可解释人工智能在雷达资源管理中的应用：深度强化学习中修改的 LIME 方法

Ziyang Lu, M. Cenk Gursoy, Chilukuri K. Mohan, Pramod K. Varshney
Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse NY, 13066
{zlu112, mcgursoy, ckmohan, varshney}@syr.edu

摘要—深度强化学习在决策过程中的应用已经得到了广泛的研究，并且在包括雷达资源管理（RRM）在内的多个领域中表现出优于传统方法的性能。然而，神经网络的一个显著限制是其“黑箱”特性，最近的研究工作越来越多地集中在可解释人工智能（XAI）技术上，以描述神经网络决策背后的原理。一种有前景的 XAI 方法是局部可解释模型不可知解释（LIME）。然而，LIME 中的采样过程忽略了特征之间的相关性。本文中，我们提出了一种将深度学习（DL）集成到采样过程中的改进 LIME 方法，我们称之为 DL-LIME。我们在雷达资源管理的深度强化学习中应用了 DL-LIME。数值结果表明，在保真度和任务性能方面，DL-LIME 优于传统的 LIME，并且在两个指标上都表现出色。DL-LIME 还提供了关于哪些因素在雷达资源管理决策过程中更为重要的见解。

***Index Terms*—认知雷达，深度强化学习，可解释人工智能，局部可解释模型不可知解释（LIME）**

I. 介绍

A. 深度强化学习与认知雷达暨其相关领域研究进展报告

深度强化学习（DRL）是解决动态环境中复杂决策问题的强大工具。通过将神经网络与强化学习相结合，DRL 算法可以处理复杂的任务并达到人类级别的性能，如 [1],[2],[3],[4] 所示。

认知雷达是一种可以增强在动态和复杂环境中雷达性能的技术，最近的研究证明了深度强化学习（DRL）在优化动态环境中的认知雷达操作参数方面的有效性，例如在拥挤频谱设置中的雷达检测和跟踪 [5]。在 [6] 中，利用了 DRL 进行场景自适应雷达跟踪。[7] 的工作介绍了一种基于 DRL 的服务质量管理方法，用于雷达资源分配，成功地平衡了多个性能指标。[8] 将强化学习应用于多功能雷达的重访间隔选择中，其中 Q 学习算法在保持与传统方法相似的跟踪丢失概率的同时降低了跟踪负荷。在我们之前的 [9] 和 [10] 研究中，DRL 被应用来决定认知雷达系统中多目标跟踪的停留时间分配，并且 DRL 在时间预算限制下比启发式方法表现更好。

B. 局部可解释的模型不可知解释（*LIME*）

随着机器学习模型变得越来越复杂并在各个领域广泛使用，理解其决策过程变得必要。可解释的人工智能（XAI）专注于解读和理解模型预测，这对于确保公平性和使模型调试成为可能 [11] 至关重要。在医疗、金融和自主系统等高风险领域中，XAI 的重要性尤为明显，在这些领域中，模型决策可能会产生显著的实际影响 [12]。

两种方法已经作为模型解释的主要框架出现：局部可解释的模型不可知解释（LIME）[13] 和夏普利加性解释（SHAP）[14]。虽然 SHAP 提供了基于合作博弈论的理论基础特征归因，LIME 通过创建局部替代模型来近似复杂模型的行为 [15]，提供了一种直观且计算效率高的方法，使其更适合实时应用。在这项工作中，我们专注于 LIME，因为其计算效率和灵活性对于雷达系统中的实时应用至关重要。

然而，LIME 的基本假设特征独立性在特征表现出强相关性的场景下构成了一个重要限制。原始的 LIME 算法在扰动过程中将每个特征视为一个独立变量，可能会生成不符合底层数据分布的实际样本。最近的工作已经开始解决各个领域的这一局限性。例如，[16] 中的工作提出了 MPS-LIME，这是一种专门针对图像分类任务修改后的扰动采样方法，它使用基于图的团构造来保持图像超像素之间的空间相关性。然而，这种方法专用于图像输入，不能直接应用于时间或状态数据，如雷达系统。为了解决相关的特征问题，我们提出了一种深度学习辅助的 LIME（DL-LIME）框架。

C. 贡献

特别地，本文有以下贡献。

- 在这项工作中，我们专注于解读我们之前开发的用于雷达驻留时间分配的 DRL 方法的决策过程。
- 虽然 LIME 为模型解释提供了一个有前景的框架，但其关于特征独立的基本假设在雷达系统中成为限制因素，在这种系统中，状态输入可能会由于物理约束和时间动态而表现出强相关性。为了应对这一局限性，我们提出了一种对 LIME 的修改方案，该方案引入深度神经网络（DNN）来捕捉并保持雷达状态元素之间的相关性，例如目标位置及其相应的跟踪成本。
- 通过全面的数值分析，我们证明了我们的改进版 LIME 在忠实度和任务性能方面优于传统的 LIME 方法，这里的忠实度衡量的是 LIME 产生的决策与 DRL 模型实际决策之间的相似程度。
- 我们通过提供 DRL 代理在各种雷达跟踪场景中资源分配策略的可解释性解释来验证我们方法的有效性。

II. 系统模型

我们考虑一个认知雷达系统，该系统在搜索潜在目标和跟踪已检测到的目标之间动态分配时间。系统模型基于我们在 [9], [10] 中的先前分析，并为完整性起见总结了关键组件如下。

A. 目标运动和测量模型

目标状态在时间 t 定义为 $\boldsymbol{x}_t = [x_t, y_t, \dot{x}_t, \dot{y}_t]^T$，其中 (x_t, y_t) 表示目标的位置坐标而 $(\dot{x}_t, \dot{y}_t)$ 表示其速度分量。目标运动遵循一个线性状态空间模型，具有高斯机动噪声：

$$\boldsymbol{x}_{t+1} = \boldsymbol{F}_t \boldsymbol{x}_t + \boldsymbol{w}_t \tag{1}$$

其中 $\boldsymbol{F}_t$ 是状态转移矩阵，$\boldsymbol{w}_t$ 是协方差矩阵为 $\boldsymbol{Q}_t$ 的过程噪声，用于建模目标的机动能力。

雷达通过非线性测量模型和高斯噪声 $\boldsymbol{v}_t$ 获得距离和方位测量值 $\boldsymbol{z}_t$:

$$\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{v}_t = \begin{bmatrix} \sqrt{x_t^2 + y_t^2} \\ \tan^{-1}(\frac{y_t}{x_t}) \end{bmatrix} + \boldsymbol{v}_t \tag{2}$$

其中测量噪声 $\boldsymbol{v}_t$ 具有协方差矩阵 $\boldsymbol{R}_t$，而 $h(\cdot)$ 代表从笛卡尔坐标到极坐标的非线性变换。

B. 跟踪性能

目标跟踪采用扩展卡尔曼滤波器（EKF）来处理非线性测量模型。每个目标的跟踪性能通过以下成本函数进行评估 [9]:

$$c_t(\tau_t) = \text{trace}(\boldsymbol{E}\boldsymbol{P}_{t|t}\boldsymbol{E}^T) \tag{3}$$

其中，$\boldsymbol{P}_{t|t}$ 是从 EKF 得到的后验状态估计协方差矩阵，$\boldsymbol{E}$ 是一个投影矩阵，用于从状态向量中提取位置分量，τ_t 是分配给每个目标的驻留时间，在测量周期 T_0 内。

C. 扫描和检测

雷达的扫描性能由信噪比 (SNR) 表示:

$$\text{SNR}_{\text{scan}} = \frac{P_t \tau_{\text{beam}} G_t G_r \lambda_r^2 \sigma}{(4\pi)^3 r^4 L k T_s} \tag{4}$$

其中，

- P_t 是发射功率，
- τ_{beam} 是脉冲持续时间，
- G_t 和 G_r 是发射和接收天线增益，
- λ_r 是雷达信号的波长，
- σ 是目标的雷达截面积，
- r 是雷达与目标的距离，
- L 是系统损耗因子，
- k 是玻尔兹曼常数，
- T_s 是系统温度。

扫描光束持续时间 τ_{beam} 与总扫描时间 τ_s 相关，由以下公式给出:

$$\tau_s = \frac{360^\circ}{\phi} \tau_{\text{beam}} \tag{5}$$

其中 ϕ 是相邻雷达光束之间的相位延迟。

扫描有效性通过 Γ 来衡量，定义为可检测最大区域与参考区域的比值:

$$\Gamma = \left(\frac{r_{\max}}{r_0} \right)^2 \tag{6}$$

其中，$r_{\max}$ 是由检测所需的最小信噪比决定的最大可检测范围，而 r_0 通常设定为雷达的默认工作范围。

航迹初始化遵循一种 3 取 4 的检测模型，其中需要在连续四个扫描中关联三个测量值来建立一个航迹。测量值使用预定义的距离阈值通过全局最近邻（GNN）方法与先前接收到的测量值进行关联。

III. 约束深度强化学习框架

遵循 [9] 中的方法，我们将雷达资源管理问题（用于扫描和多目标跟踪）形式化为一个约束马尔可夫决策过程 (CMDP)，并使用约束深度强化学习 (CDRL) 来解决它。该框架设计用于同时处理性能优化和时间预算约束。与 [9] 和 [10] 中基于深度 Q 网络 (DQN) 的框架不同，本文中使用的 DRL 框架算法为深度确定性策略梯度 (DDPG)。

A. 状态

在时间 t 时的状态 $\boldsymbol{s}_t$ 定义为

$$\boldsymbol{s}_t = \left[\left\{ \frac{(\hat{x}_{t-1}^n}{\eta}, \frac{\hat{y}_{t-1}^n)}{\eta} \right\}_{n=1}^N, \left\{ \frac{c_{t-1}^n(\tau_{t-1}^n)}{\eta} \right\}_{n=1}^N, \frac{\lambda_{t-1}}{\eta} \right] \tag{7}$$

其中 $\{(\hat{x}_{t-1}^n, \hat{y}_{t-1}^n)\}_{n=1}^N$ 是在时间 $t-1$ 通过 EKF 估计的所有 N 目标的位置，$\{c_{t-1}^n(\tau_{t-1}^n)\}_{n=1}^N$ 是在时间 $t-1$ 所有 N 目标的跟踪成本，λ_{t-1} 是对偶变量（用于管理时间预算约束，如下文所述）。状态的大小是 $3N+1$。η 是一个规范化因子，以防止输入到神经网络的数据过大。对于在时间 $t-1$ 时未被跟踪的目标，它们对应的位置估计值 $(\hat{x}_{t-1}^n, \hat{y}_{t-1}^n)$ 和跟踪成本 $c_{t-1}^n(\tau_{t-1}^n)$ 被设置为零。

1) 行动: 动作 $\boldsymbol{a}_t$ 是分配给每个目标跟踪的驻留时间集合:

$$\boldsymbol{a}_t = \{\tau_t^n\}_{n=1}^N, \quad \text{where } \tau_t^n \in [0, T_0] \tag{8}$$

剩余时间分配给扫描，即 $\tau_s = T_0 - \sum_{n-1}^N \tau_t^n$。

2) 奖励: 奖励函数 r_t 定义为

$$r_t = U_t(\{\tau_t^n\}_{n=1}^N) - \lambda_t \left(\sum_{n=1}^N \frac{\tau_t^n}{T_0} - \Theta_{max} \right) \tag{9}$$

其中第一项 $U_t(\{\tau_t^n\}_{n=1}^N)$ 是定义为 $U_t(\{\tau_t^n\}_{n=1}^N) = -\sum_{n=1}^N c_t^n(\tau_t^n) + \beta\Gamma$ 的效用函数，其中 β 是跟踪和扫描性能之间的权衡系数。第二项是违反时间预算约束的惩罚，Θ_{max} 表示跟踪的总时间预算，λ_t 表示拉格朗日松弛中的对偶变量，该变量将约束优化问题转化为无约束优化问题。

在提出的 CDRL 框架中，对偶变量 λ 与神经网络参数一起更新为

$$\lambda_{t+1} = \max\left(0, \lambda_t + \alpha_\lambda \left(\sum_{n=1}^N \frac{\tau_t^n}{T_0} - \Theta_{max} \right)\right) \tag{10}$$

其中 α_λ 是对偶变量的学习率。λ_t 的更新动态调整约束违规的惩罚，引导学习过程向可行解靠拢。

IV. DL-LIME 解释

A. 石灰

局部可解释模型不可知解释框架（LIME）通过使用一个可解释的模型进行局部近似来解释任何分类器或回归器的预测。在本文中，我们使用 LIME 来解释上一节讨论的 DRL 代理的决策过程。解释过程包括以下关键步骤:

1) **实例选择：** 给定一个需要解释的实例状态 $\boldsymbol{s}_t$ 和一个训练好的 DRL 模型 $\pi(\cdot)$，LIME 首先识别出解释应有效的围绕 $\boldsymbol{s}_t$ 的局部区域。
2) **受扰状态生成：** LIME 通过对非零连续特征进行正态分布采样，在 $\boldsymbol{s}_t$ 周围生成扰动状态：

$$\boldsymbol{s}_t^{(k)} = \boldsymbol{s}_t + \boldsymbol{n}^{(k)} \tag{11}$$

其中 $\boldsymbol{n}^{(k)} = \{n_m^{(k)}\}_{m=1}^{d}$ 是具有维度 $d = 3N+1$（状态空间维度）的扰动向量，对于所有 d 组件而言，$n_m^{(k)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_m^2)$。参数 σ_m^2 控制局部邻域中扰动的幅度。为了更好地反映 DRL 代理遇到的实际状态分布，我们通过在多样化的雷达环境中部署训练良好的 DRL 代理来收集一个经验状态数据集 $\mathcal{D}$，然后计算每个状态组件的经验均值 μ_m 和方差 σ_m^2 以指导 LIME 中的扰动采样过程。
3) **相似性计算：** 对于每个扰动状态 $\boldsymbol{s}_t^{(k)}$，LIME 使用指数核函数计算与原始实例状态的相似度得分：

$$w(\boldsymbol{s}_t^{(k)}) = \sqrt{\exp\left(-\frac{D(\boldsymbol{s}_t, \boldsymbol{s}_t^{(k)})^2}{a^2}\right)} \tag{12}$$

其中 $D(\boldsymbol{s}_t, \boldsymbol{s}_t^{(k)}) = \|\boldsymbol{s}_t - \boldsymbol{s}_t^{(k)}\|_2$ 是原始状态 $\boldsymbol{s}_t \in \mathbb{R}^{3N+1}$ 和扰动状态 $\boldsymbol{s}_t^{(k)} \in \mathbb{R}^{3N+1}$ 之间的欧几里得距离，而 $a > 0$ 是控制局部邻域大小的核宽度参数。相似性分数 $w(\boldsymbol{s}_t^{(k)}) \in [0, 1]$ 随状态之间的距离呈指数级下降。
4) **局部模型训练：** LIME 通过最小化损失函数

$$\mathcal{L}(\boldsymbol{w}, b) = \sum_{k=1}^{K} w(\boldsymbol{s}_t^{(k)}) \|\pi(\boldsymbol{s}_t^{(k)}) - (\boldsymbol{w}^T \boldsymbol{s}_t^{(k)} + b)\|^2 + c\|\boldsymbol{w}\|^2 \tag{13}$$

中的以下项来拟合一个可解释的线性模型：

- 第一项是加权均方误差：
  - $\pi(\boldsymbol{s}_t^{(k)})$ 是由 DRL 代理给出的预测，
  - $\boldsymbol{w}^T \boldsymbol{s}_t^{(k)} + b$ 是由 LIME 模型给出的预测。
  - $w(\boldsymbol{s}_t^{(k)})$ 按照与原始状态的相似性来衡量误差。
- 是第二项，$c\|\boldsymbol{w}\|^2$ 是一个正则化项。

最优权重 $\boldsymbol{w}^*$ 和偏置 b^* 是通过求解

$$(\boldsymbol{w}^*, b^*) = \arg\min_{\boldsymbol{w}, b} \mathcal{L}(\boldsymbol{w}, b) \tag{14}$$

5) **特征重要性提取：** 拟合的线性模型的系数 $\boldsymbol{w}^*$ 提供特征重要性分数，表明每个特征如何贡献于模型的局部预测。

B. 提议的 DL-LIME

1) 动机： 传统的 LIME 过程在扰动期间假设特征独立，这在考虑的雷达系统中成为一个问题，因为在这些系统中状态特征表现出强相关性。例如，跟踪成本与目标到雷达的距离呈正相关。在相同的驻留时间内，对距离雷达更远的目标进行跟踪将更具挑战性，导致更高的成本。为了分析状态组件之间的关系，我们从一个经过良好训练的 DRL 代理在其雷达环境中操作中收集了

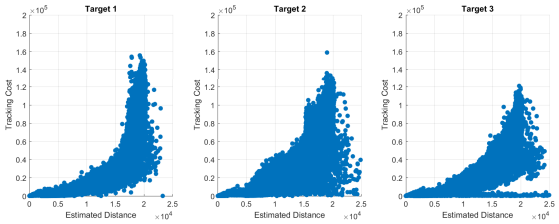


图 1: 跟踪成本和目标到雷达的距离

表 I: 实验中使用的超参数

Category	Parameter	Value
DDPG	State dimension	16
	State normalization factor η	10^7
	Action dimension	5
	Discount Factor	0.9
	Action bound (s)	2.5
	Initial dual variable λ_0	5000
	Step size of dual variable α_λ	15000
环境	Maximum number of targets N	5
	Number of Time Slots	50000
	Required false alarm probability P_f	10^{-3}
	Required probability of detection P_d	0.9
	Time budget for tracking Θ_{max}	0.9
	Target join probability	0.03
局部解释模型	Kernel width a	2.5
	Number of samples	10000
	Ridge regression regularization parameter c	10^{-3}
雷达	Range measurement noise $\sigma_{r,0}^2$ (m^2)	16
	Azimuth angle measurement noise $\sigma_{\theta,0}^2$ (rad^2)	1e-6
	Target maneuverability noise σ_w $((m/s^2)^2)$	16
	Measurement Cycle T_0 (s)	2.5
	Tradeoff Coefficient β	100000

轨迹数据 $\mathcal{D}$。对于每个目标 n，我们计算其估计距离作为 $d_t^n = \sqrt{x_t^{n2} + y_t^{n2}}$ 并提取相应的跟踪成本 $c_t^n(\tau_t^n)$。图 1中的散点图显示了跟踪成本和目标到雷达的距离之间的强正相关性，表明这些状态组件在考虑的问题中固有的关系。这种相关性表明，在传统的 LIME 过程中独立扰动这些组件可能会生成不现实的状态，可能降低由此产生的解释的可靠性。

2) 实现： 为了解决传统 LIME 算法的局限性，我们提出了一种扩展方法，该方法引入了深度神经网络（DNN）来学习状态组件之间的相关性。修改后的 LIME 算法使用这个 DNN 将选定的状态元素映射到它们的相关对应物。

如图 1所示，跟踪成本与其他状态组件表现出强烈的关联性。为了在扰动过程中保持这些固有的关系，我们设计了一个将状态组件映射到跟踪成本的 DNN。具体来说，该 DNN 以 $2N+1$ 个状态组件（不包括跟踪成本）作为输入，并预测所有 N 目标对应的跟踪成本 $c_t^n(\tau_t^n)_{n=1}^N$。这个 DNN 在收集的经验状态数据集 $\mathcal{D}$ 上预先训练，以学习底层的物理关系。在我们修改后的 LIME 方法中，我们不是独立地扰动所有的 $3N+1$ 个状态组件，而是只扰动不包括跟踪成本的状态组件，并利用已训练的 DNN 生成相应的跟踪成本。这种策略确保了扰动后的状态保持其分量之间的物理有意义的关系，从而增强了局部近似的保真度。

V. 数值结果

A. 实验设置

1) 超参数： 超参数列于表 I中。在 DDPG 算法中，有四个网络。演员网络及其目标网络具有相同的结构，

每个都有两层，分别包含 256 和 128 个神经元，并且层之间使用 ReLU 激活函数。

类似地，评论家网络及其目标网络具有相同的结构，由两层组成，每层包含 100 个神经元，并使用 ReLU 作为激活函数。演员网络和评论家网络的学习率均设置为 0.0002。

2) 目标生成模型： 在实验中，雷达系统被放置在原点，并且它可以跟踪最多 $N = 5$ 个目标。为了增加环境的多样性并验证所提出的框架在各种条件下的有效性，我们使用以下通用的目标生成模型：

- 每 100 次迭代，一个新的目标可以以 0.03 的概率加入环境。目标的初始位置和速度是随机生成的。
- 为了进一步多样化操作环境，任何在环境中出现超过 3000 个时隙的目标将被系统地移除。
- 假设雷达正在监测一个半径为 20 公里的区域。超出此范围的目标被认为已离开该环境，不再被雷达追踪。

3) 深度神经网络在 DL-LIME 中： 一个 DNN 用于根据其余状态输入预测跟踪成本，并在收集的数据集 $\mathcal{D}$ 上进行训练。输入大小为 11，输出大小为 5。有两个前馈隐藏层，其中一个包含 128 个神经元，另一个包含 64 个神经元。ReLU 用作激活函数，学习率设置为 0.0001。

B. 性能比较

为了评估所提出的框架，我们采用三个指标：

- **平均绝对误差（MAE）**：MAE 通过测量由 LIME 模型预测的动作与 DDPG 代理选择的动作之间的平均绝对差异来量化 LIME 近似的保真度。我们每 500 个时间槽设置一个检查点。在每个检查点期间，都会找到一个用于动作解释的 LIME 模型。以 $\boldsymbol{s}_t$ 为输入，可以确定 DDPG 代理 $\boldsymbol{a}_t$ 的动作以及由 LIME 模型提供的动作 $\boldsymbol{a}_t^{LIME}$。然后计算 MAE 作为
$$\mathrm{MAE} = \frac{1}{N}\|\boldsymbol{a}_t - \boldsymbol{a}_t^{\mathrm{LIME}}\|_1. \tag{15}$$
- **平均效用函数**：效用函数评估 LIME 模型在实际任务性能方面多好地近似 DDPG 代理的行为。对于每个时间槽 t，我们找到一个 LIME 模型来近似 DDPG 策略，然后在环境中执行 LIME 模型预测的动作，并记录得到的效用函数 $U_t(\{\tau_t^n\}_{n=1}^N$。该指标通过直接在环境中执行得出的动作，而不是仅仅测量策略近似的准确性（如 MAE 指标所做的那样），提供了一种衡量 LIME 捕捉 DDPG 代理决策有效性能力的实际方法。
- **运行时**：运行时通过测量每个时间槽解释决策所需的总运行时间来评估计算效率。
- **峰值性能期**：峰值性能时段衡量算法在与所有比较方法中达到最佳性能的时间槽的百分比。具体来说，它表示算法所实现的效用函数值超过其他方法的时间槽占 50,000 个时间槽中的比例。

表 II: DL-LIME、LIME 和 DDPG 的性能比较

方法	均方误差	效用	运行时间（秒）	峰值性能。周期
DDPG	-	4.39×10^4	-	48.57%
LIME	2.27	4.01×10^4	0.42	11.20%
DL-LIME	1.95	4.49×10^4	1.70	40.23%

我们评估了建议的 DL-LIME 和传统 LIME 方法的平均 MAE、效用函数、运行时间（以每决策秒为单位）以及峰值性能期，性能比较详见表 II。与传统的 LIME 方法相比，DL-LIME 实现了更低的平均 MAE 和更高的平均效用函数值，证明了所提出的 DL-LIME 方法在近似 DRL 代理行为方面的有效性。另一方面，由于深度学习算法的引入，DL-LIME 在运行时间比较中显示出更大的计算复杂性。表 II中的 DL-LIME 结果是通过生成 $10,000$ 个样本进行扰动得到的。通过调整 DL-LIME 中扰动过程中的样本数量，可以在计算效率和解释保真度之间灵活地平衡以满足特定的操作需求。我们测试了不同样本数下的 DL-LIME 性能，并在图 4中可以观察到运行时间和 MAE 之间的权衡。

从峰值性能期的比较中，我们观察到 DDPG 通过在最高百分比的时间段内保持最优性能优于其他方法，其次是 DL-LIME，它相对于传统 LIME 表现出更优的性能。

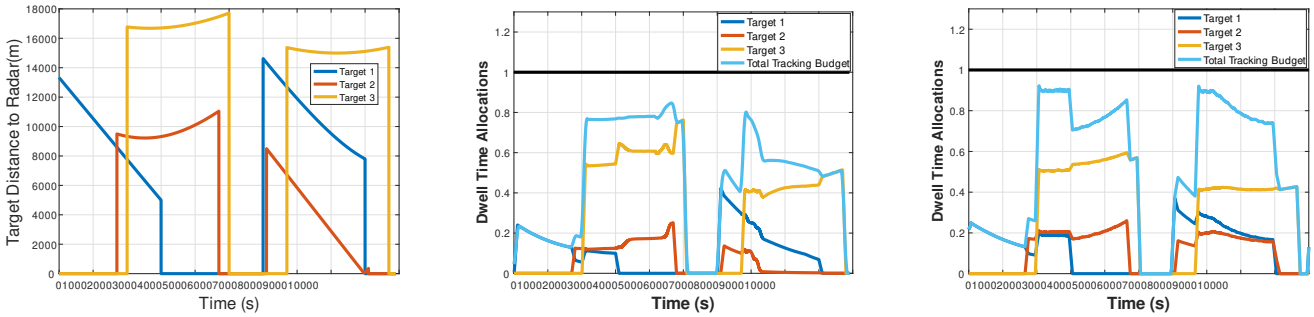
C. 深度学习 LIME 预测的动作

通过上一节计算出的 MAE，我们可以观察到 DL-LIME 与 DDPG 策略相比表现出更高的保真度。在本节中，我们展示了 LIME 模型预测的具体动作。figure 2a绘制了测试环境中目标物距离雷达的距离，时间槽为 $10,000$。figure 2b和 figure 2c分别描绘了 DDPG 代理和 LIME 模型预测的动作。比较 figure 2b和 figure 2c，我们观察到 DL-LIME 预测的动作与动态测试环境中由 DDPG 代理生成的动作非常相似，这表明所提出的 DL-LIME 模型在逼近 DDPG 代理的局部行为方面是有效的。两种方法都提供了合理的时间分配策略：

- 雷达距离更远的目标接收到更高的跟踪时间分配，这在 $t = 4000$ 处很明显。如图 2a所示，在这个时间点上，目标 2 和目标 3 正在被追踪，其中目标 3 距离雷达更远。DDPG 智能体和 DL-LIME 模型都为目标 3 分配了更多的跟踪时间。这种分配策略是直观的，因为距离更远的目标会经历更大的测量噪声，导致更加困难的跟踪条件和更高的跟踪成本。因此，将可用的时间预算更多地分配给这些具有挑战性的目标会导致更好的跟踪性能。
- 当跟踪任务的需求较低时，将分配更多时间给扫描任务，这在 $t = 1000$ 和 $t = 6500$ 处很明显。在 $t = 1000$ 处，只有目标 1 正在接受跟踪，因此跟踪需求较低。在 $t = 6500$ 处，尽管有两组目标正在被跟踪，但它们相对靠近雷达。在这两种情况下，总的跟踪时间分配仍然保持较低，如总跟踪预算曲线所示。雷达倾向于分配更多的时间来扫描环境（以便检测新出现的目标），这优化了整体效用函数。

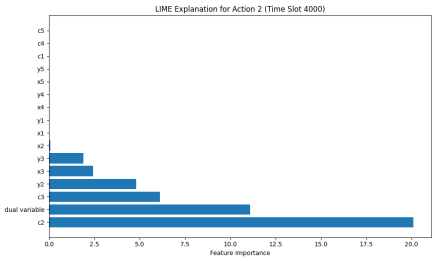
D. 说明

在本节中，我们检查了 DL-LIME 方法在特定时间槽中的解释。图 3展示了状态组件在个别行动决策解释中的重要性。请注意，动作 $\boldsymbol{a}_t = \{\tau_t^n\}_{n=1}^N$ 包括分配给跟踪当前目标的停留时间。在这里，我们将决策 τ_t^n 表示为动作 n。根据图 3中的结果，LIME 解释揭示了一种一致的模式，即与在前一时隙中跟踪目标 n 相关的成本 $c_{t-1}^n(\tau_{t-1}^n)$ 在决定动作 n（即当前时隙中跟踪目标 n 的驻留时间决策）中占据主导地位。例如，可以看出如图 3a所示，在解释动作 2 时，目标 2 的跟踪成本是最重要因素；如图 3b所示，在解释动作 3 时，目标 3 的跟踪成本是主要因素。我们仅展示了动作 2 和动作 3 的说明，因为只有目标 2 和目标 3 在 $t = 4000$ 被跟踪。

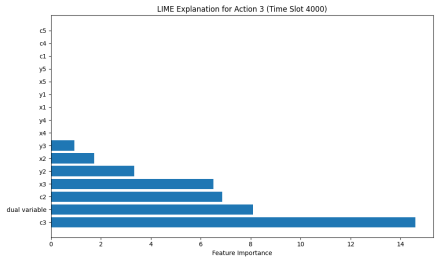


(a) 目标到雷达的距离 (b) 时间分配策略由 DDPG 代理预测 (c) 时间分配策略由 DL-LIME 预测

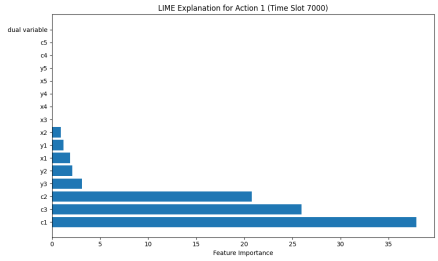
图 2: DDPG 和 DL-LIME 时间分配策略的比较



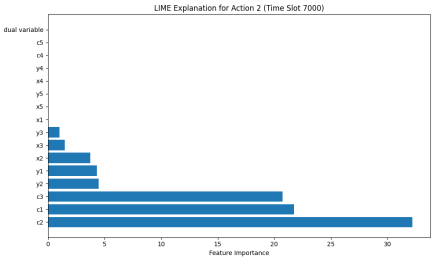
(a) 动作 2 在 $t = 4000$



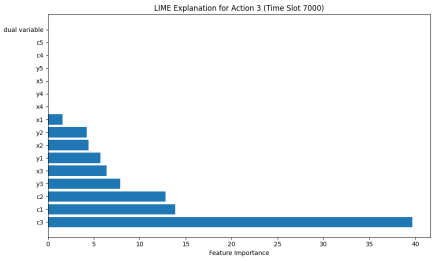
(b) 动作 3 在 $t = 4000$



(c) 动作 1 在 $t = 7000$



(d) 动作 2 在 $t = 7000$



(e) 动作 3 在 $t = 7000$

图 3: DL-LIME 解释在不同时间点的组件重要性分析。(a)-(b) 显示了对 $t = 4000$ 时间点动作的解释，而 (c)-(e) 则展示了对 $t = 7000$ 时间点动作的解释。

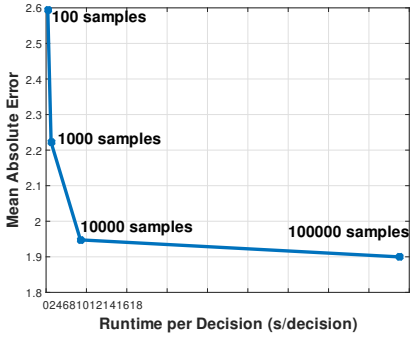


图 4: MAE 与运行时间之间的权衡关系对于 DL-LIME 而言

当 $c_{t-1}^n(\tau_{t-1}^n)$ 在解释其对应的 Action$n$ 时占据主导地位，其他目标的跟踪成本组成部分也显示出重要的作用，如图 3所示。这表明模型在进行分配决策时会考虑目标之间的相互作用。这种解释还显示 DDPG 代理并不孤立地为所有目标做出决策。相反，它权衡所有目标的跟踪成本以实现平衡分配。

VI. 结论

本文提出了一种 DL-LIME 框架，通过在采样过程中引入深度学习算法来增强传统的 LIME 方法，以考虑特征相关性。所提出的 DL-LIME 的性能使用多个指标与传统 LIME 进行了对比评估：对神经网络的保真度、任务表现和计算运行时间。数值结果表明，在雷达资源管理中，DL-LIME 在平均绝对误差（MAE）和特定任务指标上均表现出色。此外，该框架显示出了强大的能力来近似 DRL 代理的局部行为，这一点通过在具体时间点对解释进行详细分析得到了证明。这些结果显示 DL-LIME 不仅改进了传统 LIME 的表现，还为复杂的

DRL 算法在雷达资源分配中的决策过程提供了可解释的见解。

参考文献

[1] T. Lillicrap, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[3] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[5] C. E. Thornton, M. A. Kozy, R. M. Buehrer, A. F. Martone, and K. D. Sherbondy, "Deep reinforcement learning control for radar detection and tracking in congested spectral environments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1335–1349, 2020.

[6] M. Stephan, L. Servadei, J. Arjona-Medina, A. Santra, R. Wille, and G. Fischer, "Scene-adaptive radar tracking with deep reinforcement learning," *Machine Learning with Applications*, vol. 8, p. 100284, 2022.

[7] S. Durst and S. Brüggenwirth, "Quality of service based radar resource management using deep reinforcement learning," in *2021 IEEE Radar Conference (RadarConf21)*. IEEE, 2021, pp. 1–6.

[8] P. Pulkkinen, T. Aittomäki, A. Ström, and V. Koivunen, "Time budget management in multifunction radars using reinforcement learning," in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–6.

[9] Z. Lu and M. C. Gursoy, "Resource allocation for multi-target radar tracking via constrained deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 6, pp. 1677–1690, 2023.

[10] Z. Lu, M. C. Gursoy, C. K. Mohan, and P. K. Varshney, "Learning-based cognitive radar resource management for scanning and multi-target tracking," in *ICC 2024 - IEEE International Conference on Communications*, 2024, pp. 2785–2790.

[11] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[12] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley, "Explainable machine learning in deployment," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 648–657.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[14] S. Lundberg, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[15] M. Christoph, *Interpretable machine learning: A guide for making black box models explainable*. Leanpub, 2020.

[16] S. Shi, X. Zhang, and W. Fan, "A modified perturbed sampling method for local interpretable model-agnostic explanation," *arXiv preprint arXiv:2002.07434*, 2020.