EPYC 7713 CPU 和 NVIDIA GA100 GPU 上使用 Python 3.10 及 PyTorch 进行。源代码可通过 https://github.com/lab-v2/Error-Detection-and-Correction 获取。

| Evaluated Model | Error Precision via EDCR | Error Recall via EDCR | Error F1 via EDCR |
|---|---|---|---|
| LRANa | 0.999 | 0.941 | 0.969 |
| LRCN | 0.996 | 0.780 | 0.875 |
| CNN | 0.987 | 0.982 | 0.984 |

表 1: EDCR 错误检测结果 - 此表展示了 EDCR 检测三种不同模型错误的能力。

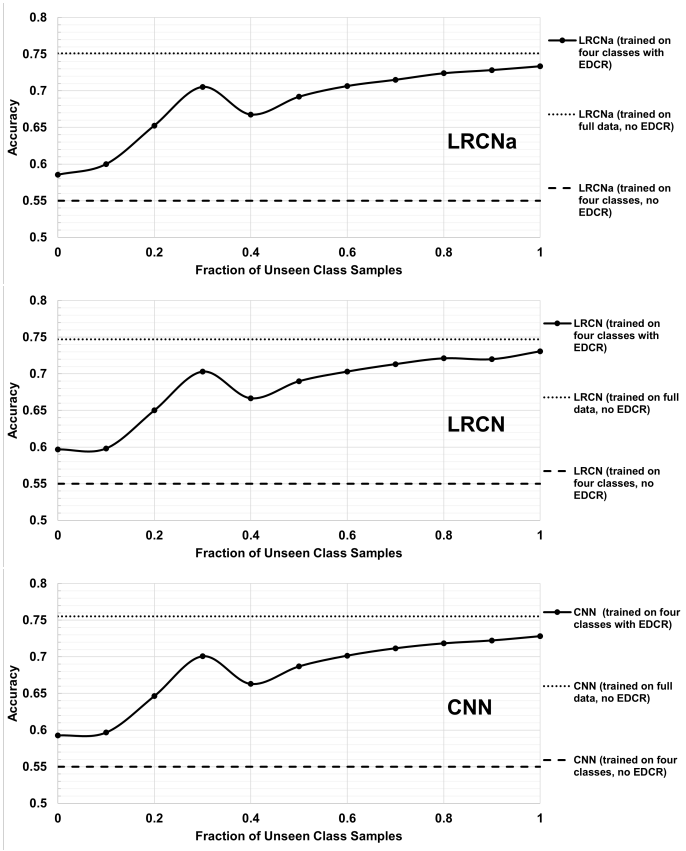


图 2: 移除训练中的两个运动类别的实验结果，针对 LRCNa、LRCN 和 CNN 模型。

**错误检测实验**。首先我们检查了学习到的错误检测规则在基础模型中检测错误的能力。在这里，我们检查了三种基础模型架构 CNN [Dabiri and Heaslip, 2018]、LRCN [Kim et al., 2022]，以及我们的带额外注意力头版本的 LRCN (LRCNa)。在这个实验中，错误检测规则是从与模型相同的训练数据中进行训练的。类似于之前

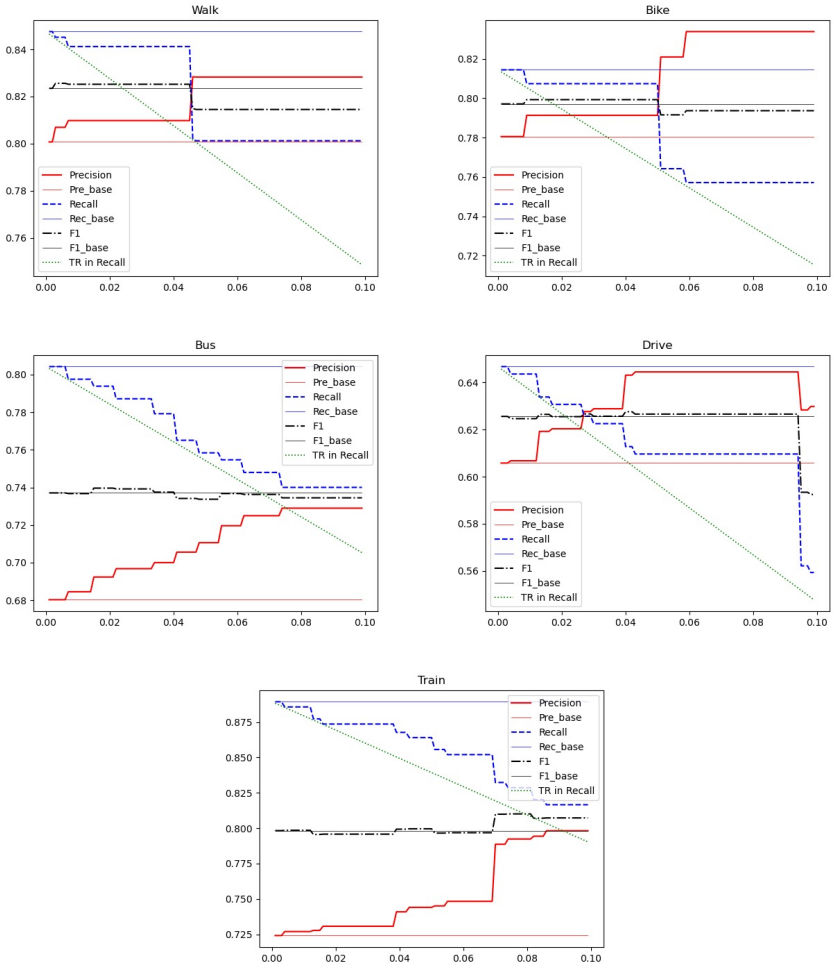


图 3: LRCN 结果作为错误检测和纠正规则应用的函数 $\epsilon$。TR 在召回率中是基于分析结果的理论召回率减少。

关于检查机器学习模型检测错误能力的工作 [Daftry et al., 2016]，我们评估了规则识别错误的精度、召回率和 F1 值。这些可以被视为我们的学习到的错误检测规则正确返回错误的结果的比例（错误精度）、已识别错误的比例（错误召回率），以及这两者的调和平均值（错误 F1）。表 1 所示的结果表明，在所有模型类型中检测错误的精度和召回率均保持较高水平——特别是对于 SOTA 模型（LRCN）中的错误，获得了一个 0.875 的 F1 值，并且最高 F1 值为 0.984（针对 CNN）。

**包含额外类别的测试数据**。我们的用例中一个关键的关注点是在数据与训练数据不同的环境中部署运动轨迹分类的能力——特别是包含以前未识别的类。为了研究这一点，我们在不纳入**行走**和**驱动**类的情况下训练了 CNN、LRCN 和 LRCNa 模型（图 2）。这里我们注意到检测和纠正都被使用了。我们最初在模型中用相同的数据学习 EDCR 规则——这导致没有样本被纠正为训练数据中未见过的类，实际上是通过 EDCR 对基础模型进行了零样本调整。然而，由于检测，这仍然为 LRCNa、LRCN 和 CNN 分别带来了 6.41%、8.51% 和 7.76% 的准确性提升。然后我们加入了未见数据中的少量样本（图 2 的 x 轴），使基础模型进行了少量样本调优。这里仅使用了 20% 的未见过类别的样本，我们在所有三个模型上获得了 0.65 的整体准确性，这代表了一个 17 – 18% 的提升。我们注意到这些结果是在没有直接访问基础模型的情况下获得的，这可能表明 EDCR 具有将任意 $f_\theta$ 模型适应新场景的潜力——这是我们政府客户的关键应用场景。

**精确率-召回率权衡**。在我们的算法设计中，一个关键的直觉是超参数能够使 $\epsilon$ 平衡精度和召回率。因此，我们考察了在与训练数据相似的测试数据上变化 $\epsilon$ 的效果（LRCN 的结果显示在图 3 中）。回忆一下，$\epsilon$ 被解释为召回率的最大减少量。我们观察并验证了理论上召回率的减少（TR）是经验性的，实验显示在所有情况下，召回率不低于超参数 $\epsilon$ 指定的阈值，尽管随着 $\epsilon$ 增加，召回

率下降。许多情况下，实验评估显著减少了预期之外的召回率。我们还看到了 $\epsilon$、精确度和召回率之间的明确关系：提高 $\epsilon$ 导致精确度上升而召回率下降——这也与我们的分析结果一致。我们也注意到尽管**相关规则学习**要求一个 $\epsilon$ 超参数，但可能为每个类别设置不同的值（例如，在召回率重要的类别中使用较低的值，在假阳性代价高的类别中使用较高的值）。这可能是有益的，因为不同类别的 F1 似乎在 $\epsilon$ 的不同值上达到峰值。我们将异构 $\epsilon$ 设置的研究留待未来工作。

| Evaluated Model | No EDCR (baseline) | With EDCR (ours) |
|---|---|---|
| LRCNa | 0.751 | 0.763 (+1.6%) |
| LRCN | 0.747 | 0.760 (+1.7%) |
| CNN | 0.755 | 0.755 (± 0%) |

表 2: 所有类别均使用和不使用 EDCR 时的整体准确性。

**通过 EDCR 提高准确性**。我们还调查了 EDCR 对基础模型整体准确性提升的能力。在这里，我们训练了三个模型（LRCNa、LRCN、CNN）以及相关的 EDCR 规则（与模型相同的训练数据），并在测试集上评估了应用规则和不应用规则的整体准确性（参见表 2）。我们发现，EDCR 在 LRCN 和 LRCNa 模型中提供了显著的改进 - 实际上，在训练和测试之间没有重叠的情况下进行评估时，这有效确立了一个新的 SOTA。我们还检查了其他训练和测试之间的划分（未显示），并获得了类似的结果。

## 5 结论

我们提出了一种基于规则的框架，用于检测和纠正监督神经模型在运动轨迹分类中的错误。我们的框架使用训练数据来学习将在测试阶段使用的规则。首先，我们使用检测规则识别被监督模型误分类的运动轨迹，然后我们使用校正规则重新对运动进行分类。此外，我们正式证明了所学规则的信心和支持与精度和召回率等分类指标变化之间的关系。为了展示 EDCR 的经验验证，我们首先报告框架识别错误的能力，F1 分数高达 0.984。我们还展示了通过采用 EDCR 框架整体准确性的提高超过了最先进的模型。我们的框架在遇到训练数据中未见的新类别时特别有用，零样本调优时未见运动的准确性提高了 8.51% 超过最先进水平。此外，我们讨论了将轨迹分类方法操作化到部署系统中的方式。未来工作有几个方向。首先，我们将探索其他创建条件的方法，特别是利用来自符合预测的想法 [Sun et al., 2023]。另一个方向是寻找替代解决方案来学习允许更复杂规则结构的规则。最后，在本文中提出的用于检测和纠正机器学习模型错误的规则可能在诸如视觉等领域中有用。为了可靠地将视觉模型整合到如物体检测、图像分类、运动追踪等现实世界应用的任务中，可以利用 EDCR 框架识别并纠正其误分类以提高整体系统的准确性和鲁棒性。

## 伦理声明

没有伦理问题。

## 6 致谢

本研究得到了智能先进研究项目活动（IARPA）通过内政部/内商业中心（DOI/IBC）合同号 140D0423C0032 的支持。美国政府授权为政府目的复制和分发重印本，不论其上是否有版权注释。免责声明：此处包含的观点和结论是作者的，并不应被解读为代表 IARPA、DOI/IBC 或美国政府的官方政策或背书，无论是明确还是暗示。此外，部分作者得到了海军研究办公室（ONR）资助 N00014-23-1-2580 和陆军研究办公室（ARO）资助 W911NF-24-1-0007 的支持。

## 参考文献

[Aditya et al., 2023] Dyuman Aditya, Kaustuv Mukherji, Srikar Balasubramanian, Abhiraj Chaudhary, and Paulo Shakarian. PyReason: Software for open world temporal logic. In AAAI Spring Symposium, 2023.

[Bavikadi et al., 2024] Divyagna Bavikadi, Dyuman Aditya, Devendra Parkar, Paulo Shakarian, Graham Mueller, Chad Parvis, and Gerardo I Simari. Geospatial trajectory generation via efficient abduction: Deployment for independent testing. In Proceedings of the 40th International Conference on Logic Programming (ICLP 2024), 2024.

[Buchbinder et al., 2012] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science, pages 649–658, 2012.

[Chen et al., 2021] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. CoRR, abs/2106.01345, 2021.

[Cornelio et al., 2022] Cristina Cornelio, Jan Stuehmer, Shell Xu Hu, and Timothy Hospedales. Learning where and when to reason in neuro-symbolic inference. In The Eleventh International Conference on Learning Representations, 2022.

[Dabiri and Heaslip, 2018] Sina Dabiri and Kevin Heaslip. Inferring transportation modes from gps trajectories using a convolutional neural network. Transportation research part C: emerging technologies, 86:360–371, 2018.

[Daftry et al., 2016] Shreyansh Daftry, Sam Zeng, J. Andrew Bagnell, and Martial Hebert. Introspective perception: Learning to predict failures in vision systems, 2016.

[Dai et al., 2019] Wang-Zhou Dai, Qiuling Xu, Yang Yu, and Zhi-Hua Zhou. Bridging machine learning and logical reasoning by abductive learning. NeurIPS, 32, 2019.

[Fikioris et al., 2023] Giannis Fikioris, Kostas Patroumpas, Alexander Artikis, Manolis Pitsikalis, and Georgios Paliouras. Optimizing vessel trajectory compression for maritime situational awareness. GeoInformatica, 27(3):565–591, 2023.

[Hospedales et al., 2021] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(9):5149–5169, 2021.

[Huang et al., 2019] Haosheng Huang, Yi Cheng, and Robert Weibel. Transport mode detection based on mobile phone network data: A systematic review. Transportation Research Part C: Emerging Technologies, 101:297–312, 2019.

[Huang et al., 2023] Yu-Xuan Huang, Wang-Zhou Dai, Yuan Jiang, and Zhi-Hua Zhou. Enabling knowledge refinement upon new concepts in abductive learning. 2023.

[Itkina and Kochenderfer, 2022] Masha Itkina and Mykel J. Kochenderfer. Interpretable self-aware neural networks for robust trajectory prediction, 2022.

[Ivanov et al., 2021] Radoslav Ivanov, Taylor Carpenter, James Weimer, Rajeev Alur, George Pappas, and Insup Lee. Verisig 2.0: Verification of neural network controllers using taylor model preconditioning. In Computer Aided Verification: 33rd International Conference, CAV 2021, Virtual Event, July 20 – 23, 2021, Proceedings, Part I, pages 249–262. Springer-Verlag, 2021.

[Iyer and Bilmes, 2013] Rishabh Iyer and Jeff Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, page 2436 – 2444, Red Hook, NY, USA, 2013. Curran Associates Inc.

[Janner et al., 2021] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In Advances in Neural Information Processing Systems, 2021.

[Jothimurugan et al., 2021] Kishor Jothimurugan, Suguman Bansal, Osbert Bastani, and Rajeev Alur. Compositional reinforcement learning from logical specifications. In Advances in Neural Information Processing Systems, 2021.

[Kim et al., 2022] Jinsoo Kim, Jae Hun Kim, and Gunwoo Lee. Gps data-based mobility mode inference model using long-term recurrent convolutional networks. Transportation Research Part C: Emerging Technologies, 135:103523, 2022.

[Lin and Hsu, 2014] Miao Lin and Wen-Jing Hsu. Mining gps data for mobility patterns: A survey. Pervasive and mobile computing, 12:1–16, 2014.

[Ma et al., 2020] Meiyi Ma, Ji Gao, Lu Feng, and John Stankovic. Stlnet: Signal temporal logic enforced multivariate recurrent neural networks. Advances in Neural Information Processing Systems, 33:14604–14614, 2020.

[Maes and Nardi, 1988] Pattie Maes and Daniele Nardi. Meta-level architectures and reflection. 1988.

[Ramanagopal et al., 2018] Manikandasriram Srinivasan Ramanagopal, Cyrus Anderson, Ram Vasudevan, and Matthew Johnson-Roberson. Failing to learn: Autonomously identifying perception failures for self-driving cars. 3(4):3860–3867, 2018.

[Simoncini et al., 2018] Matteo Simoncini, Leonardo Taccari, Francesco Sambo, Luca Bravi, Samuele Salti, and Alessandro Lori. Vehicle classification from low-frequency gps data with recurrent neural networks. Transportation Research Part C: Emerging Technologies, 91:176–191, 2018.

[Sun et al., 2023] Jiankai Sun, Yiqi Jiang, Jianing Qiu, Parth Nobel, Mykel J Kochenderfer, and Mac Schwager. Conformal prediction for uncertainty-aware planning with diffusion dynamics model. In NeurIPS, volume 36, pages 80324–80337, 2023.

[Vanschoren, 2018] Joaquin Vanschoren. Meta-learning: A survey. arXiv preprint arXiv:1810.03548, 2018.

[Wang et al., 2017] Hao Wang, GaoJun Liu, Jianyong Duan, and Lei Zhang. Detecting transportation modes using deep neural network. IEICE TRANSACTIONS on Information and Systems, 100(5):1132–1135, 2017.

[Zeng et al., 2023] Jiaqi Zeng, Yi Yu, Yong Chen, Di Yang, Lei Zhang, and Dianhai Wang. Trajectory-as-a-sequence: A novel travel mode identification framework. 146:103957, 2023.

[Zheng et al., 2008] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. Understanding mobility based on gps data. In Proceedings of the 10th international conference on Ubiquitous computing, pages 312–321, 2008.

[Zhou et al., 2022] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.